A GPS inspired Terrain Referenced Navigation algorithm

A GPS inspired Terrain Referenced Navigation algorithm

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 24 november 2014 om 10:00 uur

door

Daniela VAMAN

Inginer Diplomat Inginerie Electronică, Telecomunicaţii
Universitatea Technică Gheorghe Asachi Iaşi, Roemenië
geboren te Botoşani, Roemenië

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr. A. Yarovoy
Copromotor:
Prof.dr.ir. E. Theunissen

Samenstelling promotiecomissie:

Rector Magnificus,	voorzitter
Prof.dr. A. Yarovoy,	Technische Universiteit Delft, promotor
Prof.dr.ir. E. Theunissen,	Nederlandse Defensie Academie, copromotor
Prof.dr. P.J. Oonincx,	Nederlandse Defensie Academie
Prof.ir. P. Hoogeboom,	Technische Universiteit Delft
Prof.dr.ir. R.F. Hanssen,	Technische Universiteit Delft
Prof.dr.ir. A.J. van der Veen,	Technische Universiteit Delft
Prof.dr.ir. M. Uijt de Haag,	Ohio University
Dr.ir. G.J.T. Leus,	Technische Universiteit Delft, reservelid

This research was supported by the **Netherlands Defence Academy**.

ISBN 978-90-5335-961-7

Dissertation at Delft University of Technology.

Author e-mail: `dana.vaman@yahoo.com`

pentru tatăl meu

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 TRN in the Era of GPS

The Global Positioning System (GPS)[1] allows land, sea and airborne users to accurately estimate their location, speed and time continuously, under all weather conditions, anywhere on earth and with a relatively low-cost. Thus, it is no wonder that, for the last twenty years, GPS has been dominating the market in positioning and navigation. GPS has become a critical asset for U.S. military and for other nation's defense forces. NATO has made GPS devices standard for use by its members. GPS is needed in practically all military operations: to guide troop movements, synchronize communication networks and guide most vehicles and aircrafts used in combat zones. In the civilian sector the area of GPS applications is extremely vast, far exceeding those of the military. Besides transportation, GPS is also used in timing of financial transactions, mobile communications and data networks, to name just a few.

Yet, GPS is an unsecured technology. Main vulnerability of the system comes from its low signal power, which makes it susceptible to interference: environmental and man-made, accidental and deliberate. Jamming (i.e. the denial of the availability of a service) and spoofing (i.e. deceiving the receiver by using counterfeit signal) are the most cited threats to GPS today. In its simplest form, a jammer transmits powerful noise drowning out the GPS signal. In [7] it is shown that, theoretically, a jammer of only 10 milliwatt transmit power can prevent a GPS receiver from acquiring

[1] The USA's NAVSTAR GPS is the most widely used Global Navigation Satellite System (GNSS). Russia also owns a global operational GNSS: GLONASS. China is in the process of expanding its regional Beidou navigation system into the global Compass navigation system, meanwhile the European Union's Galileo positioning system is a GNSS in initial deployment phase, scheduled to be fully operational by 2020 at the earliest [6]. The vulnerabilities discussed in this section extend to all GNSS implementations.

the coarse-acquisition code at a distance of 10 km and can cause a receiver already tracking the code to lose lock if in the range of 1 km from the jammer. Even a military receiver using precision codes will stop tracking when being within a few hundred meters distance. Typically, jammers are used by commercial drivers to avoid vehicle monitoring, in thefts or smuggling acts and, although illegal, these actions do not pose any serious threats to public safety. However, examples that highlight the real risk of GPS jamming have also been reported. In January 2007, naval signal jamming tests performed in San Diego harbor accidentally disrupted GPS services causing havoc on the air-traffic control and harbor traffic management, affecting cell-phone use and emergency pagers from a near-by medical center [8]. In 2009 the operations at the Newark Liberty International Airport in New Jersey were disrupted twice a day for a month because of a truck operating a GPS jammer which was passing on a nearby located highway [9]. Another example comes from UK where the General Lighthouse Authorities launched eLoran as a back-up ship navigation system to tackle the problem of jamming and GPS signal loss in the English Channel. No major GPS spoofing incidents have been reported, however it has been demonstrated that a spoofing attack can work. In June 2012, Todd Humphreys and his research team from University of Texas demonstrated the controlled capture of a small, civil drone aircraft at White Sands Missile Range, New Mexico using counterfeit GPS signals transmitted from a distance of 0.3 miles [10]. And a year later, in July 2013, the team proved again its ability to broadcast false GPS signals by taking control of an $80 million yacht's navigation system and sending it on a completely different path from the intended one, using just a homemade device [11]. Thus, capability to perform attacks on GPS exists and it might be only a matter of time until these types of academic demonstrations become real threats.

Within the navigation community GPS vulnerabilities have been gradually acknowledged and solutions are being explored. Associated proposals and developments focus on different aspects: legal protection and law enforcement, enhancing receiver jamming protection, developing anti-spoofing technology and exploring technologies that can act as back-up navigation systems. Critical applications call for at least one alternate method of positioning completely independent from the primary. Inertial navigation systems (INS) have been viewed as the answer so far. However, the reckoning errors are a major drawback for the inertial technology and INS requires periodic alignment. This is normally achieved through GPS, but in this way the navigation system is still dependent on GPS and suffers from the same limitations. Terrain referenced navigation(TRN) is an alternative to GPS that could be used for INS alignment or simply to provide redundancy as a backup system. TRN methods rely on the comparison between measured terrain features and digital elevation maps (DEM) to achieve navigation. This is not a new technique. TRN has an early history, dating back to the 1950s. However, forerunner

terrain systems were completely analog and required the assistance of the human factor. With the 'digital age', TRN methods started to become practical. The 70s and the 80s were the decades which account for most developments in this field and have seen the completion of the most famous variants of TRN: TERCOM and SITAN. With GPS still on the drawing board, TRN methods were quickly picked up for military usages, finding their niche as guidance systems for cruise missiles and fighter aircrafts. One of the downsides of being a military technology is that the technical details remain practically invisible to those outside of the community. On the other hand, technologies used in civilian applications have a larger consumer market which typically provides the business case for continuous development and acts as a strong driving force for innovations. For TRN such a vast consumer market has never existed. Beginning with the late 90s the interest in terrain navigation started to fade and, with the advent of GPS, some of the originally intended applications for TRN even ceased to exist.

The main driver for pursuing a TRN capability is the desire to have a fully autonomous navigation system. TRN requires no external source of information, therefore is extremely robust against interference. Among other areas of advantage, we remind the long term stability of the terrain signal, the relative ease of maintaining accurate DEMs and the cost-effective implementation of such a system. But, if we advocate for a broader role of TRN in nowadays navigation, the limitations must also be addressed: limited availability and a rather low accuracy. A TRN system will not work over all types of terrain. The availability of the system will be limited by totally featureless terrains, such as water or extremely flat areas. The accuracy of a TRN system is highly dependent on the mapping quality and the beam width of the terrain sensor. Traditional implementations, such as TERCOM or SITAN, used radar technology to measure terrain elevation and reported accuracies of about 30 m at best. Nowadays, other terrain sensing methods exist(capable of providing more accurate measurements): interferometric synthetic aperture radar(SAR), Doppler laser or laser rangefinder. Therefore, the observational TRN data could be improved by using more accurate sensors. Airborne Laser Scanner(ALS) is also a mainstream mapping technology. Typical figures in the published literature for attainable spatial resolution for ALS created elevation maps go up to 1 m, with elevation accuracies of up to 30 cm RMS. Thus, it turns out that (at least partly) the TRN limitations are not fundamental to the technique, but are just a consequence of the employed technology and the designed implementations. So, if hardware that can provide us better data exists, why not use it? And these are not the only enablers that warrant a re-investigation of the current potential of TRN technology. Over the last decades there has been a tremendous increase in digital signal processing capabilities, real-time computing power and data-storage. Finally, we should note the growing interest in TRN as an alternative navigation technique for GPS-denied environments in both

academic environment and industry. For example, at Ohio University several research projects on the use of ALS for terrain navigation were carried out within the last few years [12], [13], [14], [15], [16]. In industry, we note the development and marketing of PTAN, a Honeywell product which achieves terrain based navigation using SAR technology for terrain measurement [17]. And more recently, in 2012, TERPROM (a TRN system developed by Goodrich Corporation) has been selected as navigation system for the Airbus Military A400M versatile airlifter.

1.2 Aim and main challenges of the thesis

Given the arguments outlined in the previous section, it is reasonable to ask what potential may exist for improving the existing TRN systems or developing new approaches. The main goal of this dissertation is to improve TRN performance through novel signal processing concept. More specifically, the approach is to explore digital signal processing techniques that were originally developed for the acquisition and tracking of GPS signals in the field of TRN. Such an approach may puzzle the reader at first because, from a technical perspective, GPS and TRN seem quite different. But, at a closer look, several correspondences exist between the process of tracking the code component of the GPS signal and the process of finding a 'particular' TRN signal within the elevation map.

A typical terrain referenced navigation system uses speed, heading and time to establish the relative horizontal position between subsequent elevation measurements. Any error in speed, heading or time will cause an error in the resulting relative position. If the speed or heading error contains a bias, this will cause a gradual reduction in the correlation. To prevent that a reduction in correlation causes the estimated position to drift away, the main idea behind the research described in this thesis is the use of arrays of terrain elevation measurements with intentional (positive and negative) offsets in speed and heading in a tracking-loop configuration. It is known that such a concept works well for optimized signals, such as the ones used in GPS. The viability of this idea for a signal defined by a series of terrain elevation measurements is further explored in the thesis concluding that the concept is feasible and promising, but that unlike constant tracking loop design parameters used in GPS, an adaptive tracking loop would be required. Based on these considerations, a GPS inspired algorithm for TRN has been designed and a model of the envisioned adaptable tracking loop has been implemented in Matlab.

The line of reasoning, followed throughout the design process and mirrored in the structure of the dissertation, can be summarized by the list of questions iterated below. For each bullet, the list also provides a brief explanation on the factors that triggered the design question.

- **What is the best way to use the terrain signal?**

  When charting the history of TRN development, it can be seen that various approaches to terrain navigation have been proposed and demonstrated to date. Thus, when aiming to improve TRN performance, the first challenge lies in the recognition of the associated opportunities in the field.

- **How to design a tracking-loop based TRN algorithm which accounts for the specific terrain elevation profile properties?**

  The key in utilizing terrain-based measurements for position estimation is the uniqueness of terrain, which allows for a singular match between the measured elevations and the database. Typical approaches consist of either making a judicious terrain selection (approach possible for position fixing methods) or providing estimation strategies to identify and eliminate false position estimations. Given that the terrain signal has such a large impact on the performance of a TRN system, it was intended to develop a framework in which there are no constraints placed on the signal, but rather the system's parameters are adapted to the properties of the terrain.

- **How to design a tracking-loop based TRN algorithm which accounts for the performance of the sensors?**

  Typically, a TRN system will integrate data coming from different types of sensors, the foremost being the terrain clearance height sensor. Because the motion of the platform between individual terrain measurements needs to be taken into account, speed and heading measurements are required. Early implementations would prohibit freedom of manoeuvre during data collection, but such an approach is not realistic. Today, most of the existing applications rely on high accuracy inertial sensors, making TRN a rather expensive technology. Additionally, there is a general lack of studies assessing how sensor accuracy impacts TRN performance. The challenge here lies in understanding, for the proposed architecture, the performance trade-offs involved in sensor choice and designing a framework that could also support low-accuracy sensors.

- **How to specify sensor performance requirements as a function of accuracy requirements for a (specified) minimum terrain signal content, given the proposed TRN architecture?**

  Once an algorithm has been designed that can be configured to accommodate a range of sensor performance, and the earlier-mentioned possible performance trade-offs are better understood, it is desired to shift from a sensor-based to a performance-based approach. Performance requirements of the identified application identify sensor specifications and subsequently the choice of

navigation sensors. Accuracy is just one of the navigation performance requirements.

The answers to these questions are given in three steps. At first, an in-depth look at past and existing TRN systems, at their implementation and associated limitations, is provided. Such an extensive analysis allows a better understanding of TRN and serves to derive guidelines for the design aspects of the proposed approach. This includes selecting a processing method for the proposed algorithm and deciding which aspects are targeted for enhancements. The design of the algorithm starts with an evaluation of the proposed concept in a purely theoretical framework to answer feasibility-related questions. For the evaluation, a first model of the algorithm is implemented in Matlab. The relation between terrain characteristics and performance is analyzed and enhancements are brought to the initial design resulting in the development of an adaptive tracking algorithm, in which the tracking loops are configured based on the analysis of the terrain signal. The algorithm uses sensor measurements and, in practice, these data are affected by errors. Thus, the third step is the investigation of how sensor performance influences algorithm performance. A first analysis is performed using simulated data and it leads to further refinements of the algorithm, such as the addition of an integrity prediction capability. Next, the relation between performance of the algorithm and data quality is further investigated using real sensor data.

1.3 Outline

This section provides an overview of the remainder of the thesis and explains how the different chapters are linked together.

Chapter 2 starts with a general introduction on TRN. This includes a description of the concept, of the hardware requirements in a TRN system and of the advantages related to the use of this technique. Next, the history of TRN development is chronologically documented. The intention is to give the reader a clear understanding of the past TRN systems, their implementation, the hardware implications and associated limitations. The information presented serves as a background for section 2.4 which discusses where the opportunities lie and what potential may exist for enhancing TRN performance (from the perspective of the author). These observations provide the basis for the approach to TRN proposed in this dissertation. The chapter concludes with a list of preliminary design concepts and a blueprint of the proposed TRN algorithm.

Chapter 3 starts by explaining the similarities between GPS signal processing and TRN; similarities which have enabled us to design a GPS-inspired algorithm for terrain navigation. Next, an initial implementation of the TRN algorithm is presented

and discussed in detail. Beyond similarities, there are also important differences between the GPS signal and the terrain elevation profile. These differences and their impact are discussed and analyzed. Mitigation strategies are presented. In order to improve the algorithm's performance, it is shown that an enhancement is needed: the parameters used in the configuration of the tracking loops must be set according to the terrain characteristics. This results in the concept of the adaptive tracking algorithm, which is discussed in the last sections of the chapter.

An in-depth sensitivity analysis is carried out and detailed in **Chapter 4**. The effect of uncertainty in speed, heading and terrain measurements are discussed. In this way, the issue of sensor performance versus algorithm performance is tackled. Based on the findings in this chapter, a new feature is added to the algorithm: integrity prediction.

Recorded sensor data from flight tests are used to validate the proposed TRN algorithm. **Chapter 5** presents the results obtained when using measurement instruments based on both radar and laser technologies.

Chapter 6 summarizes the results of the study presented in this thesis, draws the main conclusions and offers some ideas for future work.

Chapter 2

TRN history, trends and the unused potential

2.1 Introduction

The idea to use a stored sequence of terrain elevation samples along a planned route as the means for comparison-based position estimation, often referred to as TRN, can be traced back to the fifties of the previous century. Back then the technological possibilities were rather limited, so forerunner terrain sensing systems were completely analog and/or would require the need of the human factor. However, the basic concept and principles have not changed so much. In 1958, Berger was describing the rationale behind TRN technique: "A terrain profile map along a selected route of an aircraft flight over broken ground is unique, and different from any other profile map, just as a selected human fingerprint is different from every other human fingerprint. It follows that, by use of a standard profile map for comparison, it is possible by recording the terrain profile during flight to sense errors in distance flown and in course angle and to correct these navigational elements so as to follow the standard profile map path exactly" [18]. With the micro-miniaturization of computer circuits that had increased computational capabilities, the availability of compact mass memory and the development of velocity, acceleration and direction sensing devices of high accuracy, TRN methods became practical. During the 60s and the 70s TERrain COntour Matching (TERCOM), arguably the most famous variant of the TRN family, was developed from the efforts of several R&D programs within different companies. TERCOM uses a correlation based algorithm, processing a series of terrain elevation measurements at once. By mid 70s, TERCOM concept had matured, proven to work and been committed to production. It was reliable, required no intervention from

the crew and provided a high degree of accuracy (according to the 'then' standards). However, the system could not work continuously as storage capacity was a significant limitation. Eventually, TRN systems "established [themselves] as effective position fixing systems suitable for use in manned and unmanned vehicles" (A. J. Henley). In fact, a term often used for TRN techniques is Terrain Aided Navigation (TAN). With the GPS still on the drawing board, the 80s and early 90s proved to be prolific for the TRN development. Sandia Laboratories introduced the sequential processing algorithm SITAN that uses the Kalman Filter (KF) to model INS and terrain errors. The use of Bayes theorem (and other statistical tools later on) within the TRN frame also dates from this time period. During the next decade, terrain sensing technologies were somewhat left in the shade. On one hand GPS had become already a fully functional, widely deployed system and much of the attention of the navigation community was oriented in its direction. Another factor would be a certain 'limitation' that TRN had reached at that time, leaving little space for improvement within the current state-of-art. However, this began to change as technology advanced to the point where other terrain sensing methods became practical: interferometric SAR, Doppler laser and laser rangefinders. All these sensors can provide finer spatial resolution than is possible to obtain with conventional beam-scanning means. Let's take the last example: an ALS based TRN system can make measurements in the along-track and cross-track directions, with a signal having a milli-radian beamwidth. This will increase the spectral content of the ground measurements data, thus allowing higher accuracy in the TRN estimates. Typical figures in the published literature for ALS attainable elevation accuracy are: from 5cm at 500m to 15 cm at 2000m, with still better figures claimed by certain suppliers [19]. ALS is also a mainstream mapping technology. ALS created DEMs can have spatial resolutions of up to 1m and elevation accuracy of 30 cm RMS (with prospect of further improvement in performance). And these are not the only enablers that warrant a re-investigation of the current potential of TRN technology. Over the last decade there has been a tremendous increase in digital signal processing capabilities, real-time computing power and data-storage field. Given all these advances in technology, it is reasonable to ask what potential may exist for improving the existing TRN systems or developing others. The present chapter attempts to answer to this question by looking back into the history of TRN development, from the early analog systems to the present ones. It is important to have a clear understanding of the past TRN systems, their implementation, hardware implications and limitations. More importantly, were these limitations conceptual or technological? And are they still valid with the know-how of today?

2.2 What is TRN

2.2.1 TRN concept

Figure 2.1 illustrates the operational principle of a TRN system. By subtracting the above ground level (AGL) height from the mean sea level (MSL) height, an estimate of the elevation of the terrain beneath the aircraft is obtained. The measurements are then compared to a DEM to obtain an estimate of the position. The minimum hardware requirements to accomplish a TRN capability are: clearance height sensors, DEMs, computational unit. Some systems may require additional information (e.g. velocity sensors, heading sensors etc.). Throughout this thesis there is a clear distinction made between the sensors that directly relate to the terrain and those that provide supportive information. Sensors belonging to the first category are referred to as *primary sensors* and the rest as *supportive sensors*.

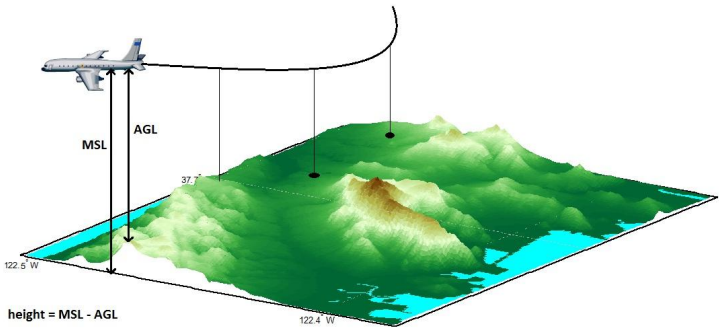


Figure 2.1: TRN concept for aerial applications

2.2.2 TRN advantages

The main driver for pursuing a TRN capability is the desire to have a fully autonomous navigation tool, both in technical and political terms. TRN requires no external sensors or emitters, therefore is extremely robust against (un)deliberate interference or jamming. This aspect clearly distinguishes TRN from GPS, as underlined by Alan J. Robins in [20]: "[TRN is] independent of the positions, the visibility or even the deployment of a network of satellites and is not dependent upon a friendly Government supplying the essential P-codes". A TRN system can operate during day/night, all weather and low/high altitude conditions. The TRN signal (i.e. terrain elevation) is directly measureable and relatively stable in time. Given all the

above mentioned advantages, it's easy to understand why TRN has found practical application primarily within the military technology (mostly as a guidance aid for cruise missiles). Paul Hinrichs, who talks about TRN benefits in [21], states that "positioning systems which utilize variations in terrain elevation as the signal closely approximate the ideal system for operation in a wartime environment".

Given their capabilities, TRN can be regarded more as a complementary technique to other navigational methods, rather than as a competitor. For example, an INS provides position, velocity, attitude, angular rate and acceleration measurements with short-time noise characteristics, but its accuracy degrades over time. The GPS navigation solution is accurate and does not drift with time, but the satellite signals are vulnerable to interference. TRN systems are autonomous, but their performance (and hence their coverage area) is fundamentally limited by terrain characteristics. Due to these dissimilarities, an integrated GPS/INS/TRN approach could offer operational advantages.

2.2.3 Related applications

Besides navigation, there are additional applications of TRN in conjunction with elevation databases. Easily incorporated, these functions can be used to enhance safety and situational awareness during flight. Passive terrain following (TF) is one of the complimentary uses of terrain elevation databases for navigation. With a TF function, the position is estimated using other means and the database is used to compute a trajectory which follows the terrain profile within a pre-determined separation margin. Closely related to the TF function is the predictive ground collision avoidance(PGCA) function which provides alerts in case the extrapolation of the current state violates terrain separation minima. Today's Terrain Awareness and Warning System(TAWS) is based on this function. Similar to the PGCA capability, an obstacle warning function scans the database to determine the obstructions that may potentially lie in the path of the vehicle. Although historically the navigation, terrain following and terrain warning functions were developed separately, today's TRN-based systems typically incorporate all three[1].

2.3 History of TRN development

This section chronologically documents the history of TRN development. The intention here is to provide an ample view of terrain sensing technologies, from systems that are currently used or were used in the past to concepts and methods, disclosed in

[1] As noted on Goodrich's website [22], the TERPROM system comprises a wide range of functions: TRN, PGCA, obstruction warning and cueing, TF and terrain awareness display. TERPROM will be described later on in this chapter.

research articles and patents, that didn't materialize into (mass) production systems. Other TRN-related literature surveys, which served as reliable information sources for this chapter, can be found in [15, 23–26]. In the view of the author the main factors that played a key role throughout the TRN development process are:

- The area of application:
  Due to their inherent advantages, terrain sensing guidance and navigation systems have found practical application primarily within military technology. Driven by the need imposed at times by the political context, substantial efforts have been directed to push technological advancements and improvements in this area of research despite cost-related barriers. However, one important downside to be noted here is the secrecy surrounding military developments in general, which limits access to information. Looking at the overall development process, the general trend is one of segregation between the existing algorithms. This is most likely a consequence of lack of proper documentation of the operation of military TRN systems.
- The technical possibilities/limitations of the component elements of a TRN system:
  A TRN system comprises different components: databases, computational unit, storage unit, clearance height sensor, aircraft height sensor, navigation sensors, Kalman filters etc. It will be illustrated that, throughout the history, technological developments (in all the previously mentioned fields) have acted as barriers or enablers for TRN development.

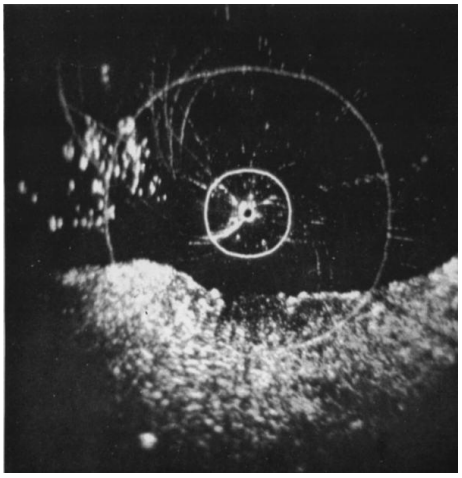
2.3.1 Analog systems

H2S[2] was the first airborne, ground scanning radar system and it was used during World War II by British forces to guide bombers to German cities during night or cloudy missions (with a first use dated on 30th of January 1943). The H2S aircraft unit can be seen in figure 2.2a. The employed radar was of pulse type, based on the cavity magnetron and operated at 10 cm wavelength (later versions reduced the wavelength to 3 and 1.5 cm). The radar was fitted on the belly of a bomber aircraft where the antenna would rotate to scan the terrain below and feed the reflections to a Plan Position Indicator (PPI) display producing a map that would be interpreted by the aircraft crew [1]. Figure 2.2b shows a photo taken from the display of the H2S radar of Normandy beach, during the liberation of France, on 6th June 1944.

U.S. Patent 2,526,682 [28] by Henry C. Mulberger and James E. Bellitt, filed in 1946 and granted four years later, describes a similar invention to the H2S. It is

[2]It is not clear how H2S received its name. In [27] is said that it may be "an acronym for height to slope or home sweet home. It is also said that the name stands for the chemical symbol for hydrogen sulfide, referring to the fact that the device 'stinks' and another explanation is that 'S' is referring to the frequency. Most likely though the name does not refer to anything for military security reasons"

(a)



(b)

Figure 2.2: (a) H2S aircraft unit [1], (b) Photo of H2S radar display [2].

claimed that the proposed system can determine "the position, direction, altitude and ground speed of an aircraft flying over an obscured predetermined terrain" by "relatively coordinating the photographic moving picture strip image record of the said terrain" with "the radar cathode ray fluorescent screen image of the terrain, while flying thereover". The correlation between the PPI display and the recorded picture strip was also done manually, by the crew.

One important component missing from these early systems was the ability to perform the correlation automatically. In 1947, the Goodyear Automatic Corporation (GAC), with the sponsorship of the Wright Air Development Center, started investigations on automatic radar map matching and by mid-1948 an experimental terrain sensing guidance system was completed: Automatic Terrain Recognition and Navigation (ATRAN). As development progressed, ATRAN became in 1950 the guidance system for TM-61B MATADOR missile. The system was also used for the successor of MATADOR, the TM-76A MACE missile, until the early 60s. ATRAN is the first fully autonomous terrain referenced automatic guidance system. R. Koch labelled it "the gran-daddy system" and was quite accurately in doing so, considering that ATRAN's basic concept and principles were embodied in several modern systems that followed on. According to [3] "through a map-matching

correlation process, ATRAN continuously tracks the ever changing pattern of terrain features by comparing the observed radar returns to a sequence of reference images in its memory". ATRAN used an X-band short-pulsed non-coherent scanning radar sensor to produce the radar imagery. The reference images were stored on a 35 mm film strip. The comparison between the two was performed automatically, using an optical correlator. ATRAN used only "a simple pendulous erected vertical gyro and a directional gyro" [3] for basic attitude and directional references. Note that ATRAN did not use an INS. Although the benefits of INS updating were experimentally demonstrated at the time, technology was not mature enough to allow integration. Being developed prior to micro-miniaturization of computer circuits, ATRAN was also very large in terms of weight, volume and power. Another distinct disadvantage was that it required extensive low-level aerial photography of each potential missile flight path. The danger to flight crews when flying over denied territories made the system impractical. To somehow overcome this limitation, the Army Map Service created 3D relief models of the terrain of interest (painting known objects in white to produce strong radar reflections) and then made trajectory films by 'flying' a motion picture camera over these models, as seen in figure 2.3. The accuracy of ATRAN system was listed to about 1000 feet [3].

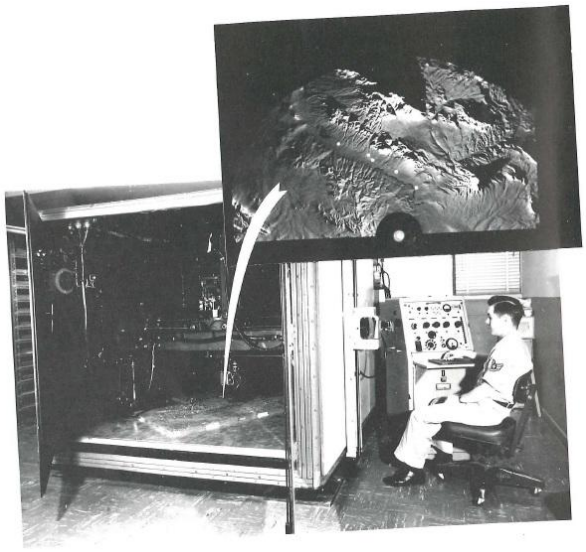


Figure 2.3: Map generation process for ATRAN [3]

Another analog aircraft terrain navigation system is described in U.S. Patent 2,847,855 by F. Berger [18]. The novelty in this approach is that terrain elevation

measurements and not images are compared to obtain position (the used correlator is described in another patent by Berger [29]). Elevation data is obtained by subtracting radar altimeter height from an absolute altimeter.

2.3.2 Digital systems

2.3.2.1 Batch processing TRN. TERCOM

The TERCOM acronym stands for Terrain Contour Matching and is the name of a guidance system developed in the U.S. to update INS for land-attack cruise missiles: air-launched, ground-launched, sea-launched and advanced cruise missiles [30]. Essentially, TERCOM determines the location of an airborne vehicle with respect to the terrain over which the vehicle is flying by digitally correlating a measured terrain profile with terrain profiles stored in the memory of an on-board computer. The TERCOM signal is the terrain elevation profile itself, therefore the system is more reliable than the previous radar map-matching systems that attempted to match reflectivity maps and were subject to weather and seasonal effects. The ground profile is acquired using a combination of radar altimeter (AGL height) and barometric altimeter (MSL height) outputs. The TERCOM 'map' is a rectangular matrix of numbers, each number representing the average elevation of the terrain, as a function of location. The TERCOM 'matching process' consists of comparing the measured profile with each down-track column in the reference map matrix. Common metrics used in the correlation algorithm are: Mean-Absolute Difference (MAD) or Mean-Squared Difference (MSD). TERCOM system diagram and map concept are illustrated in figure 2.4. More details on TERCOM's operational principle can be found in [31].

TRN techniques that process together a set of terrain measurements are known as batch processing methods. Due to the limited amount of memory available in mass storage devices at the time of TERCOM development, the amount of terrain data that could be stored in an on-board computer was too small to encompass an entire flight. The solution was to integrate TERCOM with an INS platform: while the INS assures the fundamental means of guidance, TERCOM produces occasional fixes to correct the inertial drifts. Being a guidance system developed for military applications, it was requested for TERCOM to have a high accuracy. One key element was the choice of the terrain maps. As Golden underlines in [4] "TERCOM will not work over all types of terrain. Generally the rougher the terrain, the better. However, good terrain must be more than just rough, it must be unique. That is, a given profile out of TERCOM map must not resemble any other in the map". The Defense Mapping Agency worked on map-selection methodologies and TERCOM performance predictions. One approach taken was the statistical description of the shape of the

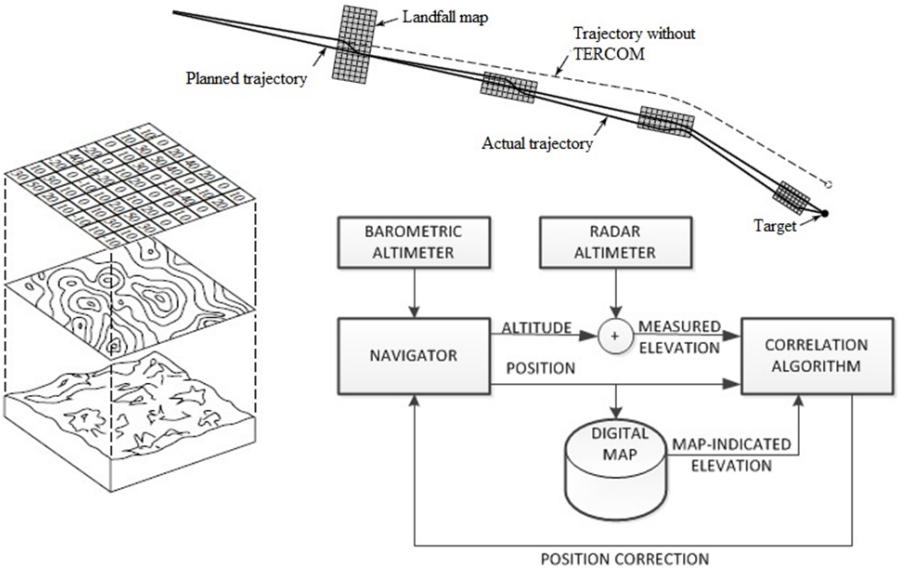


Figure 2.4: The TERCOM process: TERCOM maps [4] and system diagram

terrain, in terms of standard deviation and correlation length as discussed in [21]. Mission-planning became an important aspect in the TERCOM process. The absolute accuracy, however, is based on the accuracy of the radar mapping information and the ability of the processor to compare the altimeter data to the map quickly enough as the resolution increases. This limited the first generation TERCOM systems to target on the order of hundreds of meters. As soon as the advances in digitized imagery allowed computer storage of images, the Digital Scene Matching Area Correlator (DSMAC) method was used for terminal guidance. The development of Shuttle Radar Topography Mission (SRTM) maps was an opportunity to re-take TERCOM development. According to [30], new types of maps were developed: sub-terminal maps (originally entitled small-cell TERCOM) and the improvements consisted in reduced noise at match and more robust correlation. This new TERCOM planning capability has been implemented in the Tomahawk planning system and became operational in 2007.

Historically, TERCOM has been under development for more than 15 years and has gradually evolved from several R&D programs that perfected different areas of the overall process. According to [4] Chance-Vought originally conceived TERCOM in

1958 for application on a nuclear powered supersonic low-altitude missile (SLAM[3]). The concept was initially named "Fingerprint". Although the SLAM project was soon cancelled, Chance-Vought continued to do feasibility studies on TERCOM until 1961 with the sponsorship of Aeronautical Systems Division (ASD). Chance-Vought became part of Ling-Temco-Vought Electrosystems, Incorporated (LTV-E) and under this patronage research focused on applying TERCOM to low-flying aircrafts. The program was called Low Altitude Contour Matching, took place between 1963 and 1965 and had as a goal the design of a complete fix/update subsystem. A patent assigned to LTV-E could be found dating from this period: "Fix-taking means and method" by W. C. Hallmark, [32]. In between 1963 and 1971 research continued, but with no real, significant advancements to be made. Some of the examples are: the Rapid Contour Matching program which attempted to improve the computation procedures; Space and Missile System Organization sponsored research programs on potential applications on ballistic missiles: TPLS (Terminal Position Location System), TERSE (Terminal Sensing Experiment), TERF (Terminal Fix) and TSOFT (Terminal Sensor Overland Flight Test). In 1972 LTV split and E-systems was formed. An E-systems program under ASD sponsorship entitled "Project update" continued research on TERCOM. In the contract mission planning procedures and source data requirements were also mentioned. During the same time Naval Air Systems Command (NAVAIR) became interested in the concept and sponsored a feasibility study for incorporating TERCOM as an aided inertial navigation system for cruise missile guidance. The study was performed by the Applied Physics Laboratory of Johns Hopkins University (JHU/APL) and the investigated system was named TAINS (Terrain Aided INS TERCOM). Given the results of the study, NAVAIR contracted E-Systems to perform flight tests using TAINS. The experiments were successful, demonstrating not only that the TERCOM concept for cruise missiles was a feasible one, but also the importance of terrain selection. Until 1974 JHU/APL, McDonnell-Douglas (MDAC), E-Systems and General Dynamics/Convair performed their own studies on the subject of selectivity and suitability of terrain (characteristics) for TERCOM maps. In 1975 NAVAIR selected MDAC and E-systems to competitively develop a prototype TERCOM guidance set for Submarine Launched Cruise Missile (SLCM) program. Eventually MDAC won the Cruise Missile Guidance Set contract. In 1975 the Department of Defense decided to supply also the Air Launched Cruise Missile (ALCM) program with the guidance system used by SLCM. The first fully guided missile flight was flown in October 1976. Some of the U.S. missiles that employ a TERCOM system are: AGM-86 ALCM, AGM-129 ACM and the U.S. widely publicized Tomahawk. Information was found that TERCOM navigation was last used operationally with Tomahawk in 1998, but "remains a

[3]To not be confused with(the later on introduced) Standoff Land Attack Missile, which has the same acronym.

selectable navigation mode for all variations of current Tomahawk, a critical capability as the threat of GPS jamming increases" [30].

2.3.2.2 Sequential processing TRN. SITAN

Beginning with mid-70s, Sandia Laboratories started developing a new terrain navigation algorithm: Sandia Inertial Terrain Aided Navigation (SITAN), "initially formulated for possible use in a weapons delivery system" [33]. Although SITAN uses the same hardware components as TERCOM, as depicted in Figure 2.5, the novelty of the approach consisted in the method of processing the radar altimeter data. This approach uses KF theory to optimally combine the inertial and auxiliary-sensor (radar-altimeter) to estimate the state of the vehicle. As opposed to the batch processing methods, this algorithm recursively/sequentially treats each terrain measurement as the measurement to be Kalman processed [34–36]. As a consequence, the SITAN algorithm would not provide fixes, but is able to 'continuously' update the inertial navigator. Another advantage of a sequential processing method is the fact that terrain data is processed as it is collected, therefore reducing the amount of memory needed in the computation process. Meanwhile availability of mass-memory storage devices had been a troublesome issue for TERCOM, a SITAN map for a pre-programmed flight "could be accomplished by storing only two terrain profiles, one on each side of the flight path, from the release point to the target" [35]. Furthermore SITAN allows freedom to manoeuvre during data collection and is able to correct position errors in the presence of significant heading and speed errors. The main limitations of SITAN are: the need of an accurate initial position and terrain linearization around the assumed position (mandatory condition for incorporating data in the KF). While the assumption of linearity can be possible over certain areas, terrain generally has an undulating and nonlinear nature. The first attempt in solving this issue was to apply local slope linearization, with the use of an Extended Kalman Filter (EKF). The problem was only partially solved, as situations when the linearization error is comparable to the measurement error still lead to filter divergence. In [37] Hostetler discussed two methods to overcome the divergence: the use of modified stochastic linearization techniques within a single KF framework and the use of a bank of KF running in parallel, each linearized over a different region.

In 1982 U.S. Air Force Avionics Laboratory became interested in SITAN and began funding an evaluation of Sandia's terrain navigation system on low-level attack aircrafts. The work evolved from data collection using a prototype system on an A7 aircraft to the implementation of SITAN on the Advanced Fighter Technology Integration (AFTI) F-16 aircraft. In [33] D. Boozer demonstrated that the performance achieved through the use the AFTI/SITAN algorithm was of 75m CEP in horizontal position and 13m RMS in vertical position. In [38], a version of

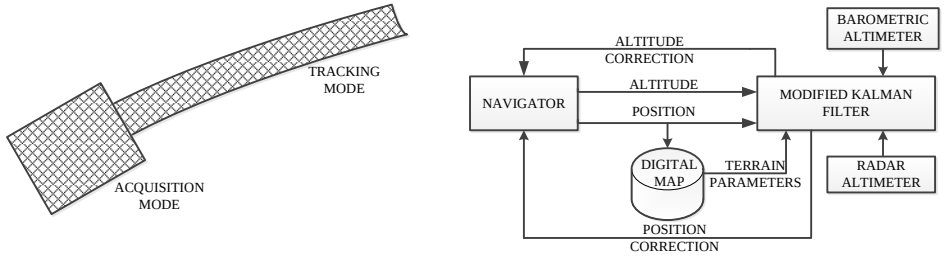


Figure 2.5: The SITAN process: SITAN maps and system diagram

SITAN developed for highly maneuvering helicopter applications is proposed. The algorithm uses Multi Mode Adaptive Estimation (MMAE) techniques. Experimental tests of Heli/SITAN showed that position estimation values could be obtained even with large initial position errors and the CEP was less than 50m.

Pei and Chen describe in [39] a terrain algorithm based on SITAN's functional principle. BITAN uses Kalman filtering, but has two main working modes: "Acquisition Mode works when the INS position error is very large and it can decrease the error rapidly using parallel filtering method, then Track Mode continues to correct the INS position errors and velocity errors using a five states KF". Performance accuracy for BITAN is stated in [39] to be of 39.5 m CEP. A modified BITAN algorithm based on nonlinear optimal filtering is described in [40]. The proposed version promises improvements in accuracy and robustness when compared to its predecessor.

VATAN, another sequential TRN algorithm, is proposed in [41] by Enns (from MDAC) and Morrell. This method uses a discretized version of the continuous Viterbi algorithm (VA) to compute the navigation solution. According to the authors: "the VA does not require linearization, and it is robust with respect to partially observable system models; thus, VATAN promises good performance over a wide range of terrain without being susceptible to the divergence problems encountered using the EKF". Although the initial test results published in [41] were promising, no further research could be found on VATAN.

2.3.2.3 Bayesian approach to TRN

Beginning with mid 80's a new trend in TRN research was developing: looking at the matching process from a statistical point of view, as a recursive nonlinear estimation problem. Dr. Runnalls explored the application of Bayes' theorem to terrain navigation in [42]. This approach is applied in the StockPot Algorithm

Robust Terrain Aided Navigation (SPARTAN) technique, also described in [43]. The algorithm resembles SITAN, in that a KF is used to incorporate both inertial and terrain measurements. However the observational information consists of a batch (transect) of elevations and the Bayes formula is used to incorporate the measurement data into the estimation. The idea behind the method is the concept of likelihood function, as stated in [42]: "What would be the probability for the observed data set Y if the true position of the aircraft had been X?". When data for a transect become available, the system calculates the likelihood function for these data, over a search area determined by the current position uncertainty, as estimated by the KF. The likelihood function is added into a so-called stockpot function, which is a "function of two variables representing the horizontal components of the position error". Next a quadratic surface is fitted to the stockpot function and its parameters represent the measurement update to be fed into the KF. SPARTAN works with shorter transects relative to TERCOM, of 400m, therefore the fix can be applied sooner (note that in case of a 1 measurement transect the system would work as SITAN, without adding errors due to the linearization of terrain). Another advantage is that it does not need any capture or initialisation phase. As other TRN systems, SPARTAN's performance is limited by terrain characteristics and underperforms over flat areas. Henley discusses this aspect in [43] and finds solution in the integration with SMAC or GPS. Henley and Runnalls continued their research and in [44] they took a further step by applying Monte Carlo Markov Chain (MCMC) methods with a Bayesian network based fusion algorithm, but this approach requires a heavy processing load. Successful flight tests of SPARTAN TRN and TF were performed on AFTI/F16 aircrafts and the system was selected to equip UK Tornado aircrafts by the end of 1995. However, the system never went into production, as GEC Avionics (SPARTAN's developer) merged with BAE system, which later produced TERPROM.

In the late 90's, Dr. N. Bergman treated the topic of recursive Bayesian estimation, MCMC methods, particle filters (PF) and point mass filters (PMF) in TRN applications [23]. The main difference in his approach is that the proposed navigation filter computes a probability distribution function (PDF) of the aircraft position and updates this description recursively with each new measurement from the altimeter and the INS estimate. The PMF computes a discretized approximation of the PDF. The performance of the PMF approach was tested using Monte Carlo simulations. Bergman concluded that "the main advantages of the PMF are that it works for many kinds of nonlinearities and many kinds of noise and prior distributions. The main disadvantage is that it cannot solve estimation problems of very high dimension, because of computational complexity". More on statistical methods can be found in the works of Nygren [24], Schon [45] and Nordlund [46].

2.3.2.4 TERPROM

In [47] TERrain PROfile Matching (TERPROM) is defined as "a terrain based navigation solution that blends multiple sensor inputs providing an enhanced situational awareness to the pilot". Besides the main capability of the system (i.e. to assure "non-GPS dependent and accurate drift free navigation"), TERPROM provides additional functions like: PGCA System (by scanning ahead the terrain and providing audio and video warnings); Advanced Terrain Awareness Cueing (by providing information both ahead and on either sides of the aircraft); Obstruction and Wires Warnings; Terrain Awareness Display; Passive Ranging to points on ground; Database TF and Terrain and Threat Avoidance (by real time prediction of an optimum route through the terrain to a future waypoint) [22]. TERPROM has been specifically developed for military operations. Several variants of the system exist, in order to satisfy the demands of Fast Jet, Aircraft Transport, UAV and Helicopter Markets[4]. According to [20] development of TERPROM started as early as 1987 at British Aerospace, as part of a Ministry of Defence funded contract and has been in production since 1991[5]. Very little literature could be found on TERPROM. According to one article from 1988 by A. Robins [20], TERPROM is a two phase system: a 'batch mode', very similar to TERCOM, assures the initialisation and a 'single-shot mode', very similar to SITAN, assures continuous navigation. A more recent paper that presents TERPROM's state-of-art in 2009 [47], gives a different description of the system: "the vertical profile is combined with horizontal component to generate a 3D terrain profile. This terrain profile information is then fed into a multi-state adaptive KF contained within the TRN capability, which models and calibrates the errors within the INS, to find a corresponding terrain profile match within the database." Another interesting, but not further developed detail is that TERPROM "estimates the errors in both the INS and the terrain database". Currently, TERPROM requires a tactical grade INS or better to function effectively.

2.3.2.5 Other processing approaches

U.S. Patent 4,584,646 [48] by Chan et al., on behalf of Harris Corporation, describes a system for Correlation And Recognition of Terrain Elevation (CAROTE) that uses a correlation scheme that operates in the frequency domain. The proposed system is

[4] According to an article from 2009, TERPROM had been selected by 14 Nations worldwide for use on numerous platforms, including A-10, C-130, C-17, F16, Mirage 2000, Harrier GR7, Jaguar and Tornado. In 2012 TERPROM was selected to provide the Airbus Military A400M versatile airlifter with TRN capabilities.

[5] TERPROM is produced by Atlantic Inertial Systems, Plymouth UK, formerly a subsidiary of BAE Systems, acquired in 2009 by Goodrich Corporation.

of batch processing type. No information could be found on aspects of performance or implementation of CAROTE.

The frequency domain is also used in Patents 4,495,520 [49] and 4,520,445, [50] assigned to L. Keearns from E-Systems. In these patents the terrain database is stored using a set of discrete cosine transforms parameters, in an attempt to reduce the amount of required data storage.

U.S. Patent 4,829,304 assigned to Baird describes a hybrid TRN system that "contains a modified KF processor which continuously receives both TERCOM and SITAN control information so that the operation of SITAN processing is effectively continuously optimized" [51].

MDC's U.S. patent 6,218,980, filled in 1987 and eventually granted in 2001, describes another improvement that could be applied to the traditional batch processing TRN [52]. Altimeter measurements degrade proportionally to the altitude at which the missile is travelling. In this approach, Goebel et al. propose to make a transformation of the reference map by "replacing each altitude value of the original terrain map with a simulated value that the actual radar will measure" (as a function of height) in an attempt to improve the correlation process.

Granted in 1995, U.S. Patent 5,450,345 by Raymer, describes a SITAN-like system that increases the positional accuracy by dynamically compensating the INS measurements for the Schuler cycle [53].

2.3.3 Discussion on TRN processing methods

TRN techniques may be grouped into two main categories: batch and sequential processing methods (with TERCOM and SITAN being the most illustrative examples of each group). The basic operational principles of these two algorithmic techniques have been discussed in the previous section. Further on a summary of their main strong/weak points will be presented, along with other concluding remarks on this topic.

Batch algorithms have the advantage of working well for any initial position error, as long as the terrain shape is unique or 'uncorrelated' within the search area. Although the length of the terrain profiles (used in the correlation process) depends on the terrain characteristics, typical values found in the literature vary between 3 and 6 km. Therefore, in these cases, position updates are available only every few km. This amount was reduced by using likelihood or Gaussian fits, with the cost of increasing the computational load. The solution of a batch algorithm will also be subject of a quantisation error, due to the size of the search interval. One way of reducing this error is to use very small search intervals, but this will increase the processing time. Another option is to use a hierarchical search process. However, it should be noted that the speed with which computers can process data nowadays has

dramatically increased from the past.

Unlike the batch algorithms, no search is required in the sequential processing methods and position updates are continuously available. The main limitations of sequential techniques are: the need of an accurate initial position and the assumption of terrain linearity. When these conditions are not satisfied, the method is likely to fail due to filter divergence. As previously described, solutions have been found with the use of nonlinear estimation algorithms, VA, PMF or PF. This does not remove the problem completely and substantially increases the computational task. A single large error in the terrain height measurement or in the database may also lead to filter divergence. The batch algorithms are not so sensitive, as they tend to average this type of error.

A conclusive remark is that both batch and sequential methods are dependent on terrain features and mapping quality. Considering the second aspect, it is rather self-evident that the main requirements for the elevation databases are: to be available and to have an adequate quality (in terms of good resolution and low local error rate) [43]. When it comes to terrain characteristics, the issue is much more complicated. Clearly terrain features represent a fundamental limit to any TRN system: batch methods rely on the assumption of uniqueness of terrain, meanwhile recursive methods rely on (linearity assumption of) local terrain characteristics. The testing of the different existing TRN algorithms has made authors to almost always conclude that "the achievable accuracy is highly dependent on the terrain features", hinting that there is a need of adaptability of the system parameters to the terrain properties. However, it is very difficult to assess the information content of a terrain area or to obtain a single measure of the terrain quality for TRN. Often encountered parameters are: average height and standard deviation of height or slopes. This accounts for the roughness of the terrain, which is obviously very important. However, the way this variation occurs (the spatial frequency distribution) is just as important as the variation itself [54]. The rule of thumb is that the rougher is the terrain, the better will be the TRN performance. For example, when flying over a lake or a plain, the algorithm has to deal with measurements that have a low information content. However, excessive roughness is also likely to cause issues. The reason for which TRN systems underperform in this case is believed to be the larger mapping interpolation errors and the increased probability that the conventional altimeter is not tracking a point directly below the aircraft [54]. In [55], N. Bergman notes that there are mainly three types of terrain that need extra attention in the design of a TAN algorithm: repetitive, rough and flat. "To handle repetitive terrain two desirable features of the algorithm are recursiveness and ability to handle parallel position hypothesis. The case of rough terrain must be properly modelled and calls for an algorithm that takes some global approach and can handle unconventional noise characteristics. The flat terrain puts restrictions on the algorithm's sensitivity to non-exciting measurements."

2.3.4 Altitude terrain sensors used in TRN systems

Commenting on the role of the terrain sensor, N. Priestly notes in [54] that "the clearance height sensor is the critical item in any TRN system and the ideal tool is probably thought to be a sensor which always tracks the ground directly below the aircraft" and it could be added "with the highest possible accuracy". The most common instrument chosen to provide elevation measurements is the radio altimeter (RADALT). However, once the technology developments made it possible, TRN systems using other types of sensors have been proposed, such as: W-band radars, weather radars, interferometric SAR and ALS. The aim was to improve the obtained spatial resolution and thus, the positioning accuracy. This section continues with a review of different TRN systems from the perspective of the used terrain sensing technologies.

2.3.4.1 Radio altimeter

The RADALT is the most frequently chosen instrument for providing elevation measurements. Typically radio altimeters operate in the C frequency band, with a central frequency of 4.3 GHz. Most often modulation schemes are FM-CW and pulse, though spread spectrum can also be encountered. The invention of a batch processing TRN system that uses a mm wave sensor operating in the atmospheric window of 94 GHz was patented in 1990 by Lerche from Deutsche Aerospace [56].

2.3.4.2 Weather radar

Autonomous Precision Approach and Landing System (APALS) is a terrain referenced approach system designed to enable low visibility landings. The system consists of a modified X-band weather radar (MWx), RADALT, IMU, GPS and a radar feature database. APALS operates in INS/GPS mode and after crossing the initial approach fix switches to INS/MWx mode until the 100 ft Height Above Threshold. Then it switches again to INS/RADALT mode. Position accuracy of APALS is stated in [57] to be 1-2 m vertical and 2-3 m horizontal range. According to the developers of this system, the key to this accuracy is the use of MWx. APALS was patented in 2005 [58].

Another TRN system that employs weather radar is described in Patent no. 6,233,522 granted in 2001 to M. Morici on behalf of AlliedSignal Inc. The invention assures only position validation (and not positioning) by comparing a reflectivity map (obtained using the weather radar) with a stored elevation map using test statistics [59].

2.3.4.3 Interferometric SAR

Precision Terrain Aided Navigation (PTAN) system, developed by Honeywell, uses an interferometric SAR altimeter. PTAN's radar operates in the C-band and uses three receive antennas to measure both the range of the closest object and the angle to the closest return. This allows to reconstruct the ground elevation: "Comparatively speaking, the interferometer returns plot high points of the terrain below the aircraft like a fine meandering pencil line, compared with the broad-brush returns from a normal radar altimeter" [17]. For best performance, PTAN uses DTED Level 4 (3 m resolution) to deliver navigation accuracy within 3m. However, PTAN can be adjusted to accommodate any DTED Level. Similar to the terrain databases, PTAN is able to accommodate various options in terms of algorithms. In 2004 Honeywell was said to intend to come up with its own "customized correlation algorithm [...] the point is that you can use any of these [existing] algorithms to get the results with PTAN. We want to emphasize the system's versatility". U.S. Patent 7,409,293 granted in 2008 discloses an improved version of PTAN. In this case the navigation solution is computed as a combination between an INS position solution and a weighted, terrain correlated position solution. The weighting process is based on a map quality factor calculated from the DEM. The patent does not give further details on how the quality factor is calculated; it only states that a random distribution measurement algorithm is applied [60]. PTAN has been selected in 2006 to be used for the Tomahawk missiles.

2.3.4.4 Laser technology

Interestingly, the idea to use a laser ranger is much older than imagined. In [4] is said that LTV-E had a contract with Air Force Avionics Laboratory in between 1963 and 1966 to perform "flight tests using a small laser ranging unit instead of a radar altimeter to possibly improve on position accuracy". Apparently the benefits of using laser technology were early anticipated, but it is only at the present time that ALS are in full operational use. As described in [15], ALS systems are capable of producing thousands of high accuracy, spatially independent aircraft-to-ground range measurements per second (range measurements typically have accuracies better than a decimeter). The fact that ALSs are primarily used in mapping systems is another huge advantage, as it is easy to match the accuracy and resolution of ALS-based TRN systems with the LIDAR digital surface maps. Among the challenges with which the laser technology is still confronting are: operation in all-weather conditions, safety issues and (current) cost[6]. However, as with many technologies, it is expected that these challenges will be overcome in the near future. The first mentioning of a

[6]According to [19], the cost of a single full-blown airborne laser scanner system (including its obligatory GNSS/IMU unit and its optional digital camera) can be between $500.000 and $1.3 million.

laser range finder within a TRN system was found in U.S. Patent 5,047,777 “Linear method of navigation” by Dornier Luftfarht Germany, granted in September 1991 [61]. The laser range finder is used to classify the overflown terrain into segments such as: “forests”, “houses”, “lakes” etc. This data is then correlated with topographic maps containing information on the type of existing vegetation and built structures. Another system patented by Dornier in [62] makes use of a laser range finder to create “range images” of the overflown terrain. Positioning is performed by edge matching them with reference images. Bae Systems patented in 2002 a TRN system that uses a laser radar operating at a laser wavelength of 10.59 microns. The device is described to have “moving means [...] to scan the terrain around the vehicle and detecting means to detect impingement of the radiation on the terrain and measure the range from the vehicle thereto”. The invention uses a sequential processing algorithm to calculate position [63]. Ohio University began investigating on the use of ALS for aircraft navigation in 2001. During his doctoral studies, Dr. Jacob Campbell explored and demonstrated the use of an ALS-based TRN system for precision approach aircraft guiding [13], [15]. The system, entitled TERRAIN - Aided Inertial navigator (TERRAIN) uses a batch processing technique to search for the highest level of agreement between ALS data (integrated with an IMU) and a high resolution DEM (1m post-spacing and 30 cm RMS accuracy). Exhaustive and gradient-based methods were investigated as part of the batch processing techniques, which calculates the most likely user position within a spatial search area using as metric the absolute disparity. Flight tests with a proto-type TERRAIN system were performed in December 2004 and January 2005. The reported accuracy was of 0.96m mean error and 1.96m STD in horizontal direction and 0.65m mean error and 0.62m STD in vertical direction. The system was patented in 2011. A. Vadlamani received his PhD title from Ohio University in 2010 with his dissertation on the use of ALS technology for navigation in unknown environments. During flight, an aircraft performs dual ALS measurements that are used to generate overlapping terrain models, which are then used to estimate the INS velocity and position errors and constrain its drift [16]. Although it has elements of conventional TRN, the system does not use an elevation database as the “in-flight mapping capability” generates the required terrain data for estimating the bias in the INS velocity.

2.4 The unused potential of TRN

2.4.1 Opportunities

To be able to determine whether and how future improvements are possible, one must understand past and existing systems in terms of both architecture and performance. Batch processing TRN methods were the first to be introduced.

TERCOM was developed throughout the 60s and at that time digital computer capabilities were limited. In particular, the memory required for the storage of terrain maps, the possibilities of creating these maps and the processing power to sustain intensive computational algorithms were constraints that influenced TERCOM's implementation. As a consequence, TERCOM applications provide position fixes and use the MAD algorithm, which imposes a modest computational requirement on a processor. TERCOM does not work over all types of terrain and the understanding of what makes a reference map reliable was a cornerstone in the development process. It was precisely the judicious terrain selection (which came at the expense of an extensive planning process) that proved to be the key to a reliable employment. Within the past years the computational, memory access and data storage capabilities have dramatically increased. Later on, advances in the technology of remote sensing and radar altimeters enabled the development of more accurate batch methods for TRN: with the PTAN system, TERCOM's concept has been extended to a better resolution (and hence better accuracy). The potential of laser technology for TRN has also been explored, with promising results. Using ALS measurements, small resolution reference maps and the simple MAD algorithm, the TRN system described in [15] obtained position estimates with accuracies in the order of 1m. Sequential processing methods have a different approach, operating on individual elevation measurements. The Kalman filter takes into account inertial and terrain data, error model propagation and prior information to estimate the state of the vehicle. Generally, sequential TRN has divergence problems because the local approximation schemes fail to model the (highly non-linear) terrain accurately. At the time it was introduced, SITAN's approach was superior to TERCOM in terms of terminal accuracy, computational requirements, heading and velocity error sensitivity. However, terrain navigation is fundamentally a nonlinear problem and, as illustrated in the present chapter, this remains the main limitation for all recursive TRN. **Whereas sequential TRN is faced with a rather conceptual limitation, it is the batch processing methods that can make the more complete use of the latest technological developments. In fact, most of the previously assumed constraints for batch TRN are no longer applicable given the technological possibilities of today: the current DSP capabilities, the increase in real-time processing power, the availability of compact yet very large storage capabilities or the developments in the field of reference data and terrain sensing methods. The potential for TRN improvement, made possible by all these enablers, is certainly not used to the full extent.** This observation led to the definition of a research project that specifically aims to explore how to benefit from all the technological enablers in the field of terrain navigation. The approach taken is based on the conventional batch processing method. A 'batch' TRN signal is formed from separate, independent elevation measurements. The signal can be considered two

dimensional. The vertical component is generally provided by a barometric sensor. Starting at a known altitude (such as performing a reset of the sensor at the runaway) can limit the errors in the vertical channel up to an extent that will allow us to reduce the TRN problem to two dimensions. Older conventional batch systems would tune the measuring tool to sample the terrain elevation with a sample period adjusted to the reference map, so that the distance between the signal's samples will match the database resolution. This resulted in severe restrictions on manoeuvring during the data collection process. Later on, such a request became unrealistic and TRN systems began incorporating heading and speed information. Predicted flight path coordinates are retrieved from other navigation systems, as IMU or Doppler type systems. Further on, the sensed vertical profile is correlated with extracted DEM profiles that are obtained by offsetting the predicted flight path with different incremented values. Although processing capabilities have significantly improved, 'traditional' correlation algorithms may be inefficient to be performed in real-time, especially in case of large sets of data (as it would be the case for laser measurements). Within the present research, the idea of integrating additional information in a batch system was further explored. Rather than acting as a position fixing system, terrain correlation is used to track and validate the changes that take place in the two parameters: speed and heading. This approach also enables for the position estimates to be provided continuously.

2.4.2 Limitations

Main limits to TRN performance are database resolution (and accuracy), clearance height sensor performance and terrain itself. With respect to the first two, research has promised for years to produce tools that measure or map terrain accurately, with higher resolutions and at a lower cost. The efforts are already starting to pay off and these technologies will eventually mature. On the other hand, terrain is a nature-given and may be seen as the 'fundamental limitation'. Options to deal with it will largely depend on the context. If the host vehicle has the freedom of manoeuvre, it can deliberately change its course and move to an area with higher terrain signal content when the quality of the position estimate has dropped under a set threshold. Unfortunately, this might not always be an option. Covert military applications or operation in a crowded environment may put constraints on free motion. Alternatively, based on the detection of a reduction in terrain signal content, the TRN system may adapt its operation by processing inertial data differently, varying the tuning of design parameters or incorporating other sources of information.

2.4.3 Preliminary design concepts

Based on the discussions in this chapter, the preliminary design concepts of the proposed TRN algorithm can be listed as:

- From the perspective of how the terrain measurements are being processed, the chosen approach is a **batch-processing/correlation-based algorithm**.
- Instead of the conventional position-fixing approach, the proposed algorithm seeks to **track the speed and heading of the host vehicle** using terrain features (this implies that parameters must first be acquired); position is estimated in a dead-reckoning manner.
- A **context-adaptive system**: the proposed system detects the operating context and reconfigures the algorithm according to it.

Given all the above considerations, a preliminary diagram system is depicted in figure 2.6.

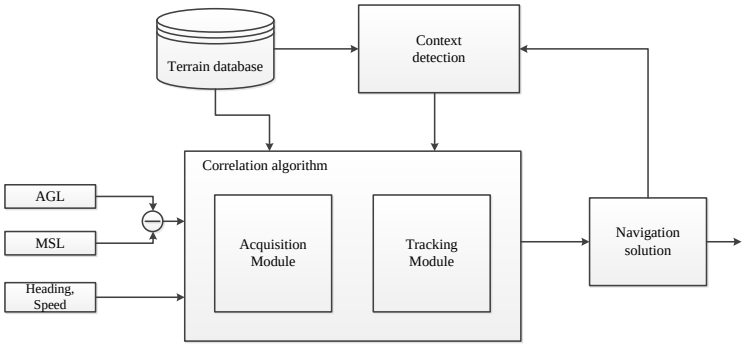


Figure 2.6: Preliminary diagram system of the proposed TRN system

Sensor measurements are used in the composition of the batch TRN signal. Meanwhile the terrain measurements form the received signal per se, speed and heading information are necessary to determine the relative positions of the consecutive samples. A wide variety of heading and velocity sensors exist. Each different sensor technology has its own attributes, its associated strengths and deficiencies. Additionally, the performance and cost of a sensor within the same 'family' will vary by several orders of magnitude. When designing a system that needs to combine information coming from different types of sensors, it is important to account for their characteristics. Additionally, the properties of the

sensors that measure the distance toward the terrain will impact/limit the maximum correlation that can be achieved with the equivalent set of samples in the terrain elevation database. The goal of the research is to improve TRN system performance through better signal processing. Still, the performance of the TRN system will depend on and be limited by the performance of the speed, heading and terrain distance sensors. As it will be illustrated in the next chapter, the specification of several configuration parameters of the tracking loops require information about the (expected) performance of the sensors. An overview of the different types of sensors and their associated properties can be found in Appendix A. The information presented in this appendix should be used as a background for Chapter 4 which will discuss the implications of sensor integration within the TRN algorithm.

2.5 Summary and conclusions

- TRN techniques determine position by comparing measured terrain features with a stored elevation map.
- The present DSP capabilities, the increase in real-time processing power, the availability of compact, large storage capabilities, the existence of hi-resolution elevation maps and the developments in the area of terrain sensors are important enablers that warrant a re-investigation of the current potential of the TRN technology.
- Whereas sequential TRN is faced with a rather conceptual limitation, it is the batch processing methods that can make the more complete use of the latest technological developments. There is ample potential to benefit from the advances developed in related signal processing applications.
- Traditionally batch TRN techniques have been used for position aiding because of the constraints in storage capabilities and processing power. Modifications need to be made to allow a terrain system to handle an increased amount of data and to provide continuous navigation.
- Terrain features represent a fundamental limitation for any TRN system. It is important to be aware of the 'quality' of information available in the surroundings and how it can be used to obtain the best possible navigation solution. This implies detecting the operational context and adapting the algorithm according to it.
- Based on the identified opportunities and limitations, a blueprint for a 'new' TRN algorithm was presented.

Chapter 3

Exploring a GPS inspired "Acquisition & Tracking" concept for TRN

3.1 Introduction

Technically, GPS and TRN seem very different. GPS relies on the measurement of signal propagation time and information on the location of the transmitters to estimate the location of the receiver. The estimation of position using TRN relies on finding a match between a measured set of terrain elevations with a similar profile in a database containing elevations of the terrain at known locations. In spite of these seemingly technical differences, some of the processes that take place in the receiver are quite similar. With GPS, a computer in the receiver uses a set of pre-defined codes to compare them with the codes modulated on the incoming signal. During a first phase, acquisition, the receiver finds the correct code-phase. During the second phase, tracking, the receiver maintains the code-phase, as the signal properties change over time. Similar to code acquisition and tracking in GPS, the TRN system needs to find and track a particular sequence of measurements in a larger dataset. Both with GPS and TRN, uncertainty is present with respect to the actual spacing between the samples of the received signal. With GPS this happens because the transit time and Doppler shift are changing parameters in time due to the movement of the satellites and of the receiver itself. With TRN uncertainty in speed and heading, while performing the measurements, yields the uncertainty in spacing between samples. Both with GPS and TRN, the correlation of the measured signal with a local replica is

used to solve for this unknown shift. In GPS, once the two parameters are determined at a coarse scale it is possible to keep track of them in an elegant way, because they only change slightly during each measurement. In a TRN system, the heading and speed are also expected not to change heavily during a dense set of measurements. With GPS, the sub-chip displacement of the local code needed to find and track the correlation maximum can be obtained from the difference between the output of an early and a late correlation channel. In this chapter it will be illustrated that the early-late tracking concept can also be applied for TRN. The chapter will start with a brief description of the GPS signal, as well as of the digital processing stages that take place in a GPS receiver. In this way, the necessary concepts to understand the similarities between GPS signal processing and TRN will be reminded. Beyond the similarities, there are also important differences between the ideal PRN signal used with GPS and the random terrain signal. The differences and their impact will be discussed and analysed. Further on, it will be shown how the GPS code “acquisition & tracking” concept was adapted to TRN, resulting in the design of an adaptive early-late based tracking algorithm.

3.2 GPS revisited

3.2.1 GPS signal

The GPS signal consists of the following three components:

- Carrier:
  Represented by a RF sinusoidal signal with frequency fL1 = 1575,42 MHz or fL2 = 1227,60 MHz;

- Ranging code/ Spreading sequence:
  Each satellite can be identified by two unique ranging codes: the freely available Coarse-Acquisition (C/A) code and the restricted (usually reserved for military applications) and encrypted Precision (P) code. They are pseudo-random noise(PRN) sequences. The C/A code consists of a 1023-chip sequence, has a period of 1 ms and a chipping rate of 1.023 MHz . The P code has a 10.23 MHz chipping rate and a one-week period.

- Navigation data:
  A GPS satellite transmits a coded message consisting of data on its status, ephemeris and clock bias parameters. The navigation data has a length of 12.5 minutes and is transmitted at 50 bits per second.

At the transmitter side, the following processes take place: the binary navigation data is combined with the ranging code using modulo-2 addition; the composite digital

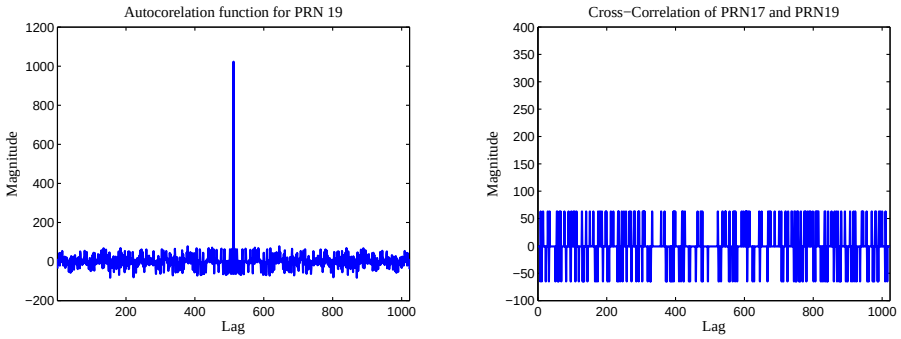
signal is then modulated onto the carrier, using binary phase shift keying (BPSK) modulation. Further details on how the GPS signal is formed can be found in [64,65]. As pointed out, our attention has been mainly oriented towards the spreading codes and the way their properties are keyed in the GPS digital signal processing stages. Before going into details, a description of these properties is given in the following section. For simplicity, we will restrict ourselves to the C/A code.

3.2.2 C/A code

The PRN sequences used as C/A codes belong to a unique family of sequences referred to as Gold codes. For details on the generation of a Gold sequence, see [64,66]. Gold codes (and hence C/A codes) are characterized by important correlation properties:

- Low cross-correlation: all C/A codes are nearly uncorrelated with each other and their cross-correlation can take only three different values;
- Small autocorrelation side-lobes: all C/A codes are nearly uncorrelated with themselves, except for zero lag where the autocorrelation function (ACF) has a peak in magnitude (of 1023, equal to the code length);

Figure 3.1 illustrates an example of cross- and auto- correlation properties of the C/A code.



(a) (b)

Figure 3.1: Correlation properties of the C/A code: (a)autocorrelation of the C/A code for PRN 19, (b)crosscorrelation between C/A codes for PRN 17 and 19.

3.2.3 GPS receiver operation overview

The received signal is a combination of signals from all visible satellites. When acquiring a satellite, the incoming signal is multiplied with the local generated C/A

code corresponding to that specific satellite. Due to the cross-correlation properties, the signals from other satellites are nearly removed during this procedure. To assure that the desired signal is not removed too, the locally generated code must be properly aligned in time with the incoming code. Once the spreading sequence was removed, the signal is demodulated by mixing it with a locally generated carrier wave. In order to wipe off the carrier, the frequency of the locally generated signal must be close to the signal carrier frequency. Thus, two parameters are needed: the frequency and code phase of the incoming signal. The motion of the transmitter (i.e. the satellite) relative to the GPS receiver causes the Doppler shift: a difference between the frequency of the received signal and the frequency at the source. In [64] is said that for a stationary receiver the shift is around ±5kHz, meanwhile for a GPS receiver moving at high speed the frequency can deviate up to ±10kHz. 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 offset are estimated through a two-step process: acquisition and tracking. The purpose of acquisition is to identify all satellites visible to the user and, for each satellite signal, to determine coarse values of the carrier frequency and code phase. The purpose of tracking is to refine the values of the two parameters and keep track of them, as the signal properties are a function of time.

3.2.3.1 Acquisition

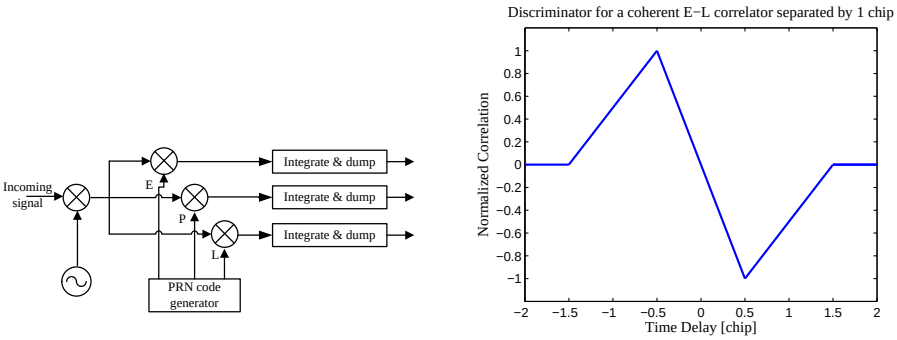
During acquisition, a GPS receiver must conduct a two-dimensional search in order to find each satellite signal, where the two dimensions are C/A code delay and carrier frequency. For each searched frequency, the receiver generates the same PRN code, moves the delay of the code in discrete steps and calculates the correlation between this locally generated signal and the incoming one. Code alignment (and match in Doppler) is indicated by the appearance of a high magnitude peak in the ACF. Acquisition can be implemented as a: serial search, parallel frequency space search and parallel code phase search. Details on the implementations can be found in [64].

3.2.3.2 Tracking

After acquisition, the receiver generated reference C/A code will be in approximate alignment with that of the signal (usually half a chip) and the frequency of the signal will be known to within the frequency bin width. Unless further measures are taken, the Doppler shift will cause the received and reference codes to drift out of alignment and the signal frequency to differ from the acquired frequency. The tracking module ensures that both accurate alignment with the received code and accurate tuning to the signal carrier are maintained. 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 tracking of the C/A code component of the GPS signal is accomplished by means of a Delay Locked Loop(DLL). The DLL is a device that permits to generate local code references in the receiver, correlate them with the incoming signal and estimate (and minimize) the time delay [67]. In its most basic implementation, the DLL is based on a two correlator structure. Figure 3.2(a) illustrates a generic code tracking loop diagram. Two replicas of the received signal are generated: one slightly early, the other slightly late, with the correlator spacing being the fixed time between them. If the replica code is aligned, then the early and late envelopes are equal in amplitude and no error signal is generated by the discriminator. If the replica code is misaligned, then the early and late envelopes are unequal by an amount proportional to the code phase error. The discriminator for a coherent early minus late correlator with a 1-chip spacing between the early and late signals is illustrated in figure 3.2(b). Besides



(a) (b)

Figure 3.2: (a)Basic GPS code tracking loop block diagram using a DLL, (b)DLL discriminator function.

the early minus late version, other discriminators exist. Some might use all three correlation values to form the correction signal. However, the functioning principle is essentially the same. Figure 3.3 shows how the early, late and prompt envelopes change as the phases of the code replica are advanced with respect to the incoming signal. For ease of illustrating the underlying rationale, only 1 chip of the C/A code is shown and the incoming signal is considered to be without noise. At the left the incoming code was shifted in time, resulting in a maximum correlation for the late replica. In the middle figure, the incoming code remained unchanged in time, resulting

in a maximum correlation for the prompt replica; meanwhile at the right the prompt replica is 1/4 chip late.

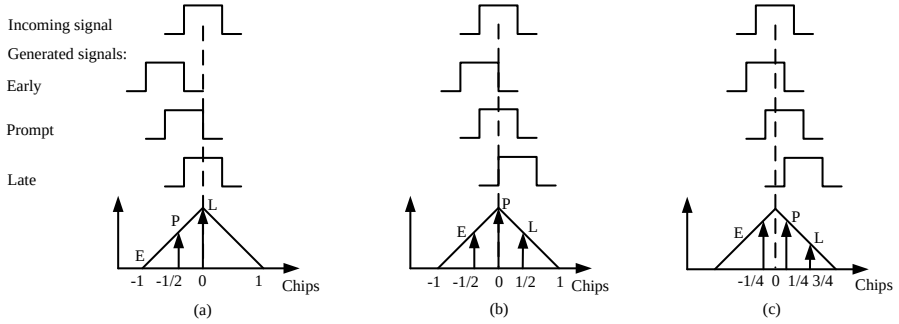


Figure 3.3: Code tracking: early, prompt and late replicas are generated and correlated with the incoming signal

The early-late tracking loop performs well for GPS, due to the correlation properties of the C/A codes. For general signals (codes) there is no guarantee it performs equivalently well. However, in the following sections it will be illustrated that this type of feedback loop is of interest in case of terrain elevation data and can be used to keep track of the heading and speed of the host vehicle.

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 was not used in the proposed TRN algorithm.

3.3 Adapting the GPS “acquisition & tracking” to TRN

Similar to the acquisition and tracking of frequency and code phase in the GPS receiver, the basic rationale for the proposed TRN algorithm is to use terrain correlation to “acquire and track” the speed and heading of the host vehicle, while the position advances are calculated using these estimates together with the previously determined positions. In accordance to the GPS receiver approach, a two-stage TRN correlation algorithm was implemented. Figure 3.4 gives an overview of the building blocks that make up the algorithm. Figure 3.5 gives a zoom-in view of the tracking module. The arrows in the figures indicate how these blocks are connected to each other and which input information they require.

During acquisition coarse estimates for speed and heading are found. Next, the system switches to a tracking mode, during which the changes in the parameters are being monitored. Hence, the acquisition phase is used to initialize the system,

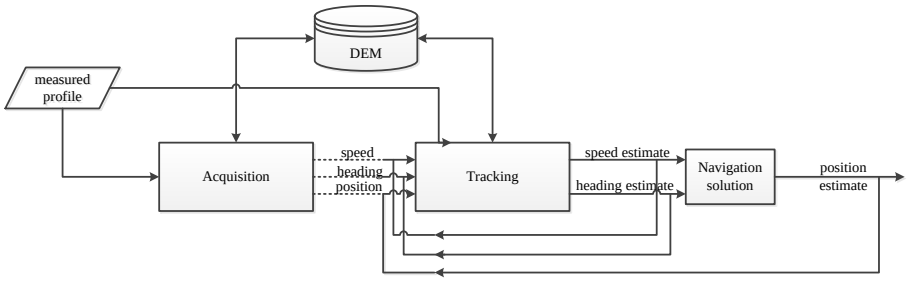


Figure 3.4: Block diagram of the proposed TRN system

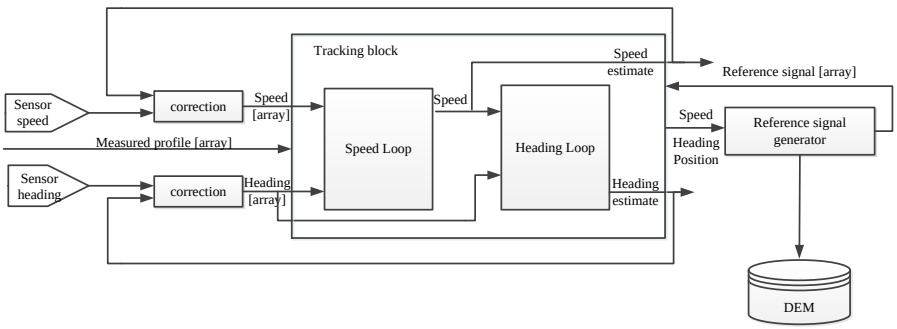


Figure 3.5: Block diagram of the TRN tracking module

meanwhile tracking is the main operating mode for providing navigation data. Tracking is performed sequentially, with two separate tracking loops: one to follow changes in speed, one to follow changes in heading. More details on the implementation and functionality of the acquisition and tracking blocks are provided in the different sections of this chapter.

In order to assess the validity of the proposed concept for terrain navigation, a simulation tool has been implemented. All discussions in this chapter are supported by simulation-based graphs and figures. As this chapter is intended to serve as a proof-of-concept, the scenarios investigated have been simplified. No errors have been simulated for the terrain measurements. Also, as illustrated in figure 3.5, speed and heading sensor data is incorporated into the tracking module. However, for

simulations used in the present chapter it was considered that the vehicle is travelling at constant speed and heading, so no sensor information was required. The aspect of both primary and supportive sensor data integration will be discussed in chapter 4.

3.3.1 The TRN signal

The received TRN signal consists of a batch of terrain heights, as measured by the terrain clearance height sensor. The TRN signal will also be referred to as: measured profile or transect. The length of signal is variable and is denoted as N. The distance between two consecutive samples of the TRN signal is a function of the speed of the vehicle and the time between measurements. The direction between two consecutive samples is a function of the heading of the vehicle.

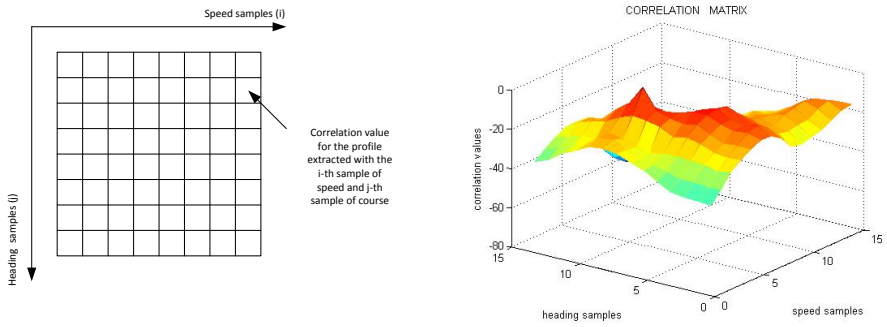
3.3.2 Acquisition process

Acquisition is implemented as a serial search process of the two parameters: speed and heading. The acquisition module must be initialized with an approximate position solution in order to determine which region of the database to search. The start position estimate becomes available out of another navigation system, such as INS or GPS. Once the location of the start position estimate is determined, an uncertainty area around it is created. The result is a square 'search area' with the start position estimate as its center and the borders situated at a distance of multiple grid points (equal to multiple of database resolution). The size of the search area depends on the accuracy of the navigation system which provides the start position estimate.

Acquisition must be performed at constant speed and heading. The uncertainty interval and the width of the search bin for both parameters may be chosen arbitrarily. If any information is available from independent sources, the search interval can be narrowed down significantly. For each speed value from the uncertainty interval, different values for the heading are considered and a profile using these values is extracted and correlated with the measured profile. When all possibilities have been verified, a correlation matrix is build and plotted, as illustrated in figure 3.6(a) . The estimates that maximize the correlation are picked up and delivered to the tracking module, together with the position estimate. Due to the randomness of terrain, cross-correlation between tracks may have a dynamic range. The correlation plot will show a noticeable (auto-correlation) peak whenever the tracks are matched, as seen in figure 3.6(b).

The serial search algorithm performs two different sweeps and will repeat the process for all points in the search area. All in all, this results into a total of (3.1) combinations.

$$\left(\frac{uncertainty\_interval\_speed}{search\_bin\_width}+1\right)\cdot\left(\frac{uncertainty\_interval\_heading}{search\_bin\_width}+1\right)\cdot[starting\_position] \quad (3.1)$$



(a) (b)

Figure 3.6: Acquisition correlation matrix: (a)Theoretical representation, (b)Simulation output from a randomly selected TRN acquisition process.

Obviously, this may lead to a large number of combinations and thus, to a time consuming procedure. To minimize ambiguity (appearance of additional peaks in the correlation matrix) it is recommended to choose a larger value for the length of the transect N. This will increase even further the acquisition time. If the starting position is accurately determined, the performance of the acquisition procedure can increase significantly.

3.3.3 TRN correlation functions

The ‘correlation’ method used in the algorithm does not compute the correlation in a strict statistical sense, but uses a quality function that indicates the similarity of the terrain height measurements with the profiles extracted from the map. **Similar to the GPS code correlation function, the TRN correlation function must provide a measurement of the alignment between the signal (the measured terrain profile) and the reference codes (database extracted terrain profiles)**. The investigated matching functions are listed below.

In (3.2),(3.3),(3.4):
N = number of measurements in the TRN profile
d = measured height at the i-th instance
$\overline{d}$ = average of d (sum of d divided by N)
h = map height in the location computed for the i-th instance
$\overline{h}$ = average of h (sum of h divided by N)

- The Pearson product-moment correlation coefficient

  This coefficient measures the linear dependence between two variables. Based

on samples of paired data, the coefficient is calculated as in (3.2):

$$C = \frac{\sum \left(d_i - \bar{d}\right) \cdot \left(h_i - \bar{h}\right)}{\sqrt{\sum \left(d_i - \bar{d}\right)^2} \cdot \sqrt{\sum \left(h_i - \bar{h}\right)^2}} \tag{3.2}$$

- Mean absolute difference(MAD)

  In this case the absolute distance between the measured height and the map height in the candidate points is computed and then averaged over the signal length, (3.3):

$$C = \frac{\sum |d_i - h_i|}{N} \tag{3.3}$$

- Mean squared difference

  The differences between the measurements and the map heights are calculated and then these values are squared and summed. Subsequently, the sum is divided by the number of measurements, resulting into the mean squared difference (3.4). 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 with respect to the deviation of a highly correlating track, which will be close to zero.

$$C = \frac{\sum \left(d_i - h_i\right)^2}{N} \cdot \sqrt{\frac{1}{N-1} \cdot \sum \left(\left(d_i - h_i\right) - \frac{\sum \left(d_i - h_i\right)^2}{N} \right)} \tag{3.4}$$

The term 'TRN correlation function' has already been encountered. Hereinafter, it will be explained how the function is actually computed. We define three types of functions: speed correlation functions(SCF), heading correlation functions(HCF) and terrain autocorrelation functions. When the searched parameters are speed and heading, it is normal that the 'correlation function' is calculated as a function of the interest parameter. Hence, SCF and HCF is a representation of how the correlation between profiles varies when the only modified parameter is speed, respectively heading. Essentially, this comes down to correlating the prompt version with a series of other profiles extracted from the DEM. These profiles are extracted using the same values for the parameters as in the prompt version's case, except for speed/heading. Speed/Heading will vary in an arithmetic progression: positive (the speed/heading values will grow to the right and we will extract in this way the late profiles) and negative (the speed/heading values will decrease to the left and we will extract in this way the early profiles). The numerical outputs computed by correlating each one of these profiles with the prompt version will form the SCF/HCF.

SCFs and HCFs need to be calculated real-time, as they require parameters that change as a function of travel (such as speed, heading, time between measurements). Instead, the terrain ACF is a more generic function. The terrain profile for which the ACF is studied is built using only database resolution points. This profile is then correlated with equal length terrain profiles (formed only from database resolution grid points) extracted using the same heading and varying only the starting point. Terrain ACF provides a qualitative description of the terrain signal, meanwhile SCF and HCF are used in the tracking algorithm.

In order to decide which one of the proposed metrics will be implemented in the algorithm, both speed and heading correlation functions for a large number of profiles were analyzed. Figure 3.7 illustrates randomly chosen examples of terrain ACF, SCF and HCF when the different functions proposed above were used as ‘correlation’ methods(for comparison sake, functions have been rescaled or reversed).

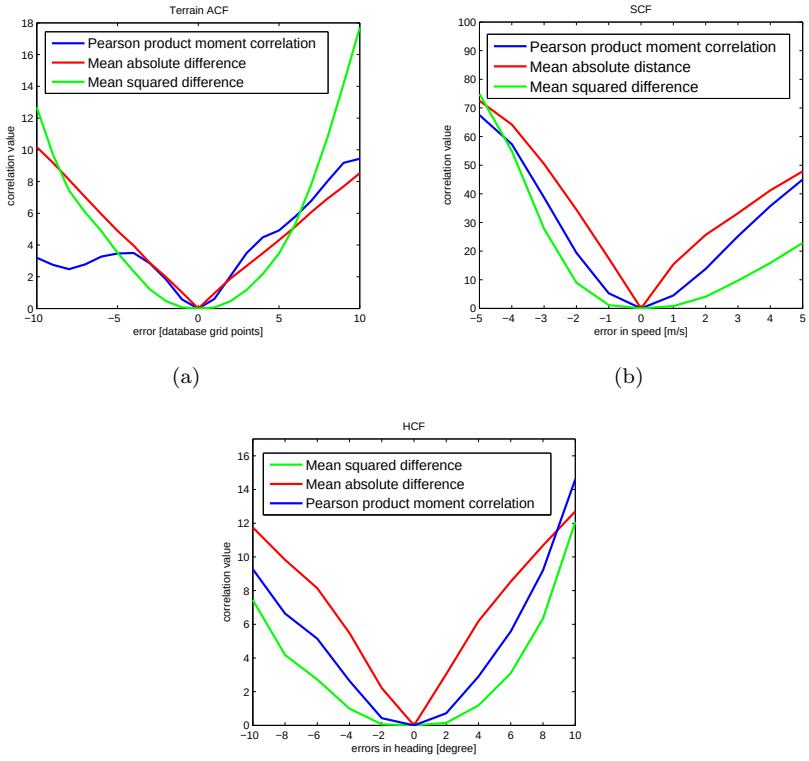


Figure 3.7: Different metrics used in the compution of the (a)Terrain ACF , (b)SCF , (c)HCF.

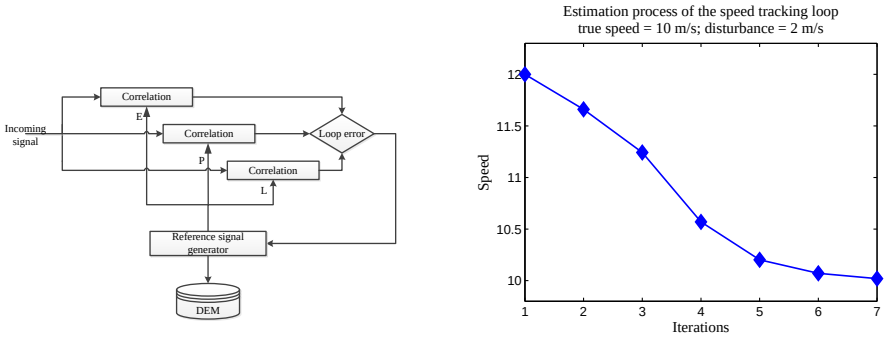
The formula chosen for implementation in the algorithm was the MAD function because it is the most discriminative in the interest area and tends to have a quasi-symmetrical triangle shape.

3.3.4 Tracking process

Tracking is performed sequentially using two separate tracking loops, as illustrated in figure 3.5. Both trackers have been designed following the DLL functional principle. As the rationale behind the development is the same, only the speed tracking loop will be described in the present section.

The TRN tracking loop has been designed as an early-late correlator, as illustrated in figure 3.8(a) . The incoming signal consists of a fixed length measured profile. Thus, as a new measurement is added, the first measurement from the transect is discarded. Three local replicas are generated: the prompt replica of the profile is extracted using the exact estimated values of the parameters; the early and late replicas are profiles extracted from the database using a small offset s (i.e. correlator spacing) relative to the estimated(prompt) speed. The output (C_L, C_E, C_P) is a numerical value indicating how much the specific signal replica correlates with the ‘incoming signal’. The correlator will send feedback to the “reference signal generator” if the speed needs to be adjusted. The error signal, ϵ is provided from the combination of the correlation outputs and the exact computational formula is given in (3.5).

$$\epsilon = s \cdot \frac{H_E - H_L}{H_E + H_L} \tag{3.5}$$



(a) (b)

Figure 3.8: (a)Block diagram of a TRN tracking loop, (b)Example of the iterations that take place during an estimation process of the speed.

Figure 3.8(b) shows the output of the speed tracking loop for a randomly chosen

example. Unlike in the GPS case, the correction is not immediate and several iterations may be required before the tracker finds lock. The spacing between the early, prompt and late replicas plays an important role in this process. If the spacing is large, the TRN tracker would be able to handle larger dynamics and determine the correction faster. On the other hand, a smaller spacing would yield a more precise result. This comes as a result of the fact that terrain 'code' are random sequences and do not possess the Gold code's correlation properties. Section 3.4 will discuss the differences in signal properties between the C/A code and terrain elevation profiles and their influence on the design and configuration of the TRN correlators. Based on results from this analysis, mitigation strategies to cope with the effect of the differences are discussed in section 3.5.3.

3.4 TRN signal vs. (the code component of the) GPS signal

Major advantages of the codes used in GPS, relative to the terrain reference signal, are the properties in terms of correlation. The GPS receiver tracks the auto-correlation peak. The resulting estimate is precise because the peak is narrow, larger than any other side peaks, and the correlation measurements are very sensitive to the location of this sharp event [65]. Meanwhile, terrain 'codes' are not deterministic sequences. Empirically it has been shown the auto-correlation peak of the function (i.e. MAD function) is distinct, but cross-correlation between tracks may have a dynamic range, affecting the shape of the correlation triangle. As a consequence, the width and the height may vary. Moreover, the shape of the function is completely unique for each track.

3.4.1 Signal/Code differences

In Figure 3.9 A and B a conceptual comparison between the shapes is performed. The rest of the figure goes on to illustrate the similarities, as well as the typical differences between the two correlation concepts.

With GPS, the width W of the autocorrelation peak (A) is 2 chips, and the height is 1023. For the TRN example (B), the width W can be expressed in multiples of the spacing between the terrain data samples (e.g. W = n x pole spacing). Both width and height depend on the properties of the terrain and the amount of samples used. The maximum output value of the prompt channel can be computed using local data, if accurate information on the prompt exists (for example immediately after the acquisition phase has been completed. With GPS, the maximum practical amount that the early channel (C) can be advanced is one chip, although normally

half a chip is the preferred value. The same limit applies to the displacement of the late channel relative to the prompt one. The correction (in chip units) needed to shift the prompt channel is obtained from the discriminator or, alternatively, by dividing the current output value of the prompt channel (magenta line near A) by the slope. With TRN, the maximum output of the prompt and shifted channels is unknown and hence no slope can be determined. One approach is to assume that these values are the same as the earlier computed maximum value of the prompt channel. This yields the black line in figure 3.9F. However, the prompt might not indicate the true speed. Additionally, the shape of the early and late correlation functions is influenced by the amount of speed offset. This is illustrated by the red and green contours relative to the black one of the prompt in figure 3.9D. The blue line in figure 3.9F represents the calibration line in case this influence has been taken into account. The slope would be equal to the sum of the maximum outputs of the shifted channels divided by maximum early-late spacing. This brings us to the equation (3.5) as the formula needed to obtain the control signal. Table 3.1 summarizes the differences in signal

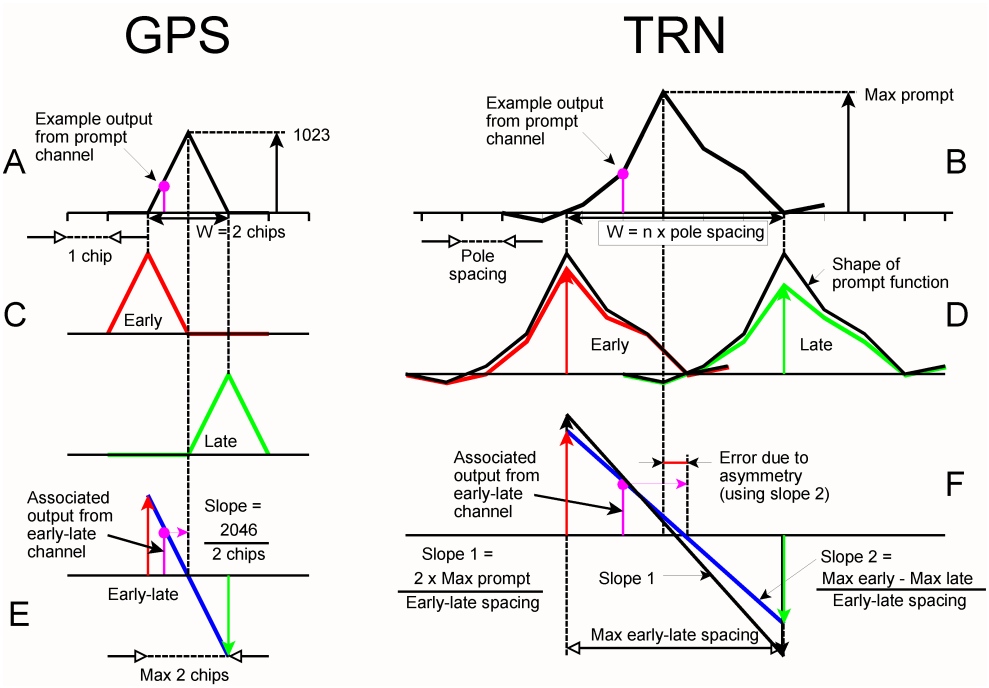


Figure 3.9: Illustration of the differences between GPS and TRN tracking concepts

properties between the TRN profiles and the GPS codes.

Table 3.1: Differences between GPS and TRN signals

Signal property	GPS	TRN
Magnitude of correlation peak	Constant, known	Variable, unknown
Width of correlation peak	Constant, known	Variable, unknown
Cross correlation values	Constant, known	Variable, unknown
Shape of correlation peak	Symmetrical	Asymmetry may appear
Early, late shape	Identic with prompt shape	Different from prompt shape; different from each other

3.4.2 Specific properties of the TRN signal

A set of specific properties that characterize the TRN signal and correlation function were identified and are described in the following paragraphs.

Bandwidth property

The bandwidth property is defined as the width of the correlation peak. Tracking can be performed only inside the correlation triangle. As soon as the V-shape is lost and non-linear behavior appears in the correlation function tracking can no longer be applied. Alternatively, bandwidth can be defined as a measure of the spacing for which the early-late algorithm can be applied. Mathematically, the bandwidth represents the space between the global minima of the correlation function and the point where the slope changes sign. The bandwidth must be balanced with respect to the peak. It can be expressed in multiples of the spacing between speed samples (hence terrain data samples). The concept is illustrated in figure 3.10.

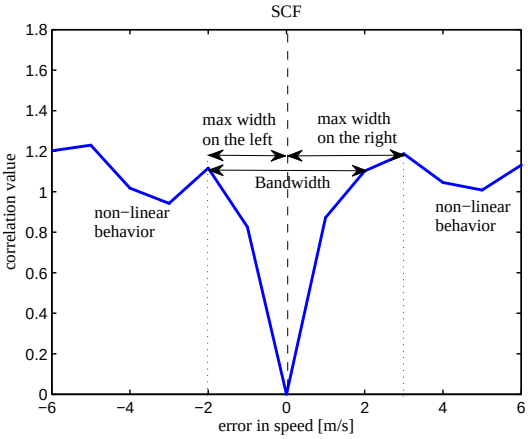


Figure 3.10: Illustration of the bandwidth property on a TRN speed correlation function

In signal processing the term ‘bandwidth’ is related to the range of frequency of the electronic signals used within a system. Though we are dealing with a different

kind of signal, the concept is essentially the same: the TRN system can only process signals within the calculated bandwidth.

Energy property

The energy property is related to the length of the transect. The more samples available for comparison, the stronger the agreement between the measured profile and the associated one that is extracted from the database. In terms of the correlation function, an increase in length triggers an augmentation of the correlation triangle's height. In fact, the signal to noise ratio of the correlation peak relative to the other correlation values improves due to the increased energy. Thus, this is a form of adjusting the strength/energy of the incoming signal. As a drawback, a longer track will always yield additional computational load. One method to gather more measurements could be to increase the sampling rate. This can become a false solution, as we are fundamentally limited by the database resolution. Once this margin is exceeded, over-sampling does not bring any new information. Hence, the best solution is to add new measurements to the track.

Symmetry property

The TRN correlation triangle would be symmetrical if it were an isosceles triangle. But to have the two halves as each other's mirror image is hardly ever the case for a TRN correlation function. This issue and its impact on the early-late algorithm will be tackled in section 3.5.3 . For the time being we are looking if asymmetry can be in any way quantified. This would be rather difficult because of the dynamic nature of the correlation function.

The slopes of the two sides can provide us with information on the asymmetry. In mathematics, the slope of a line describes its steepness. For the TRN function, the slope is computed by fitting a line through the points of the correlation function using the least squares algorithm. The resulting slopes will depend on the number of the speed samples used. This is illustrated in figure 3.11. The ratio of the slopes is a good indication of the symmetry of the correlation function. The symmetry property is used in the algorithm as a restriction on the bandwidth. This will be further explained in section 3.5.2 .

3.5 An adaptive early-late tracker

3.5.1 Problem analysis

Early-Late discriminators work on the principle that the signal power in the early and late channels is equal when the prompt reference code is synchronized with the incoming signal. This is possible because of the properties of the autocorrelation

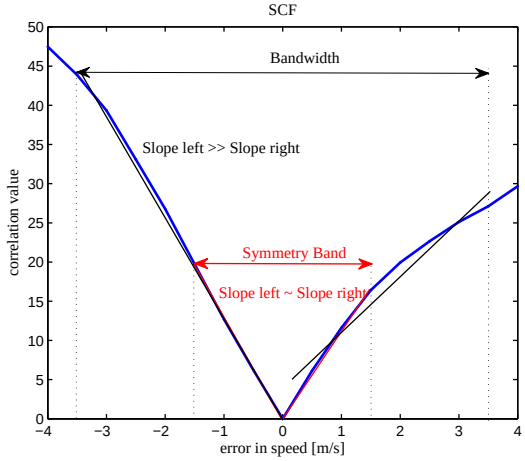


Figure 3.11: Illustration of the symmetry property on a TRN speed correlation function

function of the C/A codes. As pointed out, the TRN correlation function does not have a predefined shape, but variable. The differences between the two functions were summarized in table 3.1. In table 3.2 the impact of these differences is indicated. When applying the early-late scheme, it comes down to two main issues, as indicated in the last column of table 3.2.

Table 3.2: Impact of the TRN signal's properties on the design of an early-late based tracking loop

TRN signal property	**Impact on the E-L correlator**	**Issue**
Variable and unknown magnitude of correlation peak	No fixed upper threshold to determine when in lock	Convergence
Variable and unknown width of correlation peak	No fixed upper threshold to reject solutions	Convergence
Variable and unknown cross correlation	No standard threshold to select the spacing	Convergence
Asymmetry of correlation peak	Cause bias in the correlator output	Error bias
Non-identical early, late and prompt correlation functions	Cause bias in the correlator output	Error bias

3.5.1.1 Convergence

The convergence requirement represents the need that the correlator finds lock. With an early-late tracker it is considered that the signal is aligned with its database replica if the early and the late channels are equal and no error is generated by the discriminator. Given that the magnitudes of the TRN correlation peak and of the cross-correlation are variable, the error generated in the loop will only tend towards null value. There are two aspects of the convergence process. It is desired for the system to be **stable**, i.e. to find lock without entering an oscillating behaviour. The other aspect relates to the running time. It is desired for the correlator to have **fast**

convergence time, i.e. to find lock using a minimum amount of iterations.

3.5.1.2 Error bias

The used discriminator is designed to keep the power of the early and late correlators equal. The TRN correlation function might be skewed and asymmetric and a distorted correlation function is bound to bias the process. The changes in the early and late shapes relative to the prompt have a similar impact, adding to the error bias.

3.5.1.3 Parameters

Whereas for GPS several design parameters of the DLL can be constant (and easily assigned), for the TRN early-late correlator becomes rather difficult to decide standard pre-determined values for the following parameters:

- Correlator spacing
  Represents the difference between the prompt and the early/late replicas;
- Transect length
  Represents the length of the signal (i.e. number of samples taken into consideration);
- Correlator threshold
  Represents the threshold for which it is considered that the correlator has found lock (closely linked to the number of iterations needed to find lock).

Obviously, an early-late tracking scheme for TRN cannot be implemented straightforward. In order to cope with the issues discussed above, mitigation strategies are required.

3.5.2 Mitigation strategies

3.5.2.1 Convergence analysis

In order to converge to a solution, it will be considered that the TRN tracking loop is in lock when the difference between two consecutive estimates reaches a certain threshold. Thus, the ‘null seeking’ strategy found in the common DLL is replaced with the constraint that the early-late difference only tends to a certain/null value. Fast convergence can be achieved using an optimal early-late correlator spacing. If referring to the ‘spacing optimization’ issue, as spacing is an elective parameter, main questions that need to be tackled are “how exactly does the spacing influence convergence?” and “which are the upper and the lower limit of the choice interval ?”. Although there is no absolute maximum, the upper threshold when choosing the spacing can be calculated

from the bandwidth property. As explained in section 3.4.2, the early-late concept can be applied only while the correlation function is V-shaped. Hence, the maximum spacing must not exceed the bandwidth property. A wide spacing can assure a better capability of rejection because the correlator can handle larger dynamics and the bandwidth within which speed disturbances can be compensated for is maximized. Theoretically, there is no minimum limit when choosing the spacing. However, if the spacing between early and late is too narrow (i.e. the difference between prompt and early is smaller than the error that needs to be corrected) then both early and late replicas will be on the same slope. The correlator has insufficient bandwidth to follow the changes in speed. The correction will be too small and the desired solution will not be reached. This situation is illustrated in figure 3.12. Prompt versions are in blue; meanwhile early are in red and late in green. The chosen spacing is smaller than the introduced error and, in the initial state, all channels are situated on the same side. The correction value is too small, so in a first iteration the solution is overshot. However, the prompt channel is now within a distance from the true speed that is smaller than the spacing. During the second iteration the correction will lead to convergence. Instead, a wide spacing (which results in sufficient bandwidth to follow the change in speed) will put the correlator in an immediate lock. Figure 3.13 illustrates this, for the same test case as in figure 3.12 .

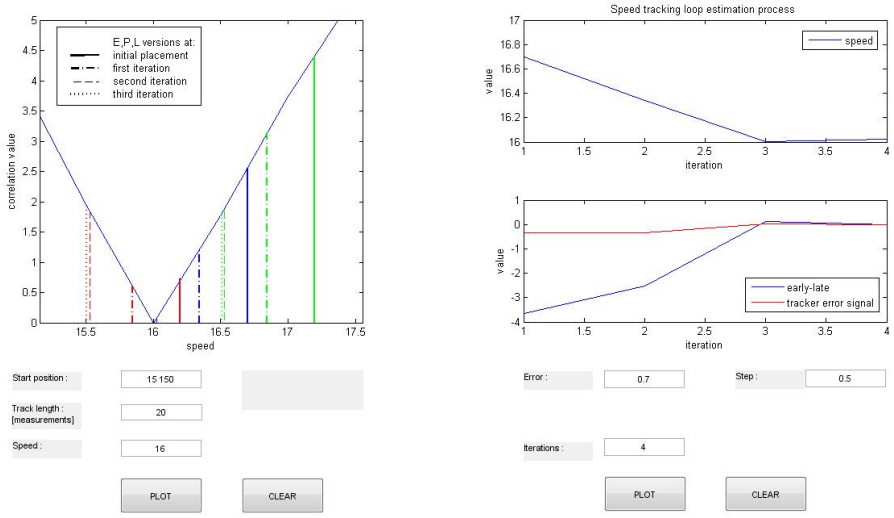


Figure 3.12: Visualization of the tracking algorithm estimation process: in the initial state all channels are situated on the same slope. The first iteration will move the prompt closer, but not yet within the early-late window. Solution converges after the second iteration.

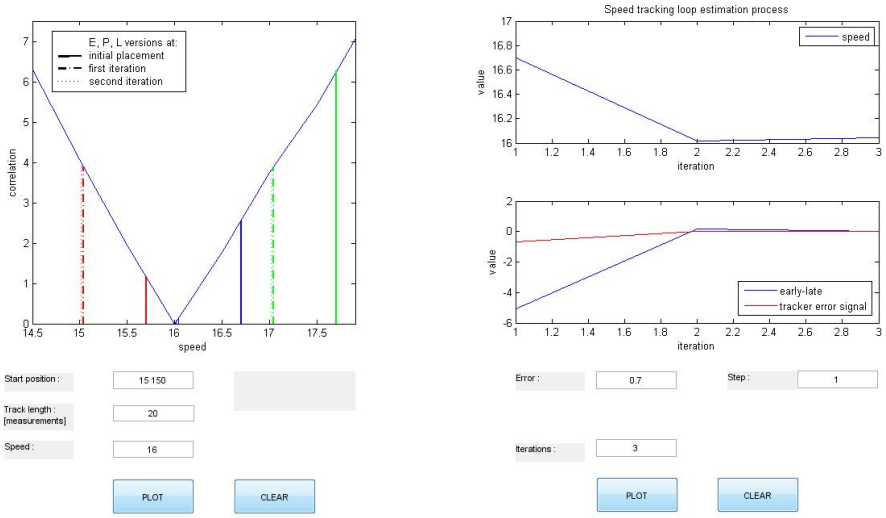


Figure 3.13: Visualization of the tracking algorithm estimation process: correction is immediate, from the first iteration

In conclusion, the convergence problem can be reduced to a sole parameter: correlator spacing. So if one would ask “What is the optimal value for the spacing?” the answer would be that a wide value (as wide as possible) is desired in order to reach fast convergence. However, a balance must also be achieved between the time the algorithm takes until reaching a stable state and the performance that this setting provides for the outcome. The wider the spacing, the further (from the prompt) the shifted values will be placed. And the further from the prompt, the more skewed the correlation function might become. A symmetry analysis, as discussed in section 3.4.2, is needed. An extra threshold is introduced that will limit the ratio of the slopes of the correlation function. This means that when the bandwidth property is calculated, a symmetry analysis is also performed. If this condition is not satisfied, the maximum limit for the spacing is decreased and the analysis is repeated. The process of assigning the spacing is explained with a flowchart in figure 3.14.

Convergence will be also influenced (although to a lesser extent) by the correlator threshold parameter. The correlator threshold dictates the difference between consecutive iterations at which it is decided that convergence has been achieved. The correlator threshold should also be elected depending on the bandwidth of the CF. A narrow bandwidth means tracking will be performed in the proximity of the peak where symmetry is greater, thus the correlator threshold can take a smaller value. If the CF has a wide bandwidth, the correlator spacing will also take a large value and

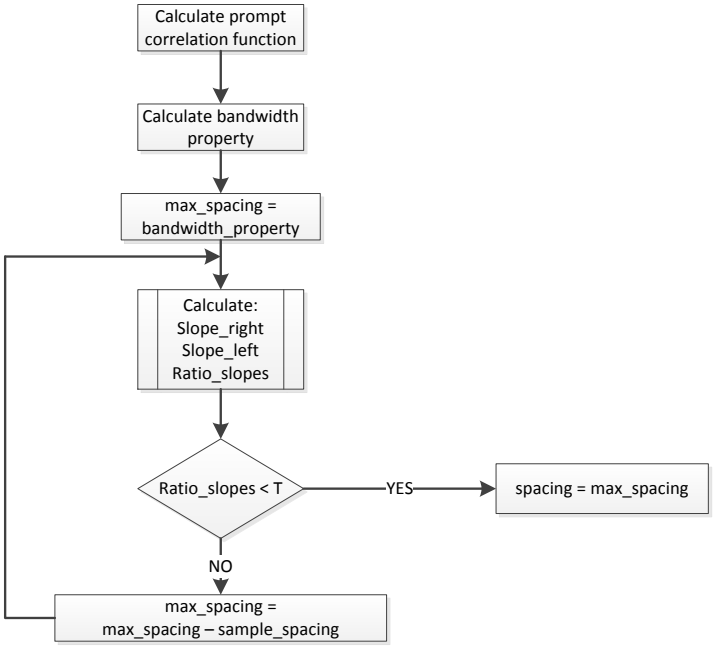


Figure 3.14: Flowchart describing the process of assigning the correlator spacing

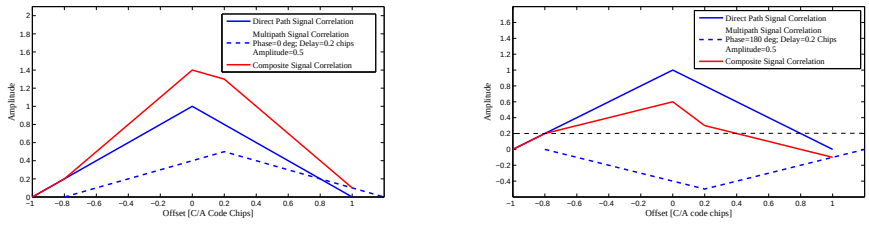
the stopping condition can be more relaxed. Alternatively, the correlator threshold can take a fixed (average) value and the algorithm can be automatically stopped if the condition is not reached within a certain amount of iterations (especially if the application has running time constraints).

3.5.2.2 Minimizing error bias

As pointed out, the distorted TRN correlation function will cause a bias in the solution. We have investigated if any similar problems were found for GPS code tracking. Although at first such an approach may seem strange because of the earlier mentioned differences in signal properties, this is because the analysis was partly based on the 'ideal' GPS code signal (i.e. as being used to modulate the carrier). In reality, the received signal may suffer from disturbances such as multipath and interference which also influence the shape of the autocorrelation triangle. In the presence of multipath signals the correlation function is distorted, becoming skewed and non-symmetric. Hence, GPS multipath mitigation strategies based on correlation techniques may also be useful (in a modified form) to effectively deal with an asymmetry in the correlation function of the TRN tracking loop.

Receiver internal correlation techniques for minimizing code multipath error in GPS

Out of the different error sources associated with GPS signal processing, multipath directly affects the code tracking process. In case of multipath propagation, the received signal is a distorted version of the transmitted one. In addition to the direct signal, the receiver observes other replicas propagated via longer paths, due to interactions with one or more obstacles in the environment. Reception of multipath can cause significant distortions to the shape of the correlation function. Figure 3.15 illustrates this effect for a reflected and direct signal that are (a)in-phase and (b)out of phase. In both cases the effect of multipath results in a skewed and asymmetric C/A code correlator, with unequal slopes of the lines on either sides of the peak. This shows similarity to the natural shape of the TRN correlator, as seen in figure 3.7. Given this argument, several algorithms used in GPS for minimizing code multipath errors were reviewed for a potential adaptation to the TRN case.



(a) (b)

Figure 3.15: (a)Constructive and (b)destructive multipath interference

Narrow Correlator Spacing Technique (NCS)

Historically, conventional GPS receivers have used a 1.0 chip early-late correlator spacing in the implementation of DLLs. The advantage of choosing a wide spacing was mostly related to having a wide bandwidth to better cope with disturbances. It was in the early 90s that the issue of tracking with a much narrower spacing was investigated. The results showed important reduction of tracking errors in the presence of both noise and multipath, especially in C/A code applications. This happens because the distortion of the ACF near its peak is less severe than in the regions away from the peak. Therefore, if tracking can be performed in the peak area, the effects of multipath will be considerably reduced. This is a statement applicable

mostly to the non-coherent DLL versions. For a coherent DLL this observation is not true under the condition of strong multipath reception, due to the susceptibility to carrier phase tracking errors [68].

Possible adaptation to TRN case:

The TRN correlation function is also less distorted in the peak area. This is a consequence of working with terrain signals: adjacent elevation samples are more correlated between each other than distant ones. It is expected for the correlation function to be symmetrical around the peak. Therefore, by performing tracking with a narrow spacing the effects of asymmetry should be reduced. When adapting and testing this strategy to the TRN algorithm, there are two main questions to be answered, namely which spacing values would form the set of inputs and what the boundaries of this interval should be.

Implementation:

The upper limit when choosing the spacing parameter s can be computed from the bandwidth and symmetry properties. However, these denote the maximum value the spacing can take and the method asks for 'narrow' values. Theoretically, there is no lower limit when choosing the spacing. Early and late profiles can be extracted from the database by using infinitesimal offsets relative to the estimated speed. In practice, it depends on the resolution of the database, the speed with which the vehicle is moving and the interpolation methods used when extracting the profiles from the map. The effect of a speed variation on the extracted terrain profiles is a change in the travelled length. The difference in distance between the shifted and prompt versions is proportional to the used spacing. When testing, we have chosen for this value to be equal to a divisor (G) of the resolution of the map (R), as written in (3.6)

$$s = \frac{G \cdot R}{N \cdot t} \tag{3.6}$$

Early-Late Slope Technique(ELS)

This strategy can easily be explained with figure 3.16. The main idea is to determine the slope at both sides of the peak of the distorted correlation function and then use these values to compute a pseudo-range correction [69]. Four correlators *K1,..,K4* are used in this scheme. They have dedicated coordinates: *x1,..,x4* representing the placement with *y1,..,y4* the corresponding ordinates that can be calculated. Using the correlators outputs, the slopes at both sides of the correlation function's peak (*a1, a2*) are determined. The abscissa of the intersection of these two straight lines can be interpreted as the desired pseudo-range correction T and

is calculated using (3.7). The tracking error, T, is used as a feed-back to the loop, estimating the amount the early-late correlators have to be moved.

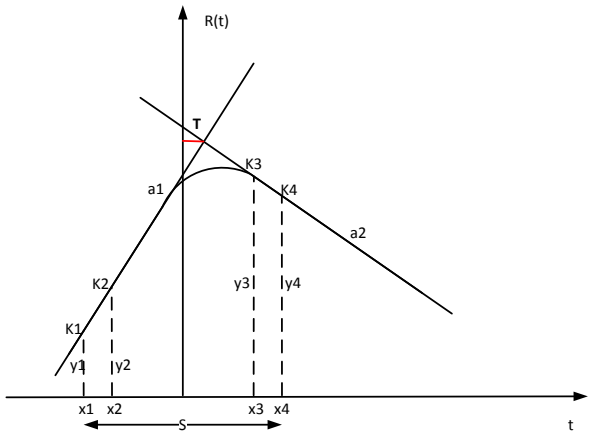


Figure 3.16: ELS technique in GPS: computation of the tracking error

$$T = \frac{(y1 - y4) + s \cdot \frac{a1+a2}{2}}{a1 - a2} \tag{3.7}$$

Possible adaptation to TRN case:
With this method, the correction of the bias is determined by analysing the distortions of the correlation function. Therefore, it fits the description of our problem. However, the sides of the TRN correlation function are not continuous lines, but most likely are composed of segments. The value of the computed slope is strongly dependent on the position of the correlators. Their location, as well as the distance between them has a large impact on the calculated value of the slope and hence, on the estimated correction. Figure 3.17 illustrates this dependency: even slight changes in the position of the correlators can result into a fairly different value of the slope.
Implementation:
Similarly to the NCS method, the upper limit in the correlator's placement is considered to be calculated from the bandwidth and symmetry properties. When choosing the lower limit, it is important to prevent the situation when the two correlators (used to compute one slope) are not placed on the same side of the function's peak. This situation, illustrated in figure 3.18(a), can occur due to disturbances. Therefore, to prevent an erroneous implementation, the first step

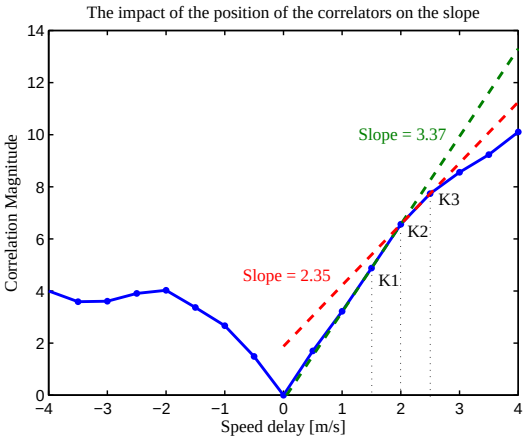


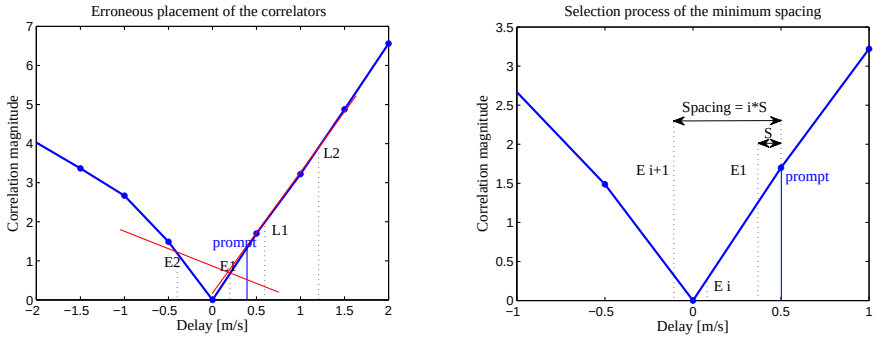
Figure 3.17: ELS technique adapted to TRN: placement of the correlators and the distances impacting the resulting slope

in the ELS technique is to search the minimum position for the placing of the correlators. Search is performed with a small sized step (S in figure 3.18(b)). If the correlation output of the delayed version E_1 is smaller than the prompt correlation value, search will begin. The delayed correlator is moved, while checking if $E_{i+1} < E_i$. In case this inequality is not satisfied any longer, the correlator will be positioned on the correct side. The abscissa of E_{i+1} represents the lower minim value (with respect to the prompt) for the placement of the correlators. A second issue would be the distance between the correlators. Intuitively, we would expect that a wider distance would estimate the slope better than a narrow one. However, in the simulations we have used different values to verify this assumption.

Other investigated strategies were: **Double-delta correlator technique** and **multipath estimation DLL technique**. We concluded that only NCS and ELS are suitable for our situation. Simulations have been carried out for better understanding how these methods perform with terrain signals.

Simulations and results

Simulations were performed in Matlab environment. For the present simulation a sample group of 50 terrain profiles was tested. The description of the terrain profiles is summarized in table 3.3



(a) (b)

Figure 3.18: (a)Erroneous implementation of the ELS method, (b)Search process for the lower limit of the placement of the correlators to avoid an erroneous implementation

Table 3.3: Simulation setup used in the testing of mitigation strategies to minimize the error bias

Parameter	**Value**
Transect length	15
Speed	10 m/s ; constant
Heading	0 deg, constant
Sampling time	10 s
Disturbance(D)	[15% 5% 3%]speed = [1.5 0.5 0.3]m/s
Resolution database	90m/3 arc sec

Narrow Correlator Spacing

The main interest here was to analyze the performance of the technique as a function of the spacing. Another important parameter is the number of iterations needed to find lock, when the difference between the early and late channel correlation outputs was set to 0.001. The tested values for the spacing have been calculated using equation (3.6), resulting in resolution gain factors G = [1, 1/2, 1/3, 1/6] and corresponding spacing = [0.6 0.3 0.2 0.1] m/s.

Early - Late Spacing

The main interest here was to analyze the performance of the technique as a function of the distance between a pair of correlators. Maximum spacing was considered half the bandwidth. The minimum spacing was calculated as described in the previous section. The tested options were:

- **Sp1**: first pair of correlators placed at the minimum possible, second pair at a distance as small as 0.1 from the first;
- **Sp2**: first pair of correlators placed at the minimum possible, second pair at the maximum distance;
- **Sp3**: second pair of correlators placed at the maximum distance, first pair at a distance as small as 0.1 from the second.

Results

When applying NCS method, the size of the disturbance D does not influence the final estimate (and thus the accuracy), only the number of iterations. Figure 3.19 illustrates NCS performance for the different tested spacing. Figure 3.20 illustrates the statistic of the running time for each tested spacing, as a function of D. It can be seen that the number of iterations is inversely proportional with both the spacing and D. The ELS method is influenced by the introduced disturbance. Figure 3.21 illustrates a comparison of the performances for the three implementations. All of them perform better when D is smaller. Sp1 and Sp2 have similar results, whereas Sp3 gives the worst solutions. When testing ELS, outliers were also observed.

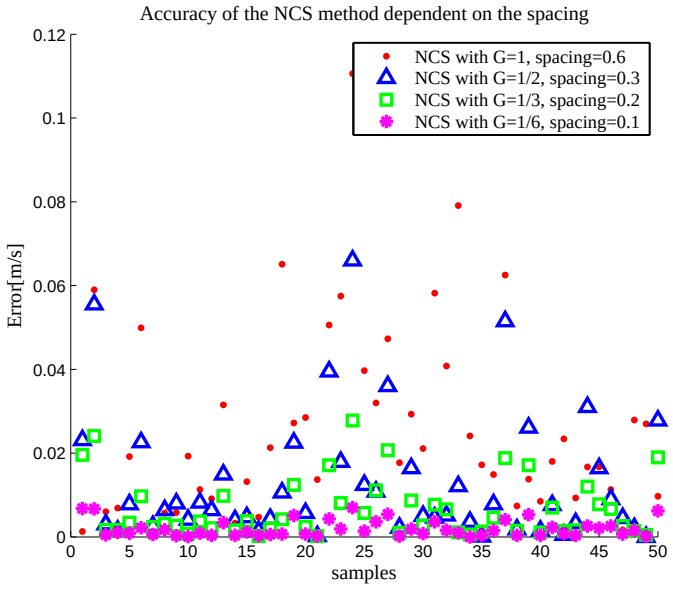


Figure 3.19: Performance of the NCS technique

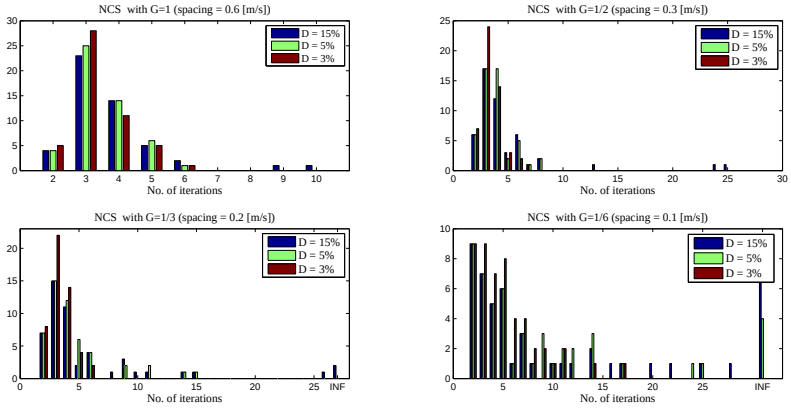


Figure 3.20: Comparison between the number of iterations for NCS

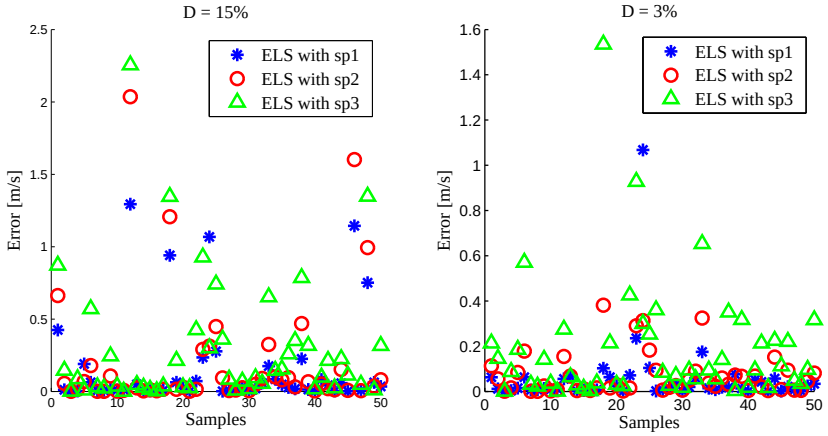


Figure 3.21: Performance of the ELS technique

Finally, in figure 3.22 the two methods are compared. The comparison is made in terms of accuracy, by analyzing the average errors obtained.

To conclude, both methods depend on the introduced disturbance but in a different way. NCS requires more processing time when D is larger, but delivers the same accuracy. ELS is fast, but its accuracy degrades as D grows. Although ELS (with Sp1) may give more precise solutions on some samples, on the average NCS has the

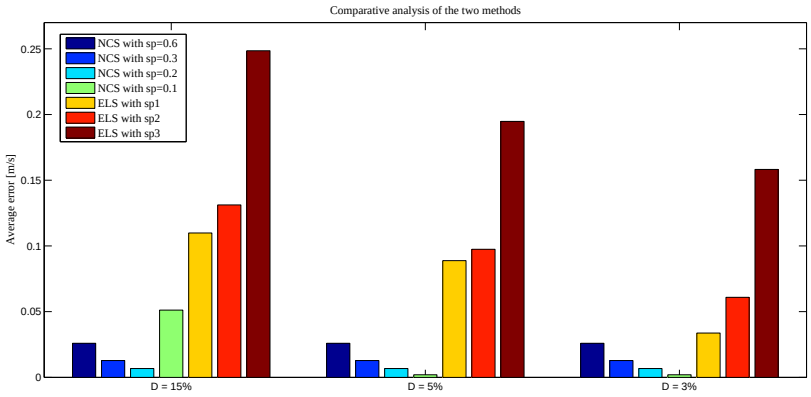


Figure 3.22: Computation of the average errors obtained using the NCS and ELS methods

best performance. ELS tends to be more useful as a refinement method. This leaves space for new ideas: if the intended application needs to be highly accurate ELS can be applied in combination with NCS. However, for the implementations throughout the rest of this thesis, we have chosen to use NCS as a mitigation strategy for minimizing the error bias.

3.5.3 Working towards an adaptive tracking scheme

The design parameters of the TRN tracking loop are: correlator spacing, length of signal (transect length) and correlator threshold. All parameters are elective. However, the choice of their value often proves to be a complicated exercise of trading one aspect of performance for another. The trade-offs involved are discussed below and are summarized in table 3.4 .

Correlator spacing

As it turned out from section 3.5.1 and 3.5.2, the correlator spacing is an essential design parameter for the TRN tracking loops. The value assigned to it is a matter of choice, but it is important to be aware of the implied trade-offs. It was concluded that with “a wider spacing the correlator can handle larger dynamics” and attain fast convergence. On the other hand, a narrow spacing leads to a more precise estimate by decreasing the bias. At this point the need of an adaptive early-late spacing becomes obvious. The tracking will be initiated with a wide spacing, as much as the correlation function allows it. Once convergence is reached and the error is reduced, the spacing

will be narrowed down to reduce the tracking error even further. It was shown that, for a fast convergence, the spacing must be at least larger than the disturbance. For sections of travel where the changes in speed/heading are minimal, tracking with a narrow spacing might be enough. However when the speed or heading change (either due to external disturbances or vehicle dynamics) it is important to start tracking with a wide spacing. This ‘intermediate phase’ (i.e. tracking with a wide spacing) will ensure convergence and already reduce great part of the error, minimizing the running time of the refinement phase, i.e. tracking with a narrow spacing.

Transect length

The length of the transect influences the signal strength. An increase in length yields a better correlation and might enlarge the bandwidth. Based on the analysis of the correlation function, the algorithm may request an increased strength in the signal. Some examples are: larger bandwidth to handle the vehicles dynamics or increased energy to control ambiguity in the correlation function. On the downside, a longer transect triggers an augmentation of the running time. Empirically, it has been noted that there is a certain range for which increasing the signals length is effective, from 15 to 40 samples. However, this depends on the terrain and vehicle dynamics or, differently stated, on the signal properties.

Correlator threshold

This parameter directly influences the convergence time. It has also an impact on the process of minimizing the error bias. In section 3.5.2.1 guidelines on the process of assigning the correlator threshold parameter were provided. As a rule of thumb, a more tolerant value may be assigned to it during the intermediate phase. When tracking is performed closer to the peak, the threshold can be considerably reduced. Just as with all the other parameters, the correlator threshold depends on the correlation function.

Table 3.4: Impact of the selection of the parameters on the performance of the TRN algorithm

Parameter	Restriction	Value		Ambiguity	Error bias	Convergence	Running time
Correlator spacing	$<$ bandwidth	elective	wide	/	+	-	-
	$\approx$ symmetry		narrow	/	-	+	+
Transect length	-	elective	long	-	/	/	+
			short	+	/	/	-
Correlator threshold	influenced by spacing	elective	large	+	depends on spacing	+	depends on CF
			small	-		-	

All design parameters needed in the TRN tracking process are strictly dependent on the properties of the terrain profile. Although “variable and unknown”, these properties can be estimated from the local reference codes in the database. It becomes

obvious that the selection of the parameters should be made dynamically, based on the analysis of the TRN signal and its correlation function. Optimal performance of the system can only be assured by a tracking scheme able to adapt to the terrain signal and to reconfigure itself continuously. In the envisioned adaptive tracking scheme all adaptation must be controlled by the system, meaning that all three parameters must be automatically selected (without explicitly provided input) based on the properties of the terrain signal and/or current performance (given a particular sensor performance). To study how this automatic selection could be performed, the performance impact of changes in the values of the design parameters needs to be evaluated for different types of terrain signals. To explore the resulting design space, an adaptable system (which provides the possibility to change the parameter values) was implemented in the simulation environment (as described by figure 3.5). In this way, various options were tested and the findings were presented throughout this chapter. Based on the results of the study performed in section 3.5.2, adaptivity was implemented for the correlator spacing and correlator threshold. No automated selection was implemented for the transect length parameter. The performance of the sensors greatly determines the choice of transect length, but this will be discussed in the following chapter. Figure 3.23 illustrates how the tracking scheme shown earlier in figure 3.5 was changed to meet the required adaptive capability. The loop is extended with a new block to comply with this demand. This is a process that needs to be performed real-time, as it takes into consideration the analysis of the current correlation function and uses the values of the speed and the heading of the platform at the given moment. An a-priori analysis would be possible only if the travelling route was entirely predetermined. Figure 3.24 illustrates with a flow chart how adaptivity is achieved(at this stage) in the algorithm. To avoid overcrowding the figure, only speed and heading TRN estimates are used in annotations (however it is implied that other data is used, such as terrain signal, sensor speed and heading etc.).

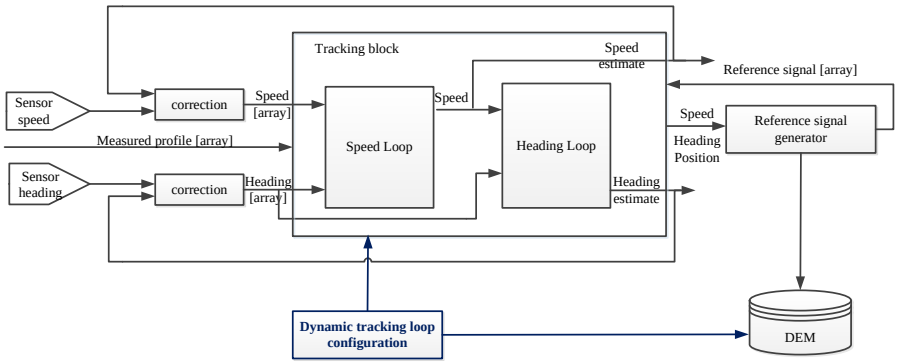


Figure 3.23: Block diagram of TRN tracking module with adaptivity block

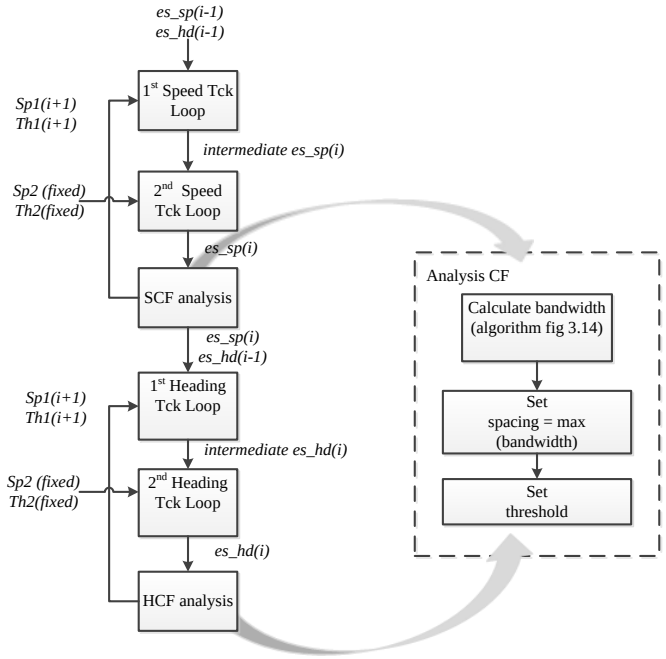


Figure 3.24: Flow chart illustrating how adaptivity is achieved in the TRN algorithm

3.6 Summary and conclusions

In this chapter the early-late concept used in GPS for code tracking has been be successfully applied in the TRN field. The main contributions of the chapter are:

1. The opportunities that the GPS-inspired tracking offers for TRN are recognized and the associated need for real-time adaptivity of the tracking loop to mitigate issues caused by (non-ideal) terrain elevation 'signal' properties is identified:

   (a) Similarly to code acquisition and tracking in GPS, a TRN system needs to find and track a particular sequence of measurements in a larger dataset.

   (b) The basic rationale for the proposed TRN algorithm is to use terrain correlation to 'acquire and track' the speed and heading of the host vehicle, while the position advances are calculated using these estimates together with the previously determined position.

   (c) Terrain 'codes' are random sequences and do not possess the Gold codes correlation properties. Thus, in terms of signal properties, significant differences exist between the uncontrollable TRN signals and the ideal PRN signal used with GPS. The TRN correlation function is asymmetric and has a variable and unknown magnitude and width of the correlation peak. A set of specific properties that characterize the TRN signal and correlation function are also identified.

   (d) When applying the DLL scheme to TRN, main issues caused by the differences in signal properties are: assessing convergence, addition of a bias error to the estimated solution and no standard pre-determined values for the parameters used in the tracking process. Mitigation strategies were proposed to effectively deal with these issues.

   (e) GPS multipath mitigation strategies based on correlation techniques were reviewed for a potential adaptation to TRN to minimize the error bias caused by the asymmetry of the correlation function.

2. The relation between terrain signal properties and tracking loop performance is analysed, in order to define guidelines for adapting the tracking loop parameters:

   (a) The convergence problem can be reduced to a sole parameter: correlator spacing. Convergence can be assessed with the use of an optimal value for the spacing.

   (b) All design parameters needed in the TRN tracking process are strictly dependent on the properties of the terrain profile. Although 'variable and unknown', these properties can be estimated from the local reference

codes in the database. The selection of the parameters must be made dynamically, based on the analysis of the TRN signal and its correlation function.

3. A simulation environment in which the associated degrees of freedom can be modified, i.e. an adaptable tracking loop is designed:

   (a) In accordance to the GPS receiver approach, a two-stage TRN correlation algorithm was designed. During the first stage, 'acquisition', coarse estimates for speed and heading are found. Next, the system switches to a 'tracking' mode, during which the changes in these two parameters are constantly being monitored.

   (b) Acquisition is implemented as a serial search process of the two parameters: speed and heading. The acquisition module must be initialized with an approximate position solution in order to determine which region of the database to search. During acquisition it is also required that the vehicle travels at constant speed and heading.

   (c) Tracking is performed using two separate tracking loops: one to follow changes in speed, one to follow changes in heading. Both trackers are designed as early-late correlators, based on the DLL functional principle.

   (d) Similar to the GPS code correlation function, the TRN correlation function provides a measurement of the alignment between the signal (the measured terrain profile) and the reference codes (database extracted terrain profiles). The 'correlation' method implemented in the algorithm is the MAD function.

4. Simulations to better understand how to adapt as a function of terrain signal content are performed:

   (a) The relation between convergence and correlator spacing was determined empirically, with the use of simulations.

   (b) ELS and NCS techniques have been adapted, implemented and tested. Based on the results, the NCS technique has been chosen.

5. A hi-level architecture to enable the envisioned adaptivity is defined:

   (a) Adaptivity for the spacing parameter was implemented.

   (b) Guidelines on how adaptivity for the transect length could be implemented were set.

Chapter 4

From concept to reality: sensitivity analysis of the TRN algorithm

4.1 Introduction

In Chapter 3 the concept of the proposed TRN algorithm was introduced. The core of the system consists of an adaptive tracking module which continuously estimates the speed and heading of the host vehicle using a scheme inspired by the acquisition and tracking process of GPS signals. The previous chapter focused on demonstrating the feasibility of this concept for terrain navigation and, for this reason, the analysis was performed in a simplified environment: with the vehicle traveling at constant speed and heading and no errors added to the measured profile. In reality speed, heading and terrain measurements are expected to be provided by sensors and, as discussed in Appendix A, this information will be affected by inaccuracies. As a consequence, the answer to many design questions will depend on the sensor data quality and availability. At this point, an in-depth sensitivity analysis is required to understand how the sensor measurement errors will impact the algorithm's performance. In the thesis, sensor data has been categorized into primary and supportive information. The discussion in this chapter is also structured according to this classification.

4.2 Supportive information

4.2.1 The need for supportive information

One of the requirements of the proposed TRN algorithm is to have, as input, speed and heading information from independent sources. The reasoning behind this requirement is the fact that speed and heading solutions are used to determine the relative positions of the samples in the TRN signal. However, before analyzing the impact of sensor uncertainty, a more legitimate question would be if supportive information from independent sources is in fact needed at all. To answer to this question, the performance of the tracking loop was investigated when no additional information is received from sensors. This implementation was entitled "unaided" tracking-loop. Results presented in this section cover only the speed tracking loop. However, the conclusions are applicable to both type of trackers. As explained, an unaided TRN speed tracking loop will track changes in the speed of the host vehicle without the use of any additional information. When a new measurement is added to the transect, the tracker will rely only on its previous speed estimation (until a correction is applied) to compute the location of the new elevation sample point. As a consequence, just one speed value will be used for the entire transect even if in reality the speed of the vehicle is not constant. During a transition in the ground speed (GS), an unaided tracking loop will need multiple new observations of terrain elevation to capture the modification. Let us assume that the vehicle's speed is increasing in a linear fashion. The typical response of the unaided speed tracking loop is illustrated in figure 4.1. The input rise *Ri* represents the time/measurements taken while the GS changes. The early-late based tracking loop seizes the change after a certain delay *Li* and then yields an unbiased estimate: the output rise *Ro*. The final response latency *Lf* is defined as the time/measurements it takes for the tracking loop's output to be synchronized with the input. Ideally the initial and the final response latencies will have equal values. Empirically it was observed that the steeper the change in GS is, the more the tracking loops response deviates from being a linear function of the input, eventually reaching a 'saturation state'. The response of a saturated tracking loop is characterized by: a large *Li* and a large ratio *Ro/Ri*. In Figure 4.2, the appearance of the saturation phenomena is illustrated. The figure shows the response of the tracking loop (marked in green), for the same test case, when the slope of the increase in GS (marked in blue) is gradually augmented.

4.2.1.1 Simulations and results

Extensive simulations have been performed to study the saturation effect. Eight different ramp profiles have been used for the change in GS. It has been considered that the GS suffers a total change of 2 m/s, but the time/number of measurements

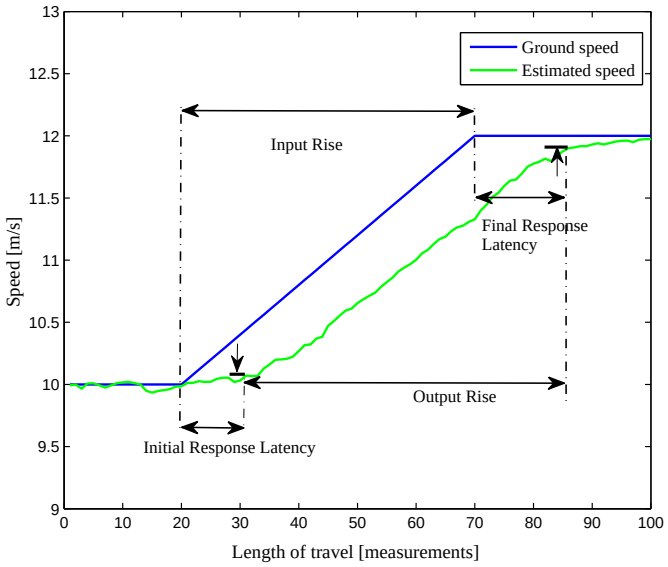


Figure 4.1: Unaided speed tracking loop response to a linear growth in GS

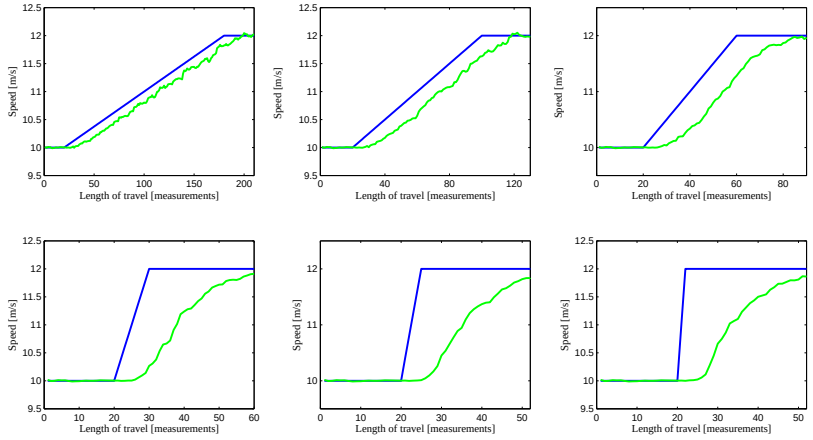


Figure 4.2: The saturation phenomena observed in the TRN speed estimate when the steepness of the linear growth in GS is gradually increased

the platform takes during this growth has been gradually varied (Ri varies from a maximum of 160 to 1 measurements). Each one of these inputs has been tested for 70 cases, taken from different databases. We were interested in a statistical analysis of the parameters illustrated in figure 4.1: Ro, Li and Lf. Ro quantifies the ability of the loop to track the change in GS from 5% to 95% of the absolute growth. It was expected that the length of the TRN signal will also influence the saturation. Therefore, the tests were repeated using a signal with a length of 10, 15 and 30 measurements. Table 4.1, 4.2 and 4.3 illustrate the obtained results. The values for Li, Lf and Ro represent the averages for the simulated examples.

Table 4.1: Ri,Li,Lf for a TRN signal length = 10

Ri	160	80	40	20	10	5	2	1
Ro	155,26	80,83	44,91	28,83	22,62	20,66	20,8	20,8
Li	7,08	6,8	6,24	6,4	5,81	5	5,2	4,4
Lf	2,35	7,77	9,16	15,23	18,43	20,66	24	24,2

Table 4.2: Ri,Li,Lf for a TRN signal length = 15

Ri	160	80	40	20	10	5	2	1
Ro	157,44	84,27	50,49	38,43	30,77	26,85	25,15	25,1
Li	11,42	10,12	9,35	7,36	7,26	6,23	5,42	5,1
Lf	8,86	14,4	19,85	25,79	28,04	28,09	28,58	29,3

Table 4.3: Ri,Li,Lf for a TRN signal length = 30

Ri	160	80	40	20	10	5	2	1
Ro	169,68	92,48	-	-	-	-	-	-
Li	19,56	17,2	14,56	12,75	10,46	8,78	7,76	7,48
Lf	29,25	29,68	-	-	-	-	-	-

Main conclusions drawn after analyzing the results of the simulations were:

- The initial latency in response Li decreases with the slope. When the GS faces a slow growth it takes around 2/3 of a transect to seize this change.

- The final latency in response Lf increases as the slope increases. During saturation this parameter is more than 3 times larger than the initial latency in response. Lf is also directly proportional with the length of the TRN signal. For a longer transect, the tracker loses lock (as seen in table 4.3).

- Ideally *Ro* should be in the same range as *Ri*. For small slopes, that is the case. The length of the TRN signal also influences this difference: longer tracks will yield a larger difference. During saturation *Ro* reaches large values: $Ro \geq 3{*}Ri$. (in case of a unitary slope $Ro \geq 20{*}Ri$).

4.2.1.2 Discussion

When no additional sensor information is used, the bandwidth with which the tracker can follow changes is rather limited. For slow increases in the GS/track, this is translated in a delay in the response of the speed/heading tracking loop. Sharp changes cannot be immediately tracked and might put the tracker in a saturation state. Clearly the unaided implementation cannot be used on a current basis. Exceptions are portions of flight at (almost) constant speed and heading. A similar issue is found in the GPS receiver design. The Doppler shift in the GPS signals is caused by the relative motion between the satellite and the receiver. Once in lock, tracking becomes rather predictable. However, there are applications where the motion of the platform is so severe that it affects the tracking process. In other cases the receiver might need to be assisted in reducing the bandwidth for better noise suppression. In these situations 'Doppler-aiding' is applied. By doing so the Doppler-estimate is provided beforehand to the GPS receiver, in order to release part of its loadings and increase its capability of dealing with weak signals. During Doppler-aiding, the platform dynamics are normally tracked by an inertial system and this information is incorporated into the GPS tracking loops.

4.2.1.3 Aided tracking loops implementation in the TRN algorithm

To provide sufficient bandwidth for detection of changes in speed/heading, the tracking loops are continuously aided with sensor data. The new sample points added to the transect are computed using the value measured by the sensor compared to the current established reference. Figure 4.3 illustrates how the sensor measurements are used in the tracking process. Unlike in the unaided implementation, when the tracker estimates one 'best' average speed/heading that maximizes correlation, here a single constant 'unknown', the reference speed V_{ref}, is kept in the early-late set and the known increases in speed Δ_V are added. For each new observation, a new value is added to Δ_V. This is calculated by subtracting the current speed reference from the value provided by the sensor V_m . Note that no corrections are applied to Δ_V. Thus, any compensation for errors in Δ_V will be reflected only in the reference value. If the error should vary over the length of Δ_V, the compensation will target the average value over the entire array and it will be made to V_{ref}.

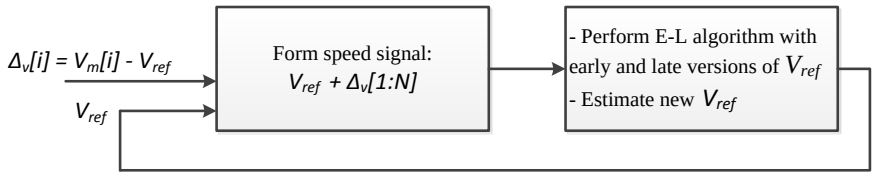


Figure 4.3: Illustration of how sensor measurements are used in the TRN tracking loop

4.2.2 Speed and heading sensor errors

In general, sensor data can be integrated at a variety of levels in the information flow, starting from the raw unprocessed sensor output to the level where the integration is performed by the user. In the context of the proposed TRN application, it is considered that the sensors are self-contained. The raw sensor signals are first processed, instrument errors are calibrated and independent estimates (in the form of speed and heading) are fed into the TRN algorithm. In appendix A the main navigation sensors that can be used to provide supportive information to an aerial TRN application are discussed. As pointed out, sensors are subject to a variety of errors and therefore, the interpretation of their measurements often differs from the true physical value of the parameter that is measured. Depending on the sensor type, part of the total error can normally be corrected or calibrated. For example, the variation error of a magnetic compass is corrected within the system's computer using charts of the Earth's magnetic variation. Similarly, a pressure error correction and a compressibility error correction must be applied to air data sensor readings to obtain the true airspeed(TAS); a temperature dependent error component can also be corrected in case of inertial sensors using laboratory calibration data and so on. It should be said that, in principle, there is no such thing as a perfect error correction and residual contributions are left over from the calibration processes. However, it is largely the post calibration sensor performance that is relevant in the design of the aided tracking loops. The (post-calibration) error sources, as well as their dynamics, typically depend on the sensing technology. Pressure sensitive sensors are mechanical and are subject to inherent errors due to frictions within the instrument and manufacturing tolerances. The drift, a slowly growing error in time, is characteristic to inertial systems. Orientation and velocity are obtained by integrating the gyroscope and accelerometer signals and therefore, all existing errors in the inertial sensors will propagate through causing the integration error, another specific error for inertial systems. Among the characteristic errors for the Doppler

velocity radar we remind errors due to scattering or poor reflection. All these aspects are addressed in appendix A. Besides specific errors, all sensors are affected by noise, a random and normally uncorrelated variation superimposed on the signal. There are various sources that can generate noise, such as: electronic components, frictions in mechanical parts, external disturbances during the measurement process etc.

Another source of errors for the TRN algorithm is not using the earth referenced speed and orientation. In other words, sensor information given relative to another reference represents a biased measurement. Examples are the use of magnetic, gyroscopic and pressure based sensors. Due to flow-fields, the quantities measured by these sensors differ from the earth referenced values. Moreover, a change in the actual speed/heading and a change in the flow-rate will both yield a change in the sensed quantities. For a better understanding, flow-fields are explained in the following paragraph.

Flow fields

Platforms typically encounter flow-fields during navigation. Airplanes are subject to air motion (cross winds, air currents). Ships are subject to water flow (tide, currents). A flow field has both a magnitude and a direction, both of which can vary substantially even over short distances due to local bathymetry/atmospheric conditions. Their effect is one of deflecting the vehicle from its original heading. Figure 4.4 illustrates the influence of wind on a moving aircraft. The track or course over ground is the actual path followed by the platform from point A to point B. This is a combination of the motion of the aircraft and the motion of the air. Heading is the direction in which the nose of the aircraft points during travel. They are both expressed in degrees from 0° clockwise to 360° in compass convention (0° being north, 90° being east). The angle between the heading and the track is named drift angle/crab angle/ correction angle. Consequently, the speed of the aircraft is affected by the flow fields in a similar manner. The earth referenced speed of the aircraft (GS) is a combination of the movement of the air mass in reference to the ground and the forward movement of the aircraft through the air mass.

Error model

To account for all possible errors that arise in the supportive sensor measurements, we chose to model the total introduced error as a combined effect of **measurement noise** and induced **bias**. All inevitable deviations caused by typically present physical phenomena in the sensors represent the measurement noise. The noise component follows a Gaussian distribution with zero mean and a known standard deviation. The remaining errors are modelled as bias. This component is characterized by an error distribution in which the mean and/or the variance have been shifted away. Note that the term 'bias' is not used in the sense of a systematic error which corresponds

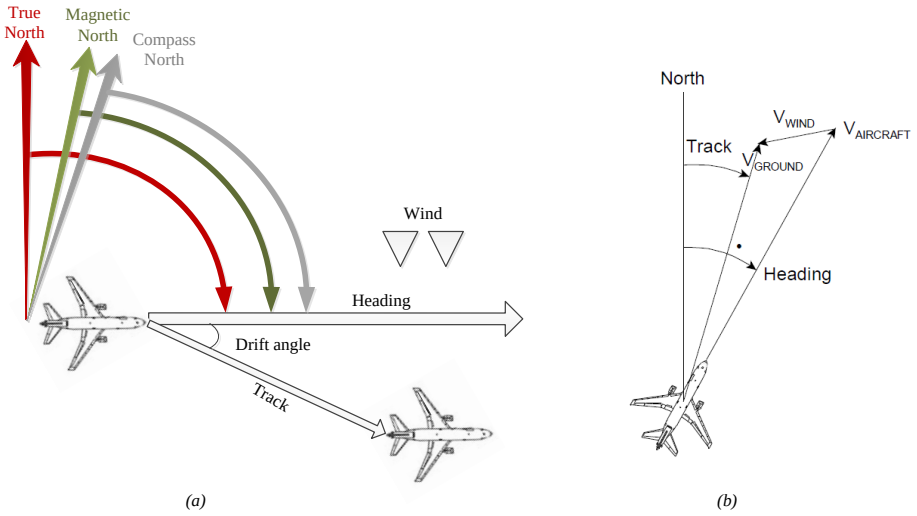


Figure 4.4: (a)Relation between true, magnetic and compass heading; (b)Relation between airspeed, ground speed and wind-speed

to an error that remains constant over an extensive period of time. In the context of this dissertation, the bias represents a way to generically model the likelihood of large errors that deviate from the zero-mean model. Their distribution may or may not be Gaussian, might possess heavy tails or may follow any given shape. In fact, even if the bias distribution may be estimated for a particular configuration, it can hardly be assumed to remain the same for all possible sensor configurations. However, it will later be shown that the dynamic behaviour of the unknown bias is much more important than the distribution itself. The bias error component is further divided into two groups: **sensor specific bias** (which is a function of sensor performance, such as inertial drift) and non-sensor specific bias or **data bias** (which is not a function of sensor performance itself, but appears due to the operational environment - such as the earlier discussed flow field induced differences between measured speed and/or heading and earth referenced speed and ground track). Data bias is to a certain extent predictable and, for this reason, the two types of biases will be treated differently in the context of the proposed TRN algorithm. Finally, a third category of errors are the 'abnormally large' measurement errors, called outliers in the statistical literature. They often occur independently. In this dissertation it was assumed that such an error is too rare to require investigation and, in the event of it, the outlier could easily be rejected by the TRN algorithm.

4.2.3 Analysis

4.2.3.1 Impact of speed errors on the TRN correlation function

Consider a TRN signal containing N samples, each obtained by taking terrain measurements with a constant sampling rate t. Let us assume the heading of the vehicle is constant and known. A speed error, ϵ, will introduce an offset that will grow with each sample, as shown in figure 4.5. The displacement error between the positions of the samples of the true TRN signal and the ones of the generated replicas can be calculated with equation 4.1.

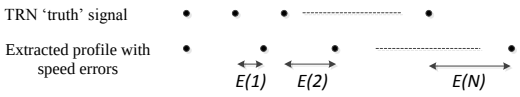


Figure 4.5: Illustration of the effect of a speed error on the TRN signal replica

$$E(i) = \sum \epsilon_k * t \tag{4.1}$$

Although ϵ might vary over the signal, only constant errors are considered when building the signal replicas(figure 4.3). In this context, equation 4.1 becomes equivalent with equation 4.2.

$$E(i) = \epsilon * t * i \tag{4.2}$$

$$E(i) = f * R_{db} \tag{4.3}$$

A speed error can be expressed as a fraction f of the database resolution R_{db}. In this way, it can be identified where a first slip of a database sample will occur. Using equation 4.3 the effect of the speed error can be expressed in terms of the amount of elevation samples after which the error has caused a sample position offset equal to one (or a multiple of the) database spacing. Alternatively, by using this equation together with the transect length, the effect of a speed error can also be expressed as a fraction of the transect length over which no slip occurs. It is important to relate the introduced errors to the database resolution because by adding new grid points to the generated profiles new information content is added. However, the impact of this fact on the cross correlation value will also depend on (the signal content of) the terrain itself. If the correlation function has a wide neat peak, shifting a sample point will not cause a significant decorrelation as it would in the case of a narrow peak. Thus, equation 4.3 can be used to determine how much resolution

in speed can be achieved based on the shape of the correlation function. A wide peak can handle larger errors, but the resolution with which the speed error can be identified is poorer. As described in chapter 3, the SCF has a bandwidth property representing the width of the correlation peak and the tracker will generate a solution as long as the early-late signals are placed within this space. The errors in speed push the generated signals away from the peak. As a result, the speed estimates will be noisy. However, **the goal of the speed tracking process is not to get an instantaneous accurate estimate of the error bias, but to prevent loss of lock and to move the tracking loop in the right direction in order to track changes in bias.** Given that the terrain in itself is unique, the maximum amount of speed error that the tracker can accommodate without losing lock is variable. A terrain with height varying slowly between subsequent samples, i.e. terrain with a low frequency signal content, will de-correlate slower and will tolerate larger speed errors at the cost of having a poorer ability to detect errors. On the contrary, if the terrain has a dominant high frequency signal content the same amount of shift caused by speed errors will push the tracker into the uncertainty region faster. The question is indeed 'how sensitive to speed errors is the terrain?'. A terrain ACF is used to obtain a qualitative description of the terrain signal content. Given the start position and the length of a transect, a profile is extracted from the database containing only database grid points. This profile is correlated with equal length terrain profiles extracted in the same direction of travel. The terrain ACF provides insight on the frequency content of the TRN signal and on the energy per database sample 'available' for the speed tracking process relative to the uncertainty band. Figures 4.6 through 4.9 picture the terrain ACF and the SCF for different terrain profiles together with the displacement from the mean elevation of the terrain, used as a visual indicator of the frequency content.

Let us discuss in detail one of the previous examples. For the scenario shown in figure 4.6 the following values were used: time sampling of 10s; speed equal to 10 m/s; signal length of 20 samples. The ACF shows that the autocorrelation peak cannot be distinguished from the cross-correlation peaks once the error exceeds the limit of 2 database samples. The SCF indicates that errors up to 2 m/s can be corrected. However, this does not contradict the ACF showings. The explanation lies in the fact that SCF and ACF are not equivalent. The information from the ACF provides an indication on the effect of speed error on the tracking loop, showing the relation between maintaining lock while losing signal power. Using 4.2 we can calculate the impact of various speed errors. For an error of 1 m/s, a full database sample slip will happen after the 9th signal sample. As pointed out earlier, this means that a 1 m/s speed error yields a ratio of 0.45 as 45% of the transect will contribute a value that is close to the max of the autocorrelation peak, the next 45% will contribute a value that is close to the value of the ACF one sample away from the peak and the last

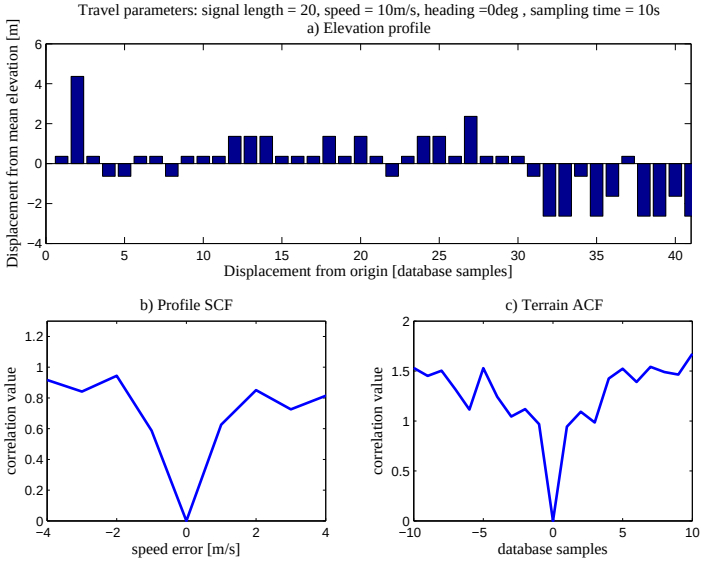


Figure 4.6: Example 1 illustrating the relation between (a)frequency content of TRN signal, (b)SCF and (c)ACF

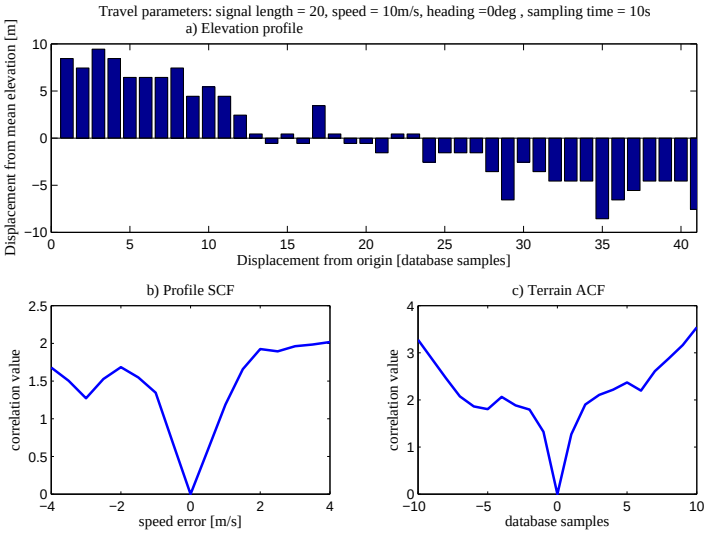


Figure 4.7: Example 2 illustrating the relation between (a)frequency content of TRN signal, (b)SCF and (c)ACF

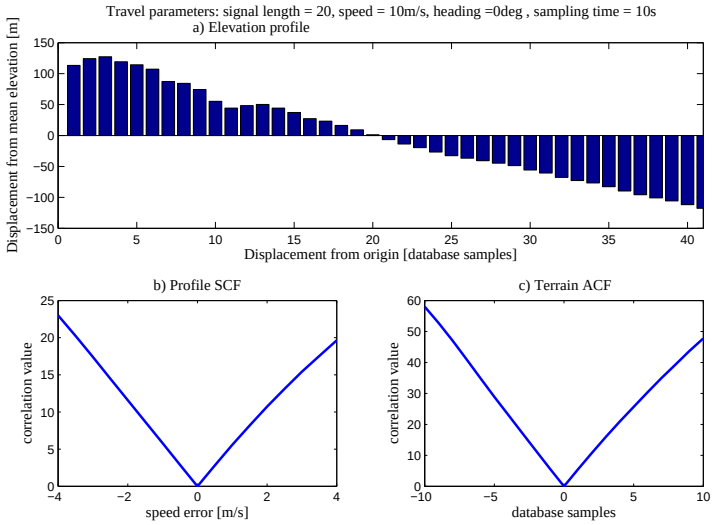


Figure 4.8: Example 3 illustrating the relation between (a)frequency content of TRN signal, (b)SCF and (c)ACF

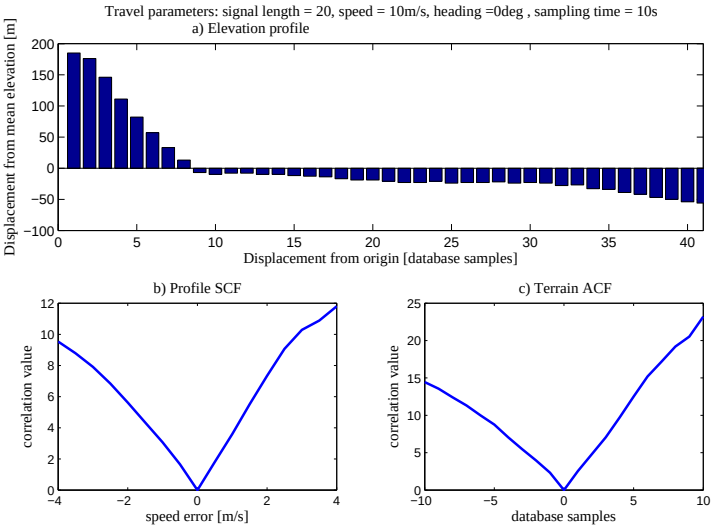


Figure 4.9: Example 4 illustrating the relation between (a)frequency content of TRN signal, (b)SCF and (c)ACF

2 samples will contribute a value that is close to the value of the ACF two samples away from the peak. An error of 2 m/s yields a ratio of 0.225 and so on. An error of 3 m/s has a rate of 0.15, which means 30% of the generated profile will contribute with values inside the autocorrelation peak and 70% will have a slip of more than 2 samples. Clearly, if the terrain ACF is wide (in this context wide meaning that a shift of the transect of 2 or 3 samples still yields rather high correlation relative to the cross-correlation values, so for example the situation in figures 4.8 and 4.9), a ratio of 0.15 for a certain speed error will be more acceptable for the speed tracking loop as compared to a situation in which the autocorrelation peak is narrow (where one shift already causes a significant reduction as seen in figures 4.6 and also 4.7).

Building the TRN speed correlation function

The spacing used in the tracking process is derived from the analysis of the SCF, as explained in Chapter 3. The SCF is a discrete function (computed for certain points, according to the resolution of the function) and therefore, the value assigned to the spacing will be a multiple of the resolution. Thus, the resolution of the SCF is also an important parameter and it should not be chosen randomly. The maximum error in speed used for the SCF is chosen based on equation 4.3, with f being an elective parameter. So, for example, if $f = 2$ it means that the SCF tests profiles that have a displacement of up to a maximum of 2 database grid points compared to the prompt profile. The resolution is computed by dividing the maximum error to the desired number of points in the SCF.

4.2.3.2 Impact of heading errors on the TRN correlation function

To analyze the impact of a heading error, let us consider a TRN signal containing N measurements taken with a sampling rate t, while the platform is travelling with a constant speed v in a constant direction. In case of a heading error ϵ, the location of the samples in the generated profiles relative to the reference samples will slip as illustrated in figure 4.10 . The value of the introduced slip E is computed using equation 4.4. Similar to the analysis in section 4.2.3.2, the effect of the heading error can be quantified in terms of the amount of elevation samples after which the error causes an offset of one or (a multiple of the) database spacing. Using the transect length, a heading error ratio can be further calculated as the fraction of the transect for which no slip (larger than the database resolution) occurs.

$$E(i) = 2 * \sin \epsilon/2 * v * t * i \tag{4.4}$$

$$d = 2 * \sin^2 \epsilon/2 * v * t * i \tag{4.5}$$

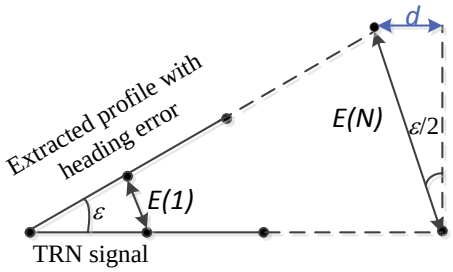
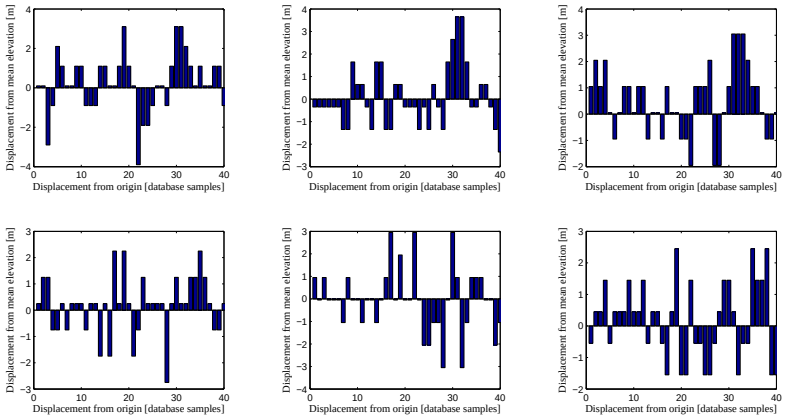


Figure 4.10: Illustration of the effect of a heading error on the TRN signal replica

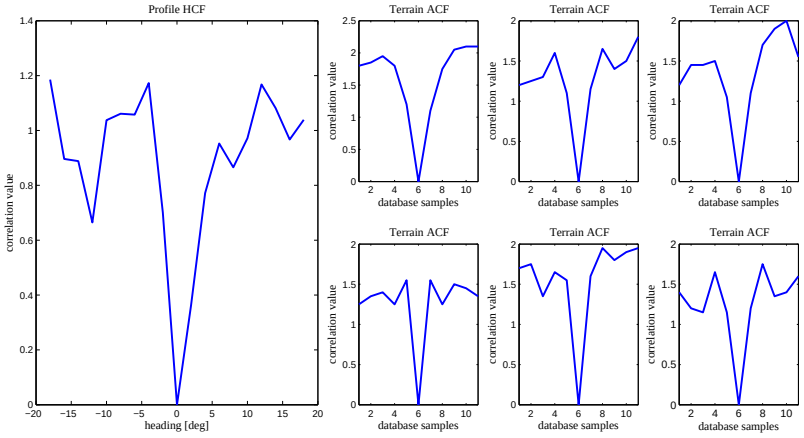
$$E(i) = f * R_{db} \tag{4.6}$$

As with the speed error, the terrain itself is expected to impact the 'sensitivity' to heading errors of the heading tracking loop. Terrain ACFs are used to obtain a qualitative description of the terrain signal content. The terrain is analyzed in the direction of the heading errors by calculating several ACFs of transects perpendicular to the direction of travel. In reality the heading error causes a displacement error at an angle of $\epsilon/2$ from the perpendicular as illustrated in figure 4.10. Thus, by computing the terrain ACF for perpendicular transects the underlying assumption is that the distance d from figure 4.10 and given by equation 4.5 is reasonably small (smaller than one database spacing). For long transects or for larger heading errors this might not be the case and d could grow beyond this limit towards the end of the analysed profile. However, by analysing the ACF of several parallel profiles (perpendicular on the direction of travel) the nature of the frequency content of the terrain can be assessed. Figures 4.11 and 4.12 illustrate the relation between the HCF and the ACFs of terrain profiles extracted on a perpendicular direction with respect to the heading. For the test case illustrated in figure 4.11 the ACF indicates a high frequency content in the terrain and the average autocorrelation peak has a width of 2 database samples. According to the HCF, the tracker will lose lock in the event of a heading error larger than 5 degrees. Given the values used in this test case an error of 5 degrees has a ratio of 0.5 meaning that half of the samples in the transect will contribute a value that is within the autocorrelation peak. Compared to errors in the speed measurements, as presented in figure 4.6, one could draw the conclusion that the heading tracking loop is more 'sensitive' to errors, especially for terrains with a strong signature. Figure

4.12 gives a better insight of the relation between HCF and ACFs. It can be seen that the frequency content of the terrain is slowly changing, varying from a high frequency content (narrow ACF) to a low frequency content (wider ACF). As a consequence the bandwidth of the HCF is maximized and the heading tracking loop is likely to maintain lock in the presence of larger errors in heading.



(a)



(b)

Figure 4.11: Example 1 illustrating the relation between (a)frequency content of TRN signal, (b)HCF and ACF

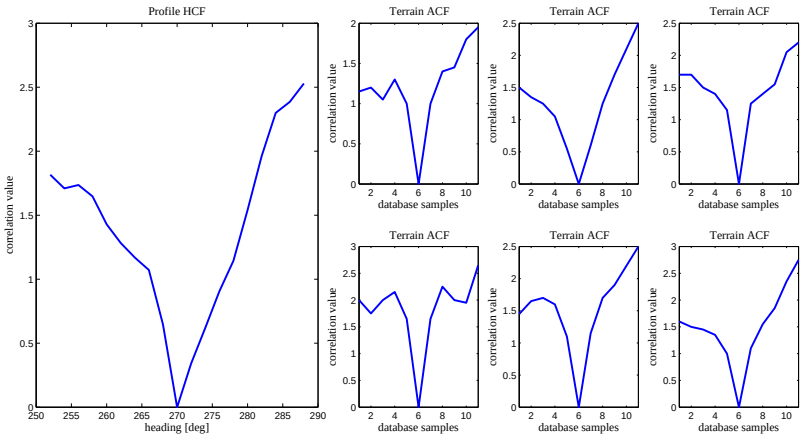


Figure 4.12: Example 2 illustrating the relation between HCF for the TRN signal and corresponding ACF

Building the TRN heading correlation function

The HCF is formed in a similar manner to the SCF. The maximum error in heading is chosen based on equation 4.6, with f being an elective parameter. The resolution of the function is computed by dividing the maximum error to the desired number of points in the HCF.

4.2.3.3 Discussion: an adaptive configuration for the tracking loops

A conclusive remark from the analysis performed in sections 4.2.3.1 and 4.2.3.2 is that terrain has a significant impact on the observed performance of the aided TRN tracking loops. In other words, based on the analysis of the terrain profile alone, it is possible to distinguish between terrain in which the speed/heading tracking loop will be less or more sensitive to errors in speed/heading. The terrain ACF is the tool used to convey sensitivity information related to the impact of sensor errors, by looking at the dominant signal characteristic of the terrain. Simulations have shown that terrain with a rather low frequency change in the elevation is less sensitive to the same amount of speed/heading error as compared to a predominantly high frequency terrain. Results of the study also indicated that the tracking algorithm is more sensitive to heading errors. In similar conditions (i.e. equal size of ACF peak), the ratio needed to prevent losing lock is larger for the heading tracking loop as compared to the speed one. The relation between the terrain ACF and the performance of the aided TRN tracking loops is exploited to prevent the trackers from losing lock. Given the nature of the terrain and the expected error characteristics of the navigation

sensors, the performance of the tracking algorithm can be predicted. Alternatively, thresholds can be computed for the maximum accepted sensor data uncertainty whilst maintaining lock. The length of the TRN signal should be selected based on the properties of the ACFs and the performance of the available sensors. A longer transect can be beneficial because of the addition in signal content, but will also amplify the impact of the sensor errors. The rate at which measurements are collected can also be adapted to the terrain profile. This can be achieved by modifying either the sampling rate or the speed [1]. Although from a database perspective this may be considered over sampling, increasing the sampling rate can be justified from the perspective of signal content. For example, when high frequency content is dominant, an increase in the sampling rate will strengthen the correlation process. It becomes obvious that **the adaptivity concept** introduced in chapter 3 **can be further extended by allowing for the tracking loops to be dynamically configured based on the properties of the terrain and the expected performance of the supportive sensors.** Further refinements are certainly possible. Such an evaluation could also provide the basis for the guidance needed for choosing routes that assure the best performance of the tracking algorithm. Depending on the context, 'best performance' will have different significations such as: higher accuracy (i.e. earlier bias detection), maximum tracking bandwidth (i.e. the ability to deal with larger errors), faster convergence etc. Clearly if information can be conveyed that indicates to the operator whether a certain manoeuvre can provide enhanced accuracy or better options to prevent loss of lock, this would be a valuable addition. The current implementation provides only the possibility of dynamic selection of the parameters of the algorithm for a given route.

4.2.3.4 Impact of the speed and heading errors on the algorithms performance

The earlier discussed analysis of the impact of supportive sensor measurement errors on the performance of the TRN trackers addressed the problem from the perspective of a single snapshot of the algorithm (i.e. one speed/heading estimate computation).The present section analyzes the effect of the sensor errors on the performance of the algorithm for longer routes. The impact of each type of modeled error (as discussed in section 4.2.2) is addressed individually. Due to space considerations, the simulations presented in this section cover mainly the speed tracking loop. However, results and conclusions are also valid for the heading tracking loop. In section 4.2.1.3, figure 4.3 explains how the aided tracking loops estimate the speed/heading: a single constant 'unknown' for the speed/heading is kept in the early and late sets and the known increases (provided by the supportive sensors) for each observation are added.

[1] Although theoretically correct, modifying the speed of the platform is harder to achieve in practice due to operating limitations.

Although sensor measurement errors will typically vary throughout the signal, the algorithm does not take into account individual variations and works by averaging the measurement errors for the entire transect. Thus, the distribution of subsequent changes in error over the analyzed signal becomes more important than the absolute value of the error itself.

Noise component

The noise component of the measurement error follows an approximately Gaussian distribution with a zero-mean and a known standard deviation (typically specified as the tolerance of the instrument). It is expected for input measurement noise to generate noise at the output of the trackers, because zero-mean errors will tend to be averaged out in the correlation process. However, the effect of noise is expected to be negligible and can be further minimized through filtering. Figure 4.13 illustrates the impact of noisy sensor data on the performance of the speed tracking loop. Noise can be removed from the output signal through filtering techniques, as seen from the figure.

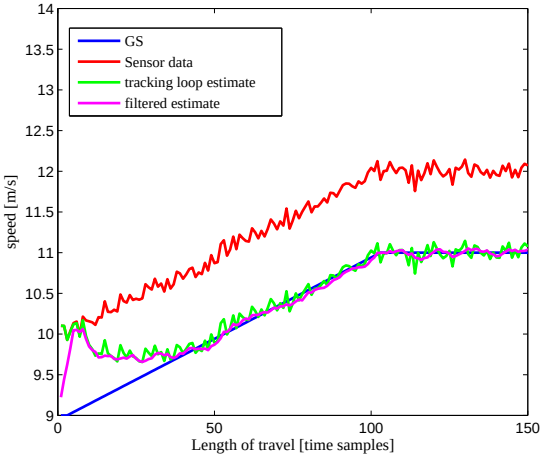


Figure 4.13: Response of the speed tracking loop when the input sensor data is affected by a constant data bias and noise

Data bias component

The grouping of the bias into a sensor specific bias and a data bias was needed because between the two components there is a significant difference in both scale and distribution. The data bias appears due to either platform motion or operational

environment. Flow fields will cause measurement errors only for particular sensors, as described in section 4.2.2. The change in error is rather low frequency because the platform is expected to change its position slowly (and in a predictable way) during a dense set of measurements. Thus, the data bias can be further modeled as static or variable. Figure 4.13 depicted the response of the speed tracking loop for a scenario in which the sensor data is affected by the combined effect of noise and a constant bias. The tracking algorithm is able to estimate the data bias after a certain period. To illustrate the performance of the tracking algorithm in the presence of a variable data bias, figures 4.14 (a) and (b) show the response of the speed tracking loop for two different scenarios. The situation imagined for the first scenario is a vehicle travelling at a constant GS, meanwhile the sensor is measuring a growing bias. As illustrated in figure 4.14(a), in the initial onset of the ramp, the estimated speed from the tracking loop still follows the sensor measurement. This is due to the fact that new (and erroneously located) elevation samples are added to the array with the local terrain reference samples. As the number of erroneously located samples grows relative to the correctly located samples, the speed estimated by the tracking loop will increase to deviate from the speed provided by the sensor. The tracking loop takes the change in speed into account, but tries to come up with an average speed that together with the profile described by the change in speed yields the highest correlation. As long as the data bias keeps changing, the tracking loop will not be able to estimate the correct speed. Once the data bias returns to constant value, it will take the tracking loop one full length of the transect array to get rid of the effect of the change in bias. During this period, the accuracy of the speed and position estimates is lower. Detection is possible by monitoring the difference between the speed provided by the sensors and the one estimated by the tracking loop. This raises the question if and how the accuracy can be restored once it has been detected that the change in measured speed is caused by a change in the data bias. One possibility is to run in parallel an unaided tracking loop because this type of implementation will provide superior accuracy in case of portions of flight at constant GS. In the second scenario, a data bias change occurs simultaneously with a change in the GS. As seen in figure 4.14(b) data bias varies in the same time as the GS and then becomes constant, at a different reference value. The behaviour of the speed tracking loop is similar to the previous discussed scenario: initially the estimated speed will follow the sensors indication, then it will settle into an average value. When the change in data bias is again zero, the tracking loop is able to estimate the new data bias reference and detect the error after a full length of the transect array. During the period in which the data bias and the GS are varying the accuracy of the tracking loop is degraded. The detection of the change in data bias is still possible, but it depends on the absolute value of the bias. A small change may remain undetected. Note that an unaided implementation of the tracking loop performs poorly in these conditions. Concluding, changes in the data bias cannot

be accurately estimated with the same pace as they occur. However, the goal of the algorithm is not to provide accurate estimates of the bias, but to keep the tracker in lock. This is achieved by providing the estimates that maximize correlation, rather than the most accurate estimations.

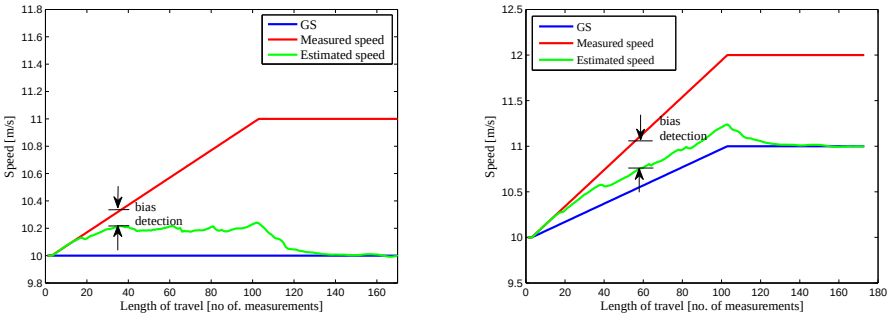
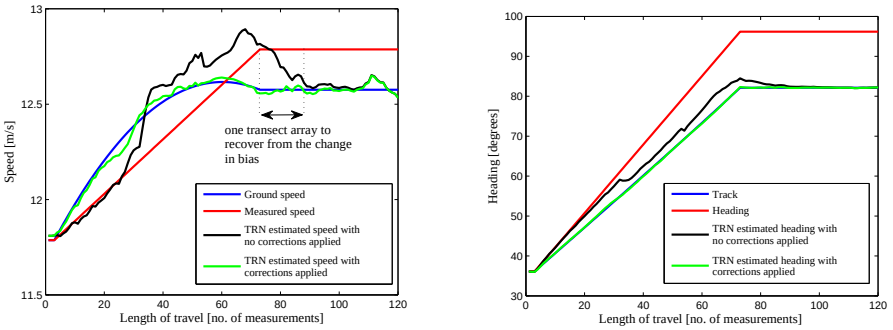


Figure 4.14: Response of the TRN speed tracking loop to a change in (a)bias, (b)both in bias and GS.

If the data bias is due to the presence of a flow field, a change in bias (in both speed and heading) will occur every time the orientation of the platform will change. If the magnitude and direction of the flow-field can be estimated, the change in data bias can be predicted as a function of change in heading. With this information the measured speed/heading can be already corrected before being used in the tracking, further increasing tracking bandwidth for changes in GS/track. This hypothesis has been tested for several scenarios. One randomly chosen example is illustrated in figure 4.15. In this test case, the body of the platform changes its heading from 30° to 90° and during this manoeuvre the speed gradually increases. The imagined flow field has a constant magnitude and a direction of 55°. Note that GS and track were accurately estimated at the start of the manoeuvre. In figure 4.15 the black line marks the response of the tracking loops when no corrections are applied. As long as the data bias keeps changing, the tracking loop will have less accurate estimates. If the flow field is estimated, it is possible to predict and correct the sensor measurements during a manoeuvre. The green line marks the response of the tracking loops when corrections are applied.

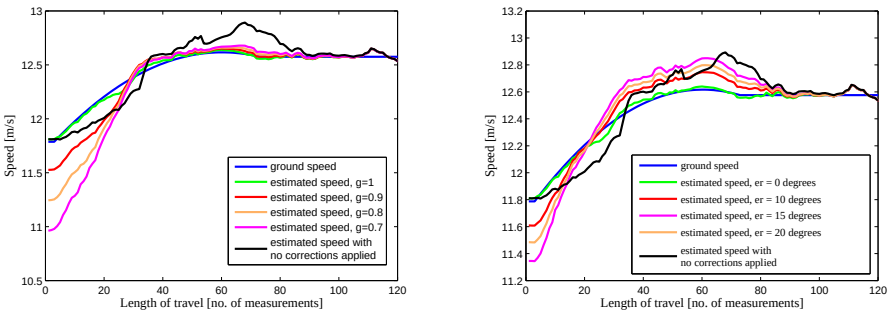
As seen previously from figure 4.4, a flow field can be estimated using vector calculations. Matlab simulations were carried out to give an insight on the behaviour of the tracking loops when the flow field is incorrectly estimated. Figure 4.16 and figure 4.17 illustrate, for the same test case depicted in figure 4.15, the response of the speed and heading tracking loops when the (a)magnitude and (b)direction of the flow

field are incorrectly estimated. The error in the estimated magnitude was considered to be a fraction (noted with g) of the flows real magnitude. The error in the estimated magnitude of the flow field was taken in absolute values, as noted in the legend. The results showed that predictive corrections improve the performance of the tracking loops, even with a prediction accuracy of 70% .



(a) (b)

Figure 4.15: Response of the (a)speed and (b)heading tracking loop during a turning manoeuvre: bias not corrected in black, bias corrected in green

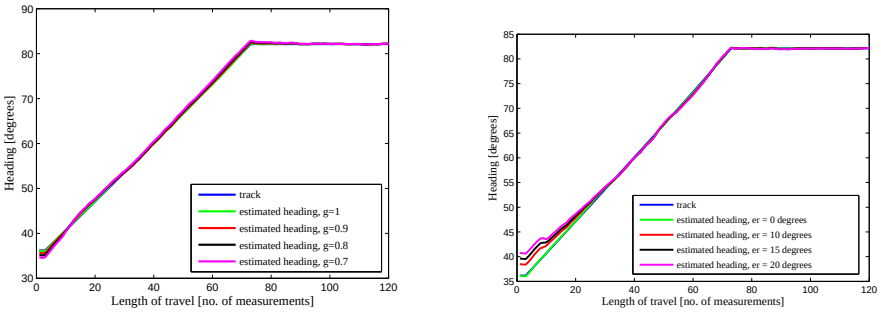


(a) (b)

Figure 4.16: Response of the speed tracking loop when bias is predicted and corrected: (a)estimated magnitude = g * flow magnitude, direction accurately estimated; (b)magnitude accurately estimated, error in the estimated direction

Specific bias component

The specific bias component models the contribution of all sensor specific errors.



(a) (b)

Figure 4.17: Response of the heading tracking loop when bias is corrected: (a)estimated magnitude = g * flow magnitude, direction accurately estimated; (b)magnitude accurately estimated, error in the estimate direction

Clearly, the error spectrum will depend on the type of sensor and generalization is hardly possible. Once again, the spectrum of the change in error has a larger impact on the TRN algorithm. A low-frequency distribution of the subsequent changes in error can be detected and compensated by the algorithm. This happens because the algorithm estimates an average error for the entire transect and the mean of a slowly varying sequence of errors will not vary drastically from the individual values. Specific bias errors which have a very low frequency within a run can be considered to be constant. The analysis here is identical to the one presented for a constant data bias. The INS drift is an example of a low frequency error. A specific bias with a high frequency rate of change will be very difficult to identify. The algorithm corrects for an average error and, as such, it will find a reference value that together with the delta vector maximizes the correlation. If the errors vary significantly within a run (as it would be the case for an error with a high frequency in the rate of change) this solution will most likely not be an accurate estimate.

4.3 Primary information

4.3.1 Radar Altimeter errors

Appendix A addresses the radar altimeter operation, examining possible errors related to the use of this instrument. When discussing the modelling of altimeter measurement errors and their impact on the performance of the proposed TRN algorithm, we will look into more detail only at two types of errors: altitude dependent noise errors and slant range errors.

4.3.1.1 Altitude dependent noise errors

The manufacturer normally provides altitude accuracy specifications in the form of noise and/or noise as a function of height. For example, a specification of an altitude accuracy of ±(2 ft. + 2% of range) will be interpreted as an error with a uniform distribution with a zero mean and a standard deviation of 2ft plus an additional noise of 2% of the true altitude, also characterized by a normal distribution. The additional noise is mainly caused by two phenomena: the decrease of the signal to noise ratio with altitude; the increase of the effective diameter of the radar footprint with altitude. Additionally, if the radar cannot fully penetrate vegetation and climatic matter, further noise is induced on the indicated altitude. In conclusion, the altimeter height errors (which typically are larger than the manufacturers specifications) are a function of the following factors:

- Performance (unique to the make, model and manufacturer);
- Altitude of the flight;
- Antenna beamwidth (and spot size);
- Geographic region;
- Climatic season.

4.3.1.2 Slant range errors

In [14] the plumb-bob height is defined as:"the distance between the aircraft and the terrain along a vector originating from the centre of the aircraft and pointing towards the centre of the earth". The terrain database height corresponding to a set of position coordinates always represents the plumb-bob height. The discrepancies between the measured range to the terrain and the plumb bob range are referred to as slant range errors. These deviations are a function of one or more of the following factors:

- Radar modulation scheme;
- Antenna beamwidth;
- Altitude of aircraft;
- Banking, rolling or pitching manoeuvres;
- Terrain roughness.

Depending on the modulation scheme used, the RADALT measures the distance to a point within its beam having either the shortest reflection time (pulse radars) or the strongest reflected power(FM-CW radars). Figure 4.18 exemplifies the situations that may lead to slant range measurements. Terrain with more variation can lead to slant range errors because of side slopes that provide faster and high-powered radar returns. A large radar footprint increases the probability of measuring slant ranges, because it means that larger terrain areas are illuminated. Pitching, rolling and banking introduce slant range measurements because the radar antenna bore sight no longer points vertically downwards during these manoeuvres. A type of error characteristic only to the FM-CW radar altimeters is terrain averaging. It occurs when the terrain under the aircraft is uneven causing for several different beat frequency to be returned to the frequency counter. The averaging errors have the same effect as the slant range errors.

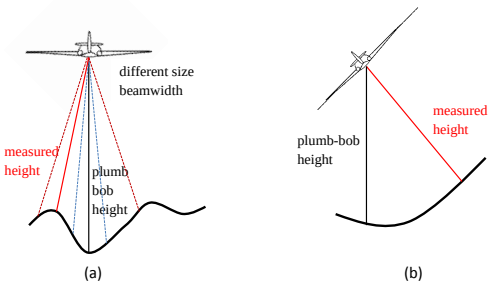


Figure 4.18: Illustration of slant range error measurements caused by: (a)terrain variation and beamwidth size; (b)pitch or rolling manoeuvres

In [14] a method to compensate for slant range errors through processing was proposed. The "spot algorithm" uses the geometry of angles to predict the point where the radar antenna beam illuminates the terrain, calculates the footprint size and chooses the database height within this 'spot' which gives the shortest range to the aircraft. This is far from being a trivial issue, as such an algorithm must consider the circularly distributed horizontal errors and possible horizontal biases in the databases. The performance of the spot algorithm was further analysed in [14] through a series of simulations and real flight tests. The RMS difference between the plumb-bob height and the shortest slant range was studied as a function of height AGL and antenna beamwidth on terrains with different roughness. The conclusion was that as the beamwidth increases the RMS grows in a logarithmic fashion, with more pronounced values at higher altitudes and on terrains with a stronger signature.

Flight tests were used to check the agreement between real-time synthesized terrain profiles and database extracted ones. The tests were performed using two different pulse radars: one with a beamwidth of 35° and one with a beamwidth of 17°. For terrains with limited undulations it was observed that no significant improvements were made when using the spot algorithm. The tests performed on rougher terrains showed a noticeable increase in agreement between the measured profile and the spot terrain profile opposed to the plumb bob profile, especially for the larger beamwidth RADALT measurements.

4.3.2 Analysis of altitude dependant noise errors

4.3.2.1 Impact on the TRN correlation functions

The radar altimeter is used to provide the measured profile, which is correlated with the early and late database extracted profiles during the tracking process. Terrain height measurement errors distort the correlation functions. Figure 4.19 illustrates the different types of distortion caused by measurement noise. The distortions can be grouped into three categories, as follows:

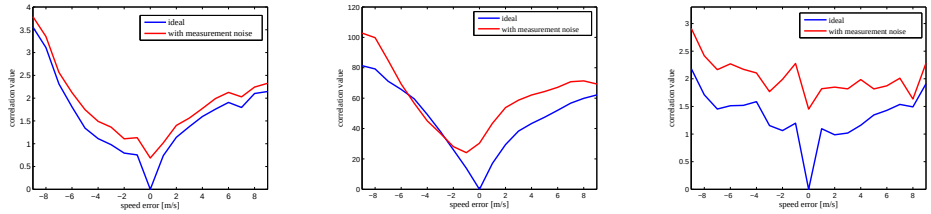
- Only the height of the correlation peak is impacted. The correlation minima differs from zero, but does not change its location. In this case, the tracker is in lock and it is expected for the performance of the tracking algorithm to not be affected. This situation is depicted in figure 4.19(a).

- The correlation minima differs from zero and a shift in the location of the correlation peak appears. This happens because the measured profile has a better correlation with an early or a late profile, rather than the true profile. In this case the tracker remains in lock, with the shift in the peak translating into a reduction in the range over which errors can be tracked. This situation is illustrated in figure 4.19(b).

- The autocorrelation peak is not visible among the side peaks, as illustrated in figure 4.19(c). Due to the high ambiguity, the early late algorithm cannot be performed and it is expected for the tracking loop to lock on a random(erroneous) speed/heading or to lose lock.

Hypothesis

The terrain profile, as found in the database, can be regarded as the 'true' TRN signal. The measured distances represent this signal, distorted by noise. Thus, a comparison between the two would be analogous to the signal-to-noise ratio (SNR) measure used in analog and digital communications (i.e. SNR is a measure of signal



(a) (b) (c)

Figure 4.19: Illustration of the three different types of distortions on the SCF caused by altimeter measurement noise

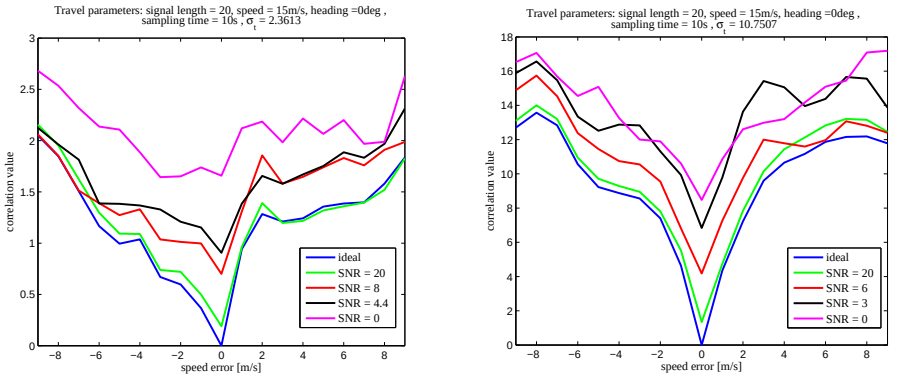
strength relative to background noise). Usually defined as a power ratio, in certain conditions SNR can also be obtained by calculating the square of the amplitudes ratio. Similarly, a signal to noise measure for TRN is introduced as the ratio between the variances of the TRN signal and the noise. The variance of the TRN signal is calculated as the variance of terrain height in the prompt extracted profile. Assuming that the noise errors are Gaussian, with a zero mean and a known standard deviation (σ_n), the variance of the noise is calculated as the square of σ_n. As with RF signals, it is desirable that the energy of the signal is higher relative to the noise. A ratio of 1:1 or lower means that the nominal random errors in the vertical direction introduced by the measurement tool are greater than the variation of the relief. The ambiguity is too large and it becomes nearly impossible to detect horizontal errors. The situation is somewhat analogous to trying to detect un unknown radio signal when the signal is below the noise floor.

With GPS, the signal power coming into the receiver is well below the noise floor. It is only through the de-spreading of the C/A code that SNR increases and the signal power rises above the noise power. By spreading the GPS signal over a wide bandwidth, only a fraction of the desired signal can be corrupted by the narrow band interference. After the de-spreading process, the narrow band noise is spread and the signal power is compressed into a smaller bandwidth. The resulting improvement in SNR is referred to as 'processing gain'. The wider the signal is spread, the higher the processing gain. The C/A code is 1023 bits long and results in a theoretical processing gain of 43 dB (processing gain = 10log(Chip Rate/Data Rate)). Although various imperfections in the correlator may degrade the gain, it is enough to turn the negative SNR environment before correlation into a positive SNR condition after the correlation process.

With TRN, the signal does not contain 1023 uncorrelated samples. The length of the transect is considerably smaller. However, the correlation process will still result in

a processing gain. Although it cannot be calculated, as with GPS, the basic principle remains valid: processing gain can be increased using a longer correlation set. Due to terrain uniqueness and randomness, it is also expected for the processing gain to be dependent on the terrain features. The possible achievable processing gain will depend upon the frequency content of the terrain signal. If the spectrum of noise and signal overlap, then it will not be possible to enhance the SNR through correlation, resulting in a low or possibly no processing gain at all for the signal. If the dominant frequency of the TRN signal is outside of the noise spectrum, the correlation process will average out the noise and increase the signal.

Because terrain varies, simulations have been used to verify the presented hypotheses. First, the dependency between the distortions in the correlation function and the SNR was analysed. The added altitude dependent noise errors were assumed to be Gaussian, with a zero mean and a known σ_n. The height of the autocorrelation peak decreases with the decrease of the SNR. Because terrain signals may have a very dynamic range, we chose to express SNR using a logarithmic scale. As expected, the impact on the correlation function depends on the terrain features. For terrains with a high frequency content, as illustrated in figure 4.20(a), the correlation function decays faster. Empirically it was noted that, for most cases, a critical minimum SNR of at least 5 dB is required. If the terrain has a lower frequency content, the correlation function can handle more noise. A randomly selected example representative for these situations is depicted in figure 4.20(b). Because the dominant frequency of the terrain signal is outside of the noise spectrum, the signal can be recovered even when the SNR is null.



(a) (b)

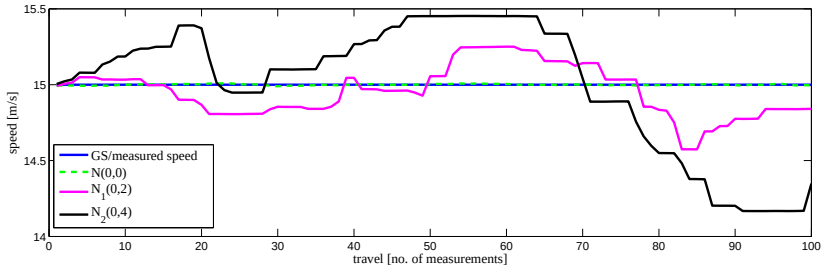
Figure 4.20: Illustration of the dependency between the distortions in the CF and the SNR: (a)for a high frequency content terrain; (b)for a low frequency content terrain

4.3.2.2 Impact on algorithm's performance

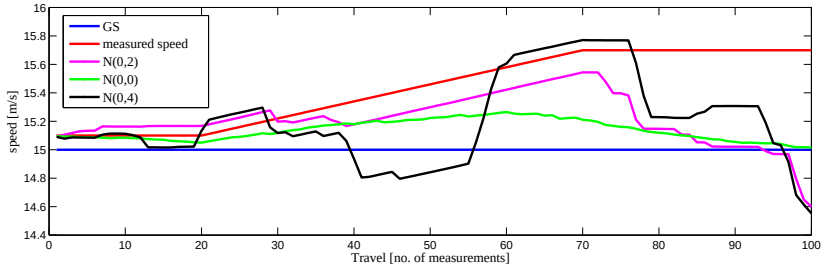
Measurement errors distort the correlation function. The immediate effect of a distortion is expected to be a local decrease in accuracy. However, it is not clear if they also lead to loss of lock. This section analyzes, using Matlab simulations, the impact of measurement noise errors on long term (not for a single snap-shot, but for longer routes). The radar measurement errors were modelled as added noise with a normal distribution. To be able to study the effect of the different types of errors in isolation, two different scenarios have been applied: in the first case it was considered that the measured speed/heading is equal to the ground referenced value; in the second scenario it was considered that sensor readings contain a measurement bias.

Figure 4.21 presents the impact of measurement noise on the performance of the speed tracking loop when the travel route is chosen over a terrain with a high frequency signal content. A route with a length of 100 measurements was considered. Terrain variance is calculated using the prompt profile plus several grid points added in the same direction of travel. The addition is made to encompass any possible speed errors. Noise with normal distribution was added to the measured signal. The SNR for the tested cases is plotted in figure 4.21(c). Even if the actual value of noise is rather low (parameters for noise distribution are given in the legend box; std is measured in meters), it is comparable to terrain signal variation for most of the travel. In other words, the minimum recommended value for SNR is almost constantly exceeded. Thus, this would be representative for the category of 'worst case scenarios'. The performance of the tracking speed clearly degrades when noise is added, as compared to the performance shown in the ideal conditions depicted in green in the figure. Due to the batch correlation, errors have a ripple effect. If the accumulated error caused, at some point, a shift in the correlation peak, it will take several iterations to get rid of this effect. This explains the rather 'fluctuating' estimates seen in the figure, since the accuracy is restored gradually. On the other hand, the same phenomena keeps the tracker in lock despite the large fluctuations observed in the output. The measurement noise was generated in Matlab using a function that generates normally distributed random values. Thus, there was no control over each individual introduced error. The outcome would be slightly different with every performed simulation, even if the parameters of the distribution would be kept identical. In fact, the aim of the analysis was to understand the trend in the impact, rather than to quantify it.

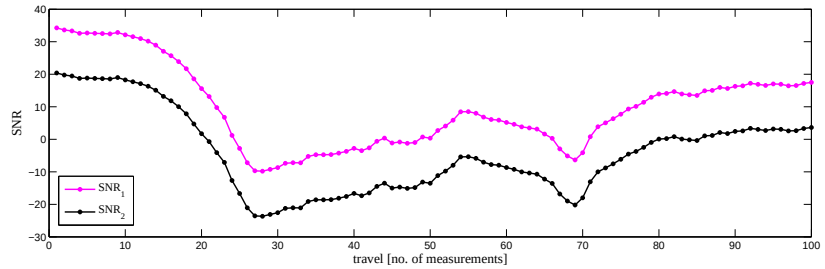
When the travel route is chosen over a rougher terrain (which provides more signal power), the impact of measurement noise is less severe. Figure 4.22 illustrates such an example. The performance of the speed tracking loop when noise is added to the measurements is close to the performance obtained in the ideal conditions when no errors are present. Some fluctuations in the estimations can still be noticed, but (as it



(a)



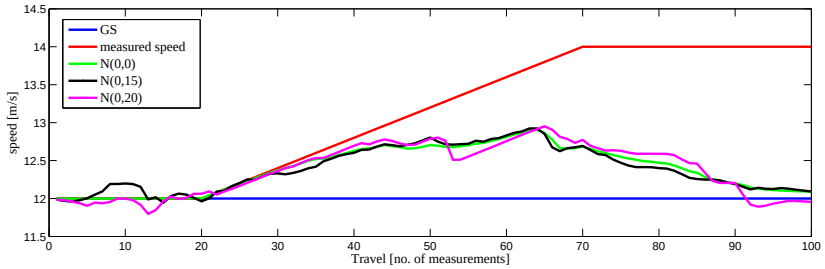
(b)



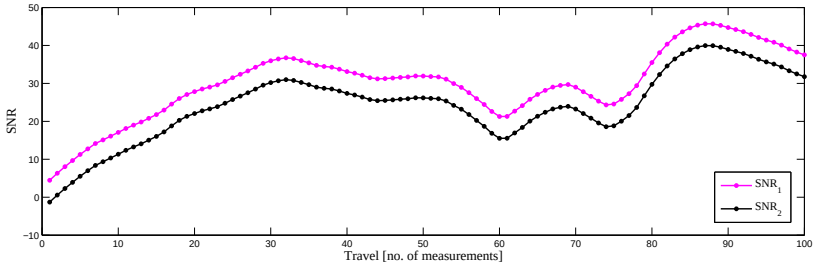
(c)

Figure 4.21: (a),(b)Speed estimation of the tracking loop when noise with distribution N is added to the measurements over a terrain with a high-frequency signal content; (c)SNR comparison

can be read from the axis) their magnitude is rather small. The fluctuations are due to the error propagation in the batch correlation process, as explained in the previous paragraph. Figure 4.23 is used to better explain the relation between SNR and the algorithm performance. From the second plot it can be seen how the SNR varies over the entire travel. Normal distributed noise, with zero mean and different values as standard deviation, was added to the measured profile. For the first part of travel



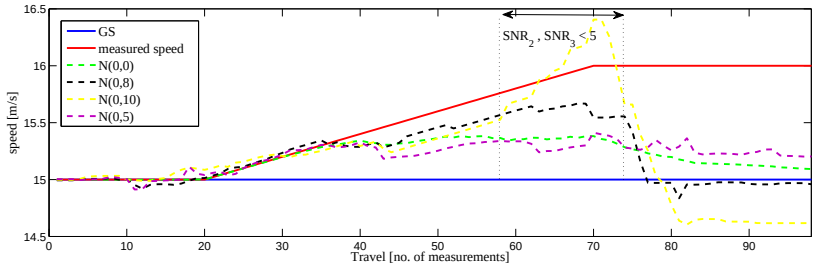
(a)



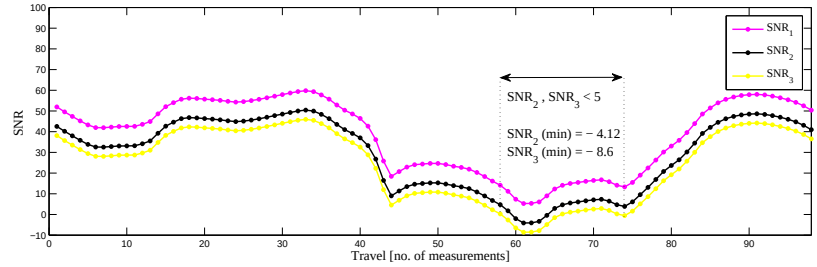
(b)

Figure 4.22: (a)Speed estimation of the tracking loop when noise with distribution N is added to the measurements over a terrain with a low-frequency signal content; (b)SNR comparison

the terrain has an increased roughness and results in a good SNR for all the analyzed cases. As a consequence, the tracking loop response (for all the analyzed cases) closely follows the ideal, no-noise performance. Terrain roughness starts to decrease with measurement x = 40, causing a drop in the SNR. The SNR remains low over the next 20 measurements, however higher than the critical threshold of 5 dB. The impact can be seen in the performance of the trackers, however not immediately but over the course of the following measurements. Starting with x = 58, SNR decreases even further and, for the cases plotted in yellow and black, it reaches negative values. The accuracy of the speed estimates is clearly affected. Further on, even as the SNR increases, the 'ripple' effect can still be seen and it takes about half/full transect length for the algorithm to recover and estimate the constant bias. For the test case plotted in magenta, the SNR was positive and larger than 5 dB for the entire travel, thus the estimates have fluctuations small in magnitude and the over-all performance is comparable to the ideal situation, plotted in green.



(a)



(b)

Figure 4.23: Illustration of the relation between: (a)algorithm performance and (b)SNR

4.3.3 Analysis of slant range errors

The impact of slant range errors on the TRN correlation functions has been analysed using Matlab simulations. Similar to measurement noise, slant range errors distort the correlation function. Figure 4.24 illustrates the different types of distortions encountered. Empirically, it has been noticed that the type of distortion depicted in figure 4.24(b) is predominant.

By comparing figures 4.19 and 4.24, it can been seen that both noise and slant range errors have a similar impact on the TRN correlation function. Thus, most of the conclusions drew from the sensitivity analysis presented in section 4.3.2.2 apply to slant range errors too. Regarding the performance aspects, the slant range errors in the measured profile will also cause fluctuations in the speed/heading estimates, with a magnitude depending on the size of the introduced errors in comparison with the terrain signal variation. However, unlike in the measurement noise case, the slant range errors could not be quantified and no measure (similar to the SNR used in the analysis of noise errors) could be defined.

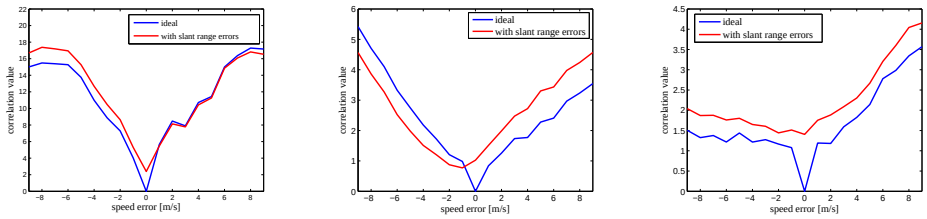


Figure 4.24: Illustration of the three different types of distortions on the SCF caused by slant range errors

4.3.4 Database inaccuracy

Another factor that causes discrepancies between the altimeter measured elevations and the database extracted elevations (thus contributing to terrain measurement errors) is the database inaccuracy. DEMs are normally described by a series of parameters, such as: spatial resolution, horizontal and vertical datum, horizontal circular and vertical linear errors. The accuracy of the database is defined by the last two parameters: circular error probability(CEP) describes the horizontal accuracy for the post-position; linear error probability (LEP) describes the height error accuracy. Additionally, the spatial resolution can also have a significant impact on the nominal error budget of a DEM. A limited DEM resolution can introduce interpolation errors. The accuracy of a database is defined at the grid-points in terms of CEP and LEP . However, these accuracy specifications do not provide information on the accuracies of the geo-spatial points between the grid points extracted by using bilinear or other interpolation techniques. Therefore, the actual error variance might exceed in practice the specified or nominal DEM error variance.

4.3.5 Discussions

4.3.5.1 The impact of terrain measurement errors on the speed and heading estimates

The presence of errors in the terrain measurements make it more difficult to estimate the errors in speed/heading. To better clarify this statement, let us assume that the set of terrain elevation samples is distorted by noise. An increase in the noise will cause less decorrelation during the tracking process (when the profiles are shifted with a certain amount of terrain samples due to speed/heading errors) and this will make it harder to detect the errors in speed or heading. In fact, depending on the noise level, errors up to a certain amount will pass undetected. It would be expected for the

speed tracking loop to be more sensitive because the shifts are made within the same direction, unlike for the heading tracker which correlates in different directions and the decorrelation between subsequent profiles should be more significant. However, this statement cannot be assumed to be valid in all situations because it also depends on terrain characteristics. A quantified relationship between the amount of uncertainty in the elevation errors and the amount of detectable errors in speed/heading has not yet been found. The larger the tracking bandwidth, the less will be the impact of the same amount of noise. On the other hand, it becomes much harder to detect small errors when tracking is performed with a large bandwidth. Simulations have shown that, due to the inaccuracy of the altitude information and the changes in speed/heading, the TRN heading and speed solutions will fluctuate. In turn, these fluctuations cause so much uncertainty that heading/speed error corrections are not possible. The tracking algorithm provides the heading/speed solution that maximizes correlation, which might not be at all times the best solution in terms of accuracy due to the errors in the measurement profile. Thus, despite not being able to deliver instantaneous accurate estimates of speed/heading bias, the tracking algorithm will maintain lock by keeping the bias within the bandwidth and will be able to track speed/heading drifts in the long term.

4.3.5.2 An adaptive configuration for the tracking loops

The analysis presented in section 4.3.2 showed that the impact of altimeter measurement errors on the performance of the TRN algorithm will depend not only on the level of errors alone, but also on the terrain characteristics. In other words, the actual values of errors are not important, rather the ratio between the level of terrain signal (expressed as the variation of heights in the terrain profile) and the level of altimeter measurement errors. For altitude dependent noise errors, a measure of the ratio between terrain signal and noise was defined in the form of SNR. As with RF signals, it is desirable to have a strong SNR. The greater the variation of heights in the terrain profile as compared to the nominal errors introduced by the measuring tool, the higher the SNR. This translates into a correlation function with a higher level of robustness against distortions and, consequently, a better performance of the algorithm. The distortions in the correlation function lead to fluctuations in the estimates of the tracking loops. The magnitude and the frequency of appearance of these fluctuations depend on the SNR. Just as with measurement noise, slant range errors distort the TRN correlation function causing fluctuations in the speed/heading estimates. However, a measure to quantify the impact of slant range errors on the performance of the algorithm(similar to SNR) has not been found. Being a batch correlation algorithm, the error introduced by a new added terrain measurement is averaged over the total TRN signal. Error propagation, as

well as error correction, happen gradually and for this reason the fluctuations in the estimates tend to compensate each other on the average. Thus, to remove the unwanted fluctuations, moving average filtering is applied. An example is shown in figure 4.25. In this scenario it was considered that the speed sensor measurements contain a constant bias. The SNR was higher than 20 dB for the entire travel. The raw response of the tracking loop when noise is added to the measurements is filtered using different fixed length moving average filters. The minimum length used for the filter is half of the length of the batch TRN signal (the TRN signal length is 20 in this example).

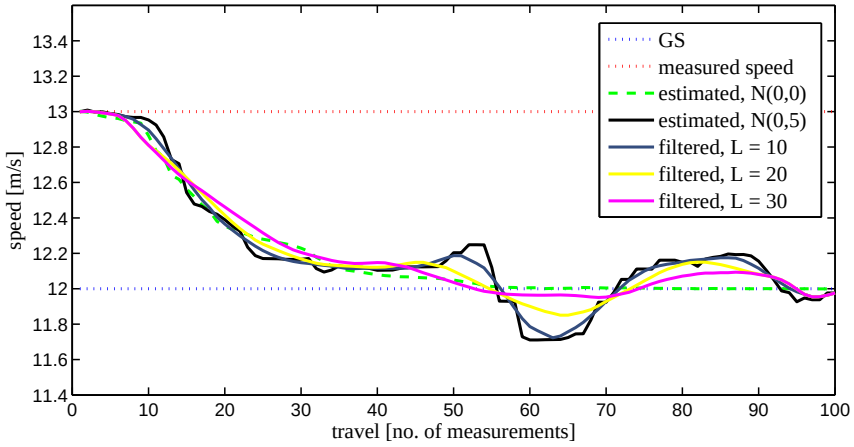


Figure 4.25: Filtered response of a speed tracking loops using different length filters

Average filtering will smooth the estimates only if the fluctuations are small in magnitude. Thus, this strategy works well and should be applied only for the situations in which the difference between subsequent estimates remains within a prior established limit. If the magnitude of the fluctuations increases (as a consequence of a poor SNR or slant range errors), average filtering cannot be used because the large deviations will have a disproportionately large effect on the filtered estimate. Solutions need to be aimed at increasing the terrain signal power (and respectively, the SNR). With GPS, the processing gain of correlation makes it possible to recover the useful signal although the SNR of the received signal at the earth's surface is negative. The processing gain is a function of the length of the code used for correlation. With TRN, the same principle is applicable: the processing gain increases with a longer correlation set. But, as explained in section 4.3.2.2, TRN processing gain is a function of terrain and a quantifiable relationship between the two has not been found. The possible achievable process gain depends on the relationship between the frequency of change

in terrain and frequency of altimeter noise. If the two spectra overlap, the correlation will amplify the signal and the noise at the same time resulting into a weak process gain. If the dominant frequency of terrain is outside of the noise spectrum, correlation will increase the TRN signal strength and it is possible to benefit to the full extent of the processing gain. Based on the expected performance of the radar altimeter and on the properties of terrain, it is possible to predict the SNR. To maintain optimal performance, the TRN tracking loops should be adaptive, meaning that it could be dynamically configured based on the SNR prediction. A SNR higher than 5 dB is considered to be a good performance region (it also depends on the length of the signal). Obviously, the higher the SNR, the better the expected performance of the algorithm in terms of accuracy. For a SNR prediction below the minimum threshold, the option in terms of configuration is to increase the processing gain by lengthening the amount of samples used in the correlation. Although using a longer transect would seem like the better choice, it could be less practical in some situations. The amount of extra samples needed depends on the local terrain characteristics and a longer correlation set will increase the running time of the algorithm. A compromise between accuracy and running time should be settled depending on the application. Alternatively the host vehicle could change course and move to an area with a higher SNR, if freedom of manoeuvre is possible.

4.4 Summary and conclusions

1. When no additional sensor information is used, the bandwidth with which the tracker can follow changes is rather limited.

2. Similar to Doppler aiding in GPS, the tracking loops need to be aided with sensor data to provide sufficient bandwidth for detection of changes in speed/heading. In practice, this means that the new sample points added to the transect are computed using the value measured by the sensor compared to the current established reference.

3. In the context of the proposed TRN application, it is considered that the sensors are self-contained.The raw sensor signals are first processed, instrument errors are calibrated and independent estimates are fed into the TRN algorithm.

4. The total introduced error is modelled as a combined effect of measurement noise and induced bias. The bias error component is further divided into two groups: sensor specific bias (which is a function of sensor performance) and data bias (which is not a function of sensor performance itself, but appears due to the operational environment). The grouping of the bias was needed because

between the two components there is a significant difference in both scale and distribution. Additionally, data bias is to a certain extent predictable.

(a) Given the implementation of the tracking process, the distribution of subsequent changes in error has a higher impact on the performance of the algorithm than the absolute value of the error itself.

(b) Similar to Doppler prediction, data bias caused by flow-fields can be predicted and compensated for. Corrections can be applied to the sensor measurements prior to the use as input to the tracking algorithm.

(c) The impact of speed and heading errors on the performance of the algorithm has been analysed. It was concluded that:

i. Input measurement noise generates noise at the output of the trackers, because zero-mean errors are averaged out in the correlation process.

ii. The tracking algorithm can detect and compensate for an almost constant bias error.

iii. The tracking algorithm can detect and partially compensate for a low frequency change in the bias error.

iv. Errors with a high frequency rate of change are very difficult to identify.

5. Sensitivity to speed and heading errors is dependent on the terrain features. The terrain autocorrelation function is the tool used to convey sensitivity information related to the impact of sensor errors, by looking at the dominant signal characteristic of the terrain. Simulation-based analysis of the terrain autocorrelation function in relation to the performance of the algorithm showed that:

(a) It is possible to distinguish between terrain in which the speed/heading tracking loop will be less or more sensitive to errors in speed/heading.

(b) The tracking algorithm is typically more sensitive to heading errors.

6. Given the properties of the terrain autocorrelation function and the expected error characteristics of the navigation sensors, the performance of the tracking algorithm can be predicted. If not sufficient, the tracking loop can be reconfigured in order to obtain an improved predicted performance. Also, thresholds can be computed for the maximum accepted sensor data uncertainty whilst maintaining lock.

7. In this chapter, the adaptivity concept was further extended by allowing for the tracking loops to be dynamically configured based on the properties of the terrain and the expected performance of the supportive sensors.

8. Two types of altimeter measurement errors were identified: altitude dependent noise errors and slant range errors. Additionally, database inaccuracy and interpolation errors may cause discrepancies between altimeter measurements and database extracted elevations, contributing to the total terrain measurement errors.

9. The impact of noise and slant range errors on the TRN correlation functions has been analysed. It was concluded that both types of errors distort the correlation functions. Distortions are categorized as follows:

   (a) The correlation minima shifts from zero, but does not change its location.

   (b) The correlation minima differs from zero and the location of the correlation peak is shifted.

   (c) The autocorrelation peak is not visible among the side peaks.

10. The impact of measurement noise errors on the performance of the algorithm has been analysed. It was concluded that:

    (a) Similar to RF signals, a signal to noise measure for TRN is introduced as the ratio between the variance of the TRN signal and the variance of noise.

    (b) A good SNR translates into a correlation function with a higher level of robustness against distortions.

    (c) The distortions in the correlation function lead to fluctuations in the estimates. The magnitude and the frequency of appearance of the fluctuations depend on the SNR.

    (d) For a high SNR, the fluctuations in the estimates are rather small. They can be removed by filtering the output of the tracking loop.

    (e) For a low SNR, the fluctuations become large. The option in terms of configuration is to increase the process gain (and the signal power) by lengthening the amount of samples used in the correlation.

11. The impact of slant range errors on the performance of the TRN algorithm is similar to the one of noise, namely causing the appearance of fluctuations in the speed and heading estimates. However, a quantifiable measure (similar to the SNR used in the analysis of noise errors) could not be found.

12. The impact of terrain measurement errors on the accuracy of speed and heading estimates has been analyzed. It was concluded that:

    (a) The presence of errors in the terrain measurements make it more difficult to estimate the errors in speed/heading. This particularly limits the value of the minimum detectable errors.

(b) A quantified relationship between the amount of uncertainty in the elevation errors and the amount of detectable errors in speed/heading has not yet been found.

(c) The inaccuracy of the altitude information together with the speed/heading changes cause so much uncertainty that the tracking algorithm cannot provide solutions beyond a certain accuracy. However, it can maintain lock and track the changes in bias in the long term.

(d) The goal of the algorithm is not to make accurate corrections to speed and heading, but to track the movement of the platform by preventing the biases in the supportive data to pass undetected.

13. Sensitivity to measurement errors is dependent on terrain characteristics. Based on the expected performance of the radar altimeter and on the properties of the terrain, it is possible to predict the expected SNR. If not sufficient, SNR can be improved through processing gain. A method to predict slant range errors has not been found.

14. In this chapter, the adaptivity concept was further extended by allowing for the tracking loops to be dynamically configured based on the SNR.

Chapter 5

Evaluation using real sensor data

5.1 Introduction

In the previous chapter, sensor data was simulated to explore the relation between performance of the algorithm and data quality. In this chapter, two different sets of recorded sensor data are used to evaluate the performance of the proposed algorithm for two particular types of sensors: a radar altimeter and a laser altimeter. Both sets of data were provided by Ohio University. The radar altimeter data was collected in Asheville (AVL), North Carolina. The laser altimeter flight tests were conducted in the vicinity of the Ohio University Airport Albany(KUNI). To provide the reader with an overview of how the available data was collected, a brief description of the flight test setups is provided. Next, the data is analysed. Following this, the results obtained from using the data as input to the TRN algorithm are presented. The chapter concludes with a discussion on the challenges encountered with the processing of real data.

5.2 Radar altimeter flight data

5.2.1 Flight test equipment overview

The flight tests performed at AVL by Ohio University took place on September 26th and 27th 2000. A Douglas DC-3 aircraft was used. The aircraft was equipped with a prototype Synthetic Vision System (SVS) using the Delft Program for Hybridized Instrumentation and Navigation Systems (DELPHINS) display software to generate the Visual Virtual Environment. Sensor information was obtained from a Honeywell Inertial Reference System. Height AGL measurements were obtained from

a Honeywell HG8505DA01 Radar Altimeter. The specifications of the RADALT are provided in table 5.1. A Kinematic GPS was used for guidance, providing the geographical coordinates corresponding to the terrain measurements. Figure 5.1 illustrates how information is provided to the tracking algorithm.

Table 5.1: Specifications for Honeywell HG8505DA01 Radar Altimeter

Specified altitude accuracy	± 3 feet plus 1
Modulation	Pulse
Pulse repetition frequency	25 kHz
Bandwidth 3 dB	17 degrees

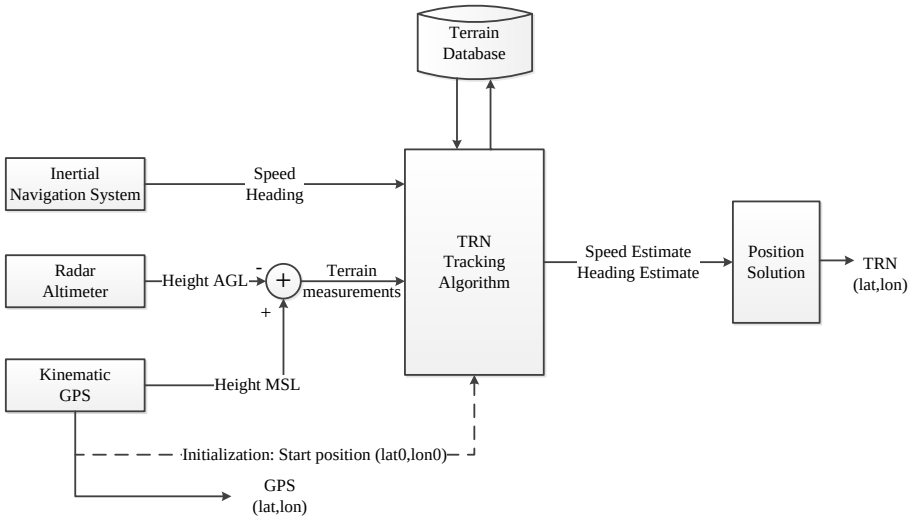


Figure 5.1: TRN algorithm: data inputs and interactions

5.2.2 Terrain database

For the evaluation of the AVL flight test data, Shuttle Radar Topography Mission (SRTM) digital elevation maps were used. The SRTM tile needed for the analysis of this data is illustrated in figure 5.2. The resolution of the database is of 1 arc second.

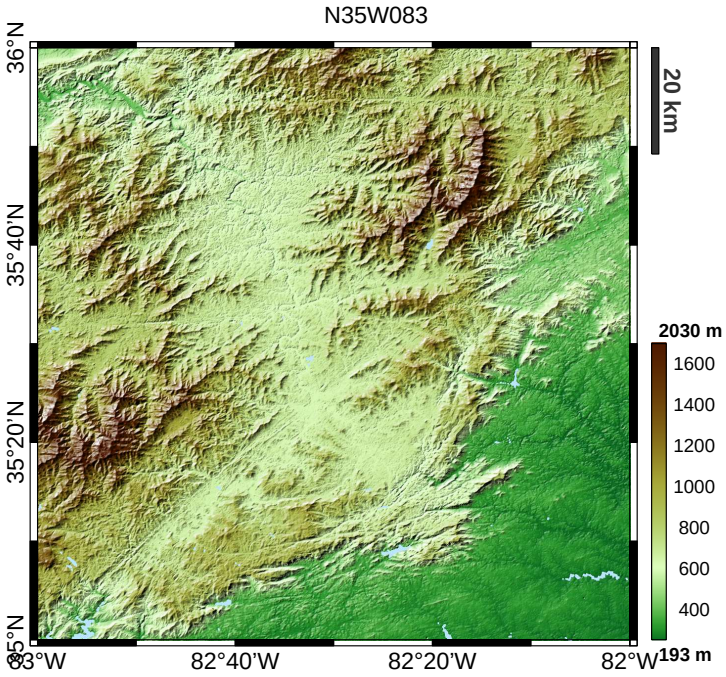


Figure 5.2: Illustration of the SRTM tile used as database

Brief introduction to SRTM DEMs

The SRTM data set represent a near-global DEM of the Earth obtained using radar interferometry. This was an international project resulted from a collaborative effort by the NASA and NGA with the participation of German and Italian space agencies. The data was collected during an 11-day mission in February of 2000. The resulting databases cover the Earth's terrain from 56 degrees South latitude to 60 degrees North Latitude. The elevation models are arranged into tiles, each covering one degree of latitude and one degree of longitude, named according to their south-western corners. The tiles corresponding to the United States territory have a resolution of 1 arcsecond (30 m) and the rest have a resolution of 3 arcsecond (90 m). The dimensions of a 1 arc second tile are 3601 x 3601. A 3 arcsecond tile has 1201 rows and 1201 columns. The vertical datum is the MSL defined by the Earth Gravitational Model (EGM) 1996 and the horizontal datum is World Geodetic System (WGS) 84. The accuracy specifications for the 1 arcsecond tile are: horizontal accuracy of 20 m (90% CEP)

and vertical accuracy of 16 m (90% LEP). More details on SRTM data can be found in [70].

5.2.3 Description of the route

To evaluate the performance of the TRN algorithm, a route containing 1050 measurements was used. The measurements were taken at a sampling time of 1s. Given the average speed of the aircraft, it translates to a total distance of approximately 81 km. The average distance between two consecutive samples is approximately 80 m. The geographical coordinates provided by the Kinematic GPS were considered to represent the truth trajectory. Figure 5.3 illustrates the agreement between the RADALT measurements and the plumb-bob profile extracted from the database at the location indicated by the GPS coordinates. Cubic interpolation was used to estimate the elevation for locations located between the data grid points. A statistic of the disagreement between the measurements and plumb-bob profile is provided in figure 5.4(a). Although this function does not have a normal distribution, the mean and standard deviation were calculated and can be read in table 5.2. These values are not used in the algorithm, they only have an indicative purpose. In Chapter 4 the terrain measurement errors were modelled as altitude dependent noise errors. Given the RADALT altitude accuracy specified in table 5.1, the distribution of the expected terrain measurement errors could be calculated. From the comparison of the real and expected terrain measurement error distributions, as illustrated in figure 5.4(b), it can be seen that the actual measurement errors have a much larger variance than expected. Additionally, the actual measurement errors show a large displacement in the mean. The terrain measurement errors obtained from the real data are the sum of noise, slant range errors and database inaccuracy (Conclusion 4.8), thus it was expected to see larger values in the actual error distribution. However it is unlikely that such a large discrepancy can be caused by slant range errors alone. Conclusion 4.12.a states that the presence of errors in terrain measurements make it more difficult to estimate the errors in speed and heading. Given this distribution, it can be anticipated that the accuracy of the TRN estimates will be drastically impacted.

Table 5.2: Parameters of best normal fit distribution for RADALT errors

Mean [m]	Std [m]
23,13	33,21

Speed and heading truth profiles were calculated from the GPS coordinates. For each pair of consecutive GPS coordinates, a geodesic arc length and an azimuth were calculated. The WGS84 ellipsoid model was used. Knowing the time elapsed between

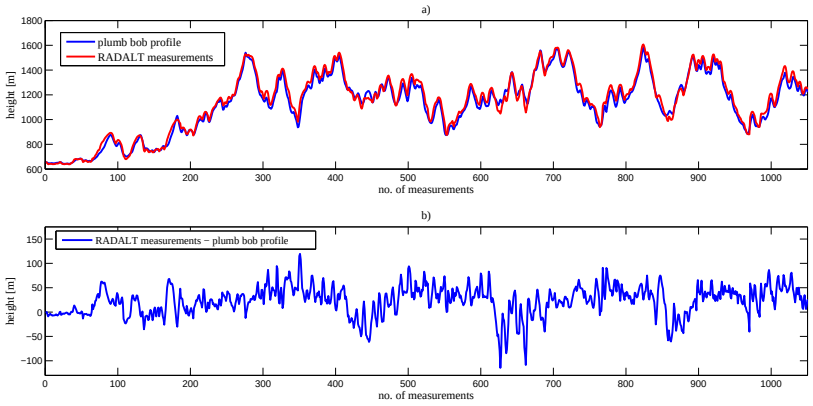
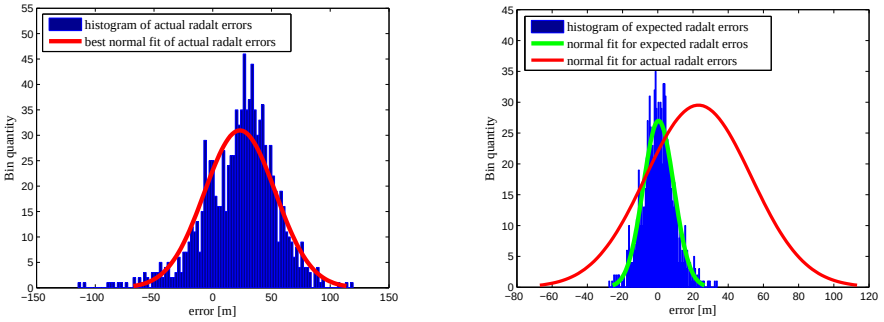


Figure 5.3: Agreement between the RADALT measurements and the truth profile: (a)absolute values; (b)difference



(a) (b)

Figure 5.4: (a)Histogram of difference between terrain measurements and the plumb bob profile with best normal fit distribution overlay; (b)Comparison with the expected terrain measurement errors

consecutive GPS measurements, speed was obtained from the arc length. This speed profile should provide the closest estimation of the GS and will be further referred to under this name. The heading calculated from the azimuth should provide the closest estimation of the track and will be further referred to under this name. Figure 5.5 illustrates the difference between the GS and the IRS indicated speed. Figure 5.6 illustrates the difference between the track and the IRS indicated heading. As summarized in conclusion 4.4, the total introduced error of a supportive sensor is

modelled as a combined effect of measurement noise and induced bias: sensor specific bias and data bias. The IRS measured speed is an estimation of the GS. Thus, it contains no data bias and the error illustrated in figure 5.5 is a combined effect of noise and specific bias only. From figure 5.6 it can be noticed that the difference between track and heading changes as a function of heading, indicating the presence of a flow field (i.e. a data bias). However, the data set did not provide any information on the TAS. With TAS unknown, it is impossible to test the ability of the algorithm to track a changing speed data bias, unless artificially introduced. Due to the same reason, the bias caused by the flow field cannot be predicted to verify with real data the claim from conclusion 4.4b. Figure 5.7 illustrates the trajectory that would be obtained in a dead reckoning mode (using only the measured speed and heading). The profile was plotted using Google EarthTM mapping service. When compared to the truth trajectory, a position drift would be experienced in a standalone IRS mode. The error in position grows linearly, reaching almost 3 km after 1050 seconds of travel.

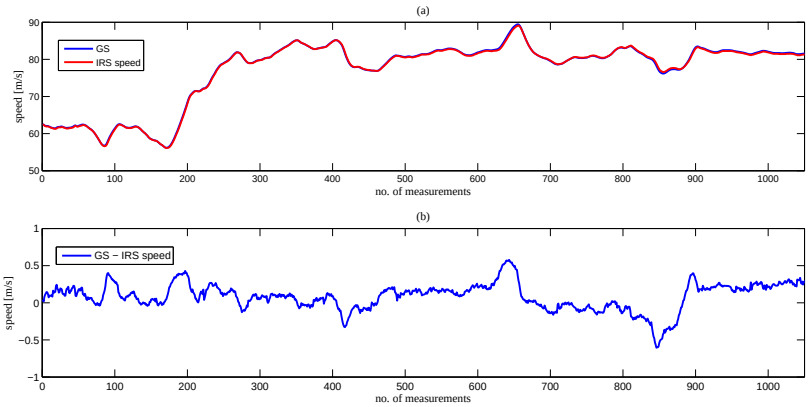


Figure 5.5: Comparison between GS and measured speed: (a)absolute values; (b)difference

5.2.4 Results

Conclusions 4.1 and 4.2 summarize why supportive sensor information is needed for the TRN algorithm: in the absence of sensor information, the bandwidth with which an unaided tracking loop can follow changes is rather limited and a significant growth of the searched parameter will put the tracker in a saturation state. To verify these claims, the performance of the TRN algorithm was evaluated when only heading sensor data was provided, leaving the speed tracking loop in

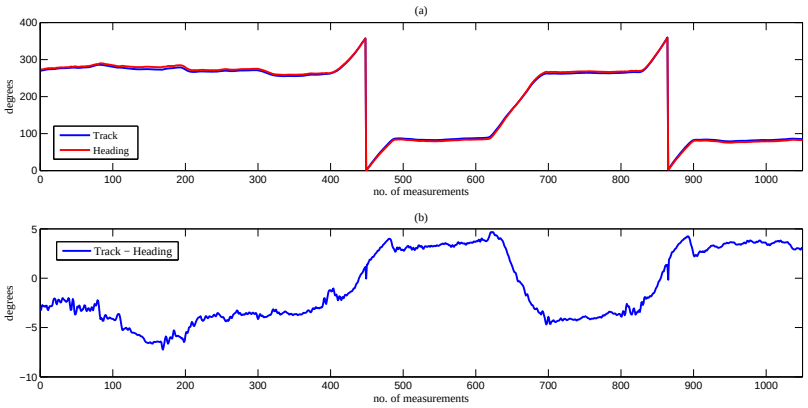


Figure 5.6: Comparison between track and measured heading: (a)absolute values; (b)difference

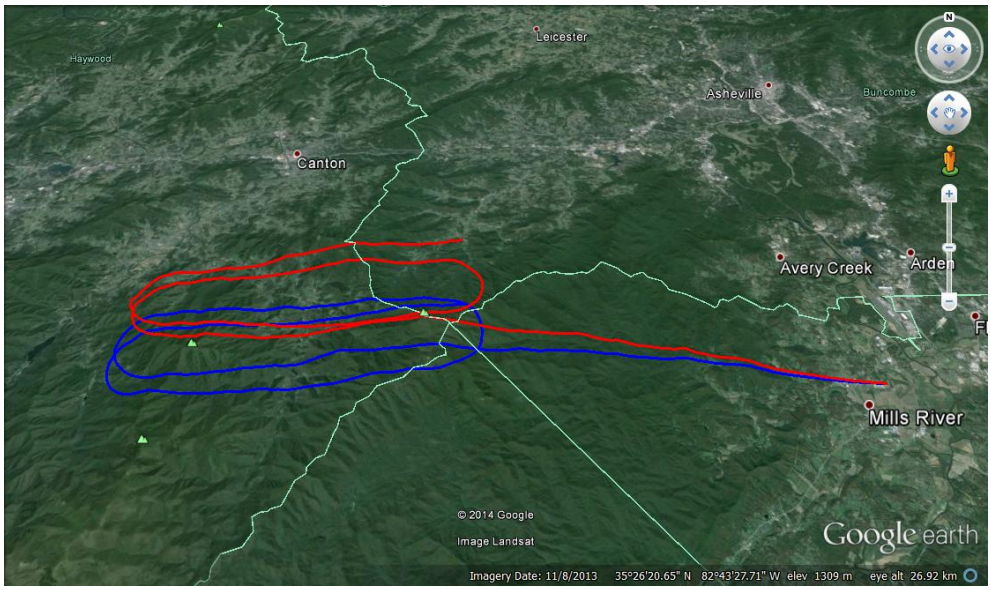


Figure 5.7: Comparison between GPS(blue) and IRS(red) trajectories

an unaided implementation. Figure 5.8 illustrates the TRN speed estimate. As it can be seen, the ground speed is rather stable for the first part of the travel.

The unaided implementation of the speed tracker can follow these small changes. After measurement 150, the speed increases significantly causing the tracking loop to eventually lose lock.

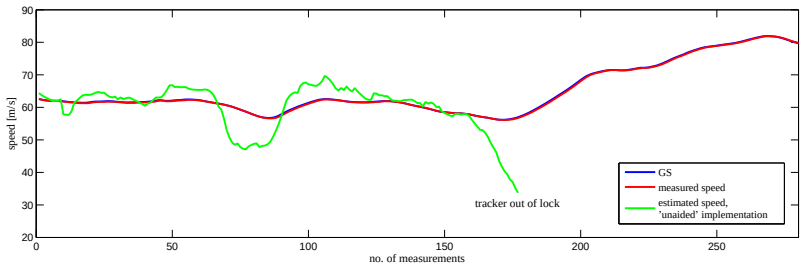


Figure 5.8: TRN speed estimate for the un-aided implementation

Next, the performance of the tracking algorithm was evaluated when both speed and heading sensor information are provided as input to the system. No acquisition phase was implemented. Instead, it was considered that the starting point and the starting array of speed and heading changes are accurately known. In the simulation, the arrays were provided from the truth references. In accordance with the theory from chapter 3, the spacing in the speed/heading tracking loop was continuously adapted to the properties of the speed/heading correlation function. The spacing can only be a multiple of the resolution of the correlation function. The resolution of the SCF and HCF have been decided according to the explanations presented in section 4.2.3.1 and section 4.2.3.2. The correlator threshold is fixed (with different values for the different tracking loops; slightly larger ones for the intermediate tracking loops), with an additional condition to break the tracking process if convergence is not reached within 100 iterations. Table 5.3 provides the values assigned to all parameters of the algorithm.

The TRN position estimates are shown in a plan view in figure 5.9. The kinematic GPS positions are used as the truth reference. The TRN estimates follow the truth trajectory, avoiding the gradual drift that would put the aircraft off path in an IRS standalone mode. Figure 5.10(a) shows the position error with respect to the GPS truth trajectory. The error was computed as the geodesic arclength between the truth geographical coordinates and the corresponding estimated coordinates. The distribution of the position error is illustrated in figure 5.10(b). The TRN speed estimate is illustrated in figure 5.11 and the TRN heading estimate in figure 5.12(a). Due to the large vertical scale (the transition from 360 to 0), it is hard to discern the details in figure 5.12(a). For this reason, the difference between the TRN heading

estimate and the track is illustrated in figure 5.12(b). Both speed and heading estimates provided by the tracker have large fluctuations. This effect is attributed to the errors in terrain measurements (conclusions 4.10 and 4.11). The obtained results are in accordance to the theoretical analysis provided in chapter 4 (conclusion 4.12). The outcome of the speed tracking loop is not the best speed estimation, but rather the speed for which the correlation is maximized given the uncertainty in terrain measurements. In this way the discriminality in the heading tracker is maximized. Although not very accurate in results, the TRN algorithm remains in lock and is able to follow the trajectory of the aircraft, preventing the drift.

Table 5.3: TRN Tracking parameters

Parameter		**Value**	**Comments**
N(transect length)		25 samples	On an average the TRN signal has 2 km and contains about 66 database grid points
1st speed tracking loop	spacing	adaptive	Multiple of the SCF resolution.It can take one of the following values: [0.48; 0.96; 1.44; 1.92; 2.4] m/s
	SCF resolution	0.48[m/s] 10 points	The minimum spacing (0.5 m/s) is equal to a shift in the test profile of database resolution points The maximum spacing (2.4 m/s) is equal to a shift in the test profile of 2 database resolution points.
	threshold	0.005	Fixed, break at 101th iteration
2nd speed tracking loop	spacing	0.24 [m/s]	Fixed
	threshold	0.002	Fixed, break at 101th iteration
1st heading tracking loop	spacing	adaptive	Multiple of the SCF resolution. It can take one of the following values:[1; 2; 3; 4; 5]°
	HCF resolution	1[°] 10 points	The minimum spacing (1 °) is, with the average speed, equal to a shift of about 1 database resolution points The maximum spacing (5 °) is, with the average speed, equal to a shift of about 5 database resolution points.
	threshold	0.03	Fixed, break at 101th iteration
2nd heading tracking loop	spacing	0.5 °	Fixed
	threshold	0.003	Fixed, break at 101th iteration

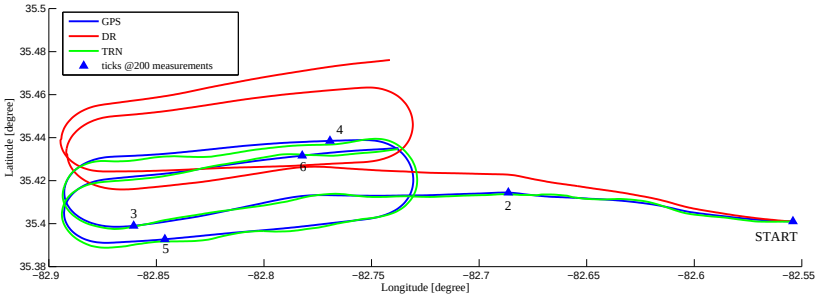
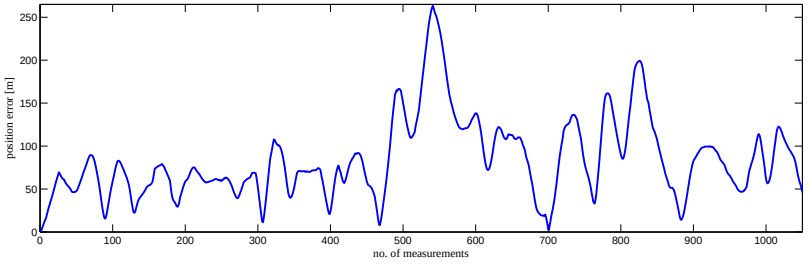
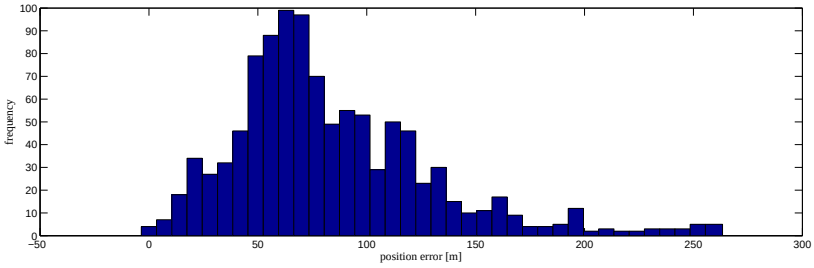


Figure 5.9: TRN position estimates using RADALT data



(a)



(b)

Figure 5.10: Position error: (a)absolute values; (b)histogram

The IRS measured speed contains no data bias. In order to test the ability of the algorithm to track a changing speed data bias, simulations were performed with an artificial bias added to the measured speed. Figure 5.13 illustrates the introduced bias as well as the obtained TRN speed estimate.

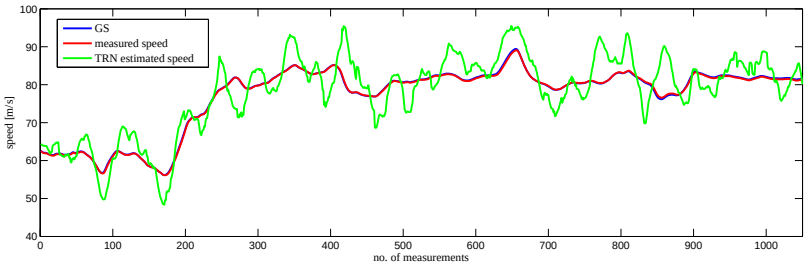
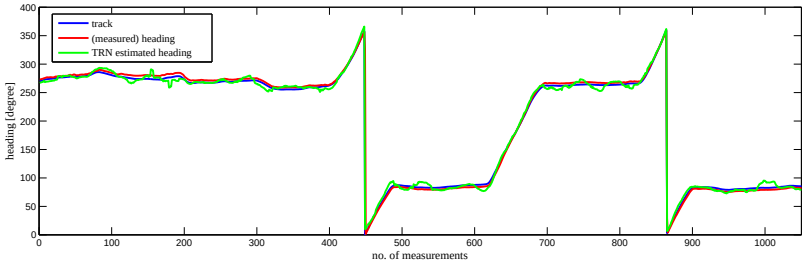
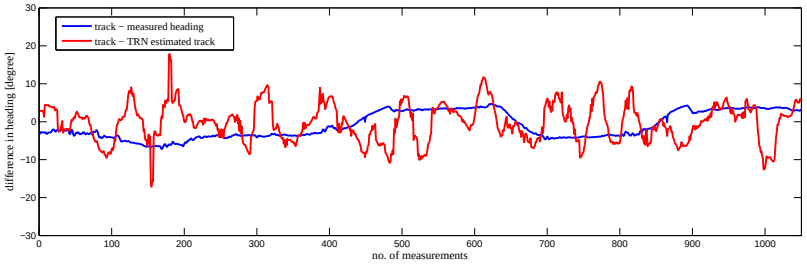


Figure 5.11: TRN speed estimates using RADALT data



(a)



(b)

Figure 5.12: TRN heading estimates using RADALT data:(a)absolute values; (b)difference

During the turning manoeuvres there is a larger likelihood of slant range errors, which may lead to a decrease in performance. In the results presented so far no roll correction has been applied. The following algorithm was implemented for roll correction: 1) while the roll is smaller than half of the beamwidth, the radalt will still measure the plumb-bob height and no correction is necessary; 2) when the roll

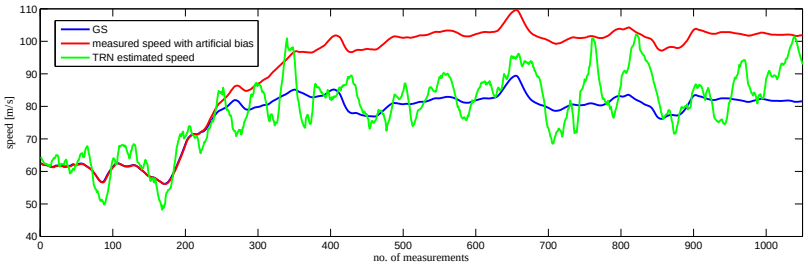


Figure 5.13: TRN speed estimate when an artificial data bias is added to the speed measurements

becomes larger than this angle, the closest return will be a slant range error and roll correction is applied. Figure 5.14 illustrates a plan view of the TRN position estimates when roll correction/compensation was applied. Figure 5.15 zooms in on each of the three turns, providing also the position error. The performance of the TRN algorithm is improved and more accurate estimates are obtained. During the first turn, the roll correction introduces a small deviation from the GPS trajectory. This effect can be attributed to terrain characteristics. However, for the next two turns the roll compensation algorithm corrects the offset in the TRN position estimates, illustrated in the green line, being able to estimate the truth trajectory with a better accuracy.

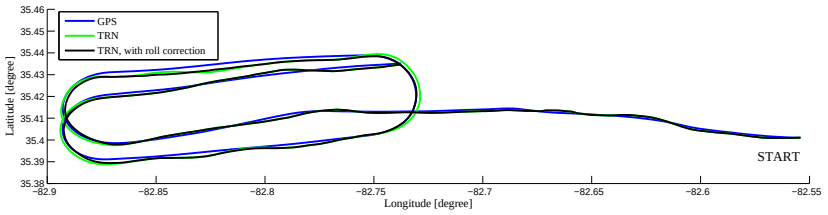


Figure 5.14: TRN position estimates with roll compensation using RADALT data

Conclusion 4.10 explains the relation between the SNR and the fluctuations in

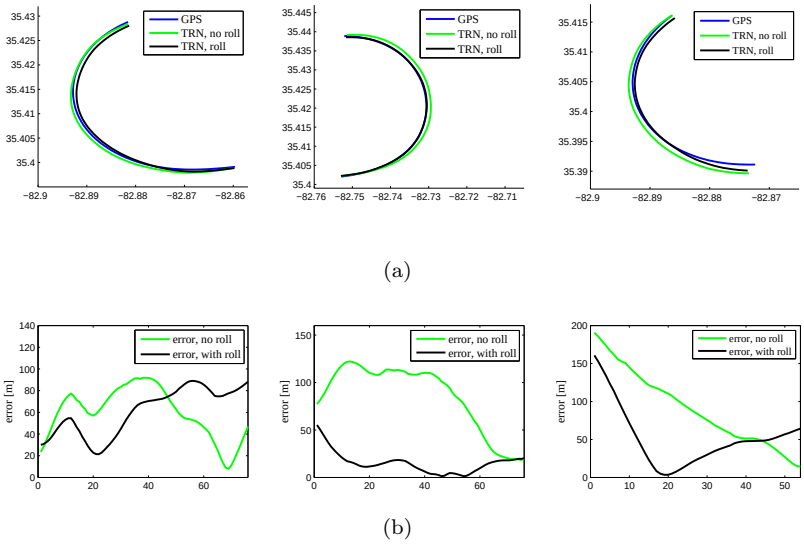


Figure 5.15: Zoomed-in illustration of TRN position estimates with roll compensation using RADALT data: (a)plan view; (b)position error

the speed/heading estimates. In figure 5.16 the SNR for this trajectory is plotted for a TRN signal of 25 samples (the length used in the simulations for which results have been presented so far) and for a TRN signal double in length. The SNR is calculated as described in chapter 4. For the measurement noise statistics, the values from table 5.2 were used. Note that the measurement errors described with figure 5.3 and 5.4 do not have a normal distribution. Thus, figure 5.16 is only indicative of the real SNR. Because of the processing gain, it would be expected that the overall performance of the TRN algorithm is improved when the length of the TRN signal is doubled (Conclusion 4.10.e). However, figure 5.16 shows that, even with a signal of 50 samples, the SNR will sometimes drop under the recommended threshold of 5 db. Figures 5.17 to 5.19 illustrate the position, speed and heading estimates obtained when a TRN signal of 50 samples is used. The effect of a larger signal can be best observed from the speed and heading estimates, where a decrease in the magnitude of fluctuations is clearly visible. Figure 5.17(b) illustrates how the distribution of the position error changes when doubling the length of the signal (normal best fits were used and not the actual histograms).

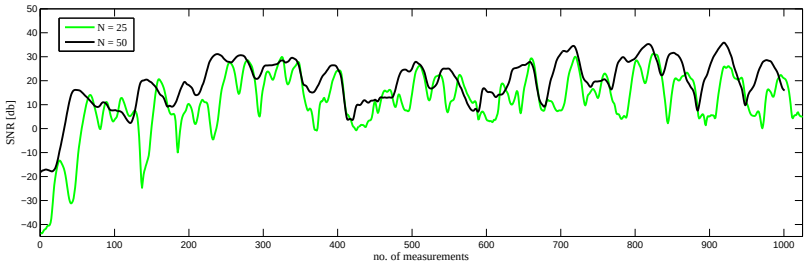
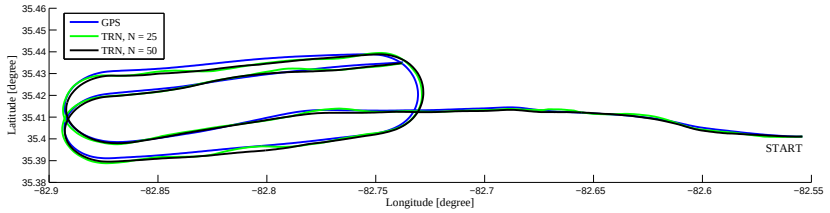
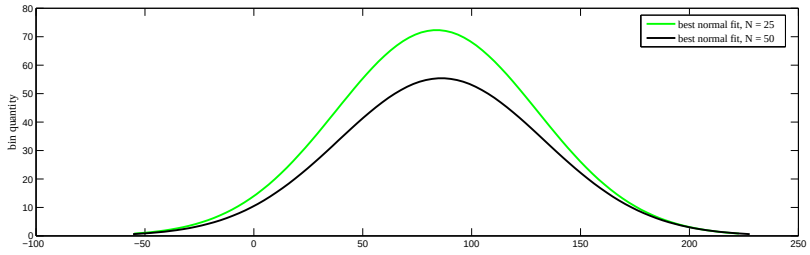


Figure 5.16: SNR for the RADALT TRN signal using different transect lengths



(a)



(b)

Figure 5.17: TRN position estimates using different length signals: (a)plan view; (b)normal best fit distributions for the position error

5.3 Laser scanner flight data

The second set of real data used to test the performance of the TRN algorithm was obtained from a laser altimeter flight campaign. The flight tests were not performed in the context of this research project and the data were provided on the basis of a

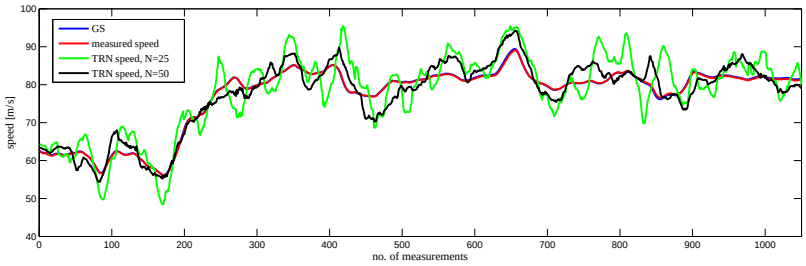
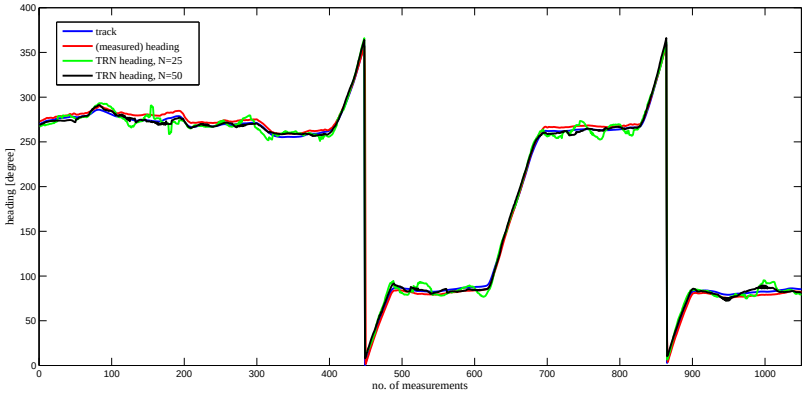
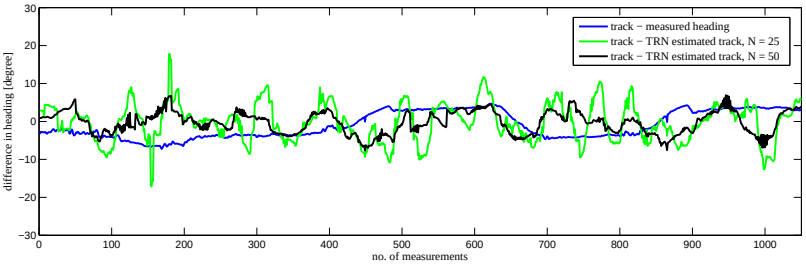


Figure 5.18: TRN speed estimates using different length signals



(a)



(b)

Figure 5.19: TRN heading estimates using different length signals (a)absolute values; (b)difference

cooperation with Ohio University. As a consequence, the format of the data was not optimal for the TRN algorithm and further processing was required. The processing applied would not be possible, nor desirable in a real situation and the purpose was purely theoretical. The goal of using laser data was not to assess the performance of the algorithm when using this type of data, but to determine whether some issues caused by RADALT limitations (due to beamwidth, roll, slant range errors etc.) can be overcome with the use of better quality data. Additionally, LIDAR data has specific limitations which have not been discussed in this section. This is beyond the scope of this thesis, but in a scenario in which the TRN algorithm would use LIDAR data, these specific limitations must be addressed.

5.3.1 Flight test equipment overview

The flight test was conducted over Ohio University's Airport (KUNI) in Albany, OH and the main campus of Ohio University in Athens, OH. For the flight campaign, a Douglas DC-3 aircraft was used. The hardware used included: a Honeywell Navigation Grade INS (HG1150), two NovAtel OEM4 GPS receivers and a RIEGL Medium Range ALS (LMS-Q280i). Each of these sensors is described in the following sub-sections. All technical details regarding the flight test were taken from [16].

Honeywell HG1150 Navigation Grade INS

This INS uses ring laser gyros for angular measurements. During the flight test, the attitude from the INS was treated as accurate (specifically the roll and pitch angles, the heading does have some drift). Position accuracy typically showed 1 nmi/hr drift performance. The INS provided positions and altitude were used in ALS footprint geo-referencing.

NovAtel OEM4 GPS receiver

Two GPS antennas were mounted on the aircraft fuselage along its centre line, one on the front and another towards the rear of the aircraft. A GPS/WAAS receiver connected to the rear antenna was used to collect pseudo-range, carrier phase, ephemeris and timing data to be used in post-processing and formation of the truth trajectory. The second GPS was used to synchronize the clock of the processor that was used in time-tagging each of the sensor measurements. The NovAtel receiver logs were processed with a commercial kinematic processing software.

RIEGL LMS-Q280i Medium Range ALS

The Riegel LMS-Q280i is a medium range, pulsed ALS operating in the near infrared frequency, designated as Class 1 for eye-safe operation. The ALS, as seen from outside of the aircraft, is shown in Figure 5.20. The laser beam is deflected using a polygonal

rotating mirror that provides the scanning mechanism. This creates a 'candy-cane or striped' laser measurement distribution. The cross-track spacing is a function of the laser PRF, the scanning rate and the height of the aircraft. The along-track measurement spacing is a function of the speed of the aircraft and scanning rate. The technical specifications of the instrument are provided in table 5.4 .

Table 5.4: Technical specifications for RIEGL LMS-Q280i Medium Range ALS

Scanner	
Scanning mechanism	Rotating polygon mirror
Number of mirror facets	4
Scan angle range	45 degrees
Angular movement	Linear
Scan speed	5 100 lines/s
Laser Rangefinder	
Measurement principle	Single shot time-of-flight
Measurement range	1200 m
Measurement accuracy	20 mm (1 σ value)
Pulse repetition rate	Typ. 25000 Hz
Laser wavelength	Near infrared
Laser beam divergence	0.5 milli-radian
Eye safety class according to IEC6085-1:2001	Class 3B laser product (under continuous scanning)

Brief introduction to ALS systems

An ALS system consists of a laser scanner and a position and orientation system (POS), typically realised by an integrated DGPS and an IMU. The laser scanner sweeps cross-track to the aircraft, taking hundreds of range measurements per sweep. The POS unit records the position, height and attitude of the airborne platform. Typically, the DGPS can achieve cm level accuracy. The IMU data are post-processed with the DGPS solution to solve for IMU biases and drifts.

The laser scanner is composed from the following units: laser ranger, opto-mechanical scanner, control and processing unit. The laser rangefinder consists of the laser emitter and receiver, typically co-located. Two major ranging principles are applied: pulsed ranging principle and ranging by measuring the phase difference between the transmitted and received signal. The scanning mechanism consists of moving mirrors that deflect the laser pulses in a predefined pattern. Different kinds of mirror assemblies can create various scan patterns, some of which are: unidirectional pattern, bidirectional pattern, circle pattern, elliptical pattern. The control and processing unit handles pulse timing and calculates measurement observables. More details on airborne laser scanning operating principles, as well as on the main ALS error sources can be found in [19], [15].

Figure 5.20: ALS aperture window on the DC-3 aircraft

5.3.2 LIDAR generated DSM

The State of Ohio, in partnership with Local and Federal government agencies, developed a program to provide high-resolution digital imagery and LIDAR datasets for state government entities and the general public. The program, entitled The Ohio Statewide Imagery Program (OSIP), collected the high resolution imagery and the elevation datasets from 2006 to 2008. The colour imagery dataset was produced at a 6 inch resolution. The LIDAR dataset was taken with an average laser pulse spacing of 7 feet (approx. 2 m) and with an accuracy of 1 foot [71]. Ohio University has provided us three consecutive database tiles, each tile covering approximately a 1.5 x 1.5 km2 area. This covers only a small part of the entire trajectory, as it will be discussed in the following section. The OSIP imagery data of the three tiles is illustrated in figure 5.21. It should be noted that the DSM used was not processed to remove features, such as trees or buildings.

Brief introduction to LIDAR generated DSMs

The large number of data points generated using the LIDAR technique can be used to produce high-resolution models of the surface topography, with an accuracy and a detail level that could not be achieved before. Today's LIDAR systems are capable of

measuring multiple returns from one laser pulse, allowing the detection of subtle topographic features, as well as the separation of vegetation from buildings and terrain. Software packages exist which separate the feature data from the terrain data, creating separate features databases and digital terrain maps (DTM). A DTM is defined in [15] as a DSM minus feature, or bare earth model of the terrain.

(a) (b) (c)

Figure 5.21: LIDAR generated DSM: (a)Tile 1, (b)Tile 2, (c)Tile 3

5.3.3 Description of the route

The post processed GPS data is considered as the truth reference for the flight trajectory. Figure 5.22 illustrates the entire truth trajectory of the flight, overlaid on a terrain map obtained from the Google Earth mapping service using the Google Earth Toolbox for MATLAB. The route for which databases were available is plotted in red line, in the same figure. The route contains 700 measurements, taken with a sampling time of 0.1s. A velocity and an orientation profile were derived from the GPS coordinates. For each pair of consecutive GPS coordinates the geodesic arc length and azimuth were calculated. Using them, the velocity and the orientation profiles were deduced. The WGS84 ellipsoid model was used. Although they may still include errors with respect to the actual DC-3 trajectory, they are considered to be the 'truth' GS, respectively the 'truth' track. Figure 5.24 illustrates the difference between the GS and the INS indicated velocity. Figure 5.23 illustrates the difference between the track and the INS indicated heading. The spikes noticed in the 'truth' GS and track are a result of the differentiation of subsequent position estimates in the context of the high resolution of the map. The profile obtained from a standalone INS mode is plotted in figure 5.25 using Google Earth mapping service. As explained in section 5.3.1, the scanning profile of the ALS is defined by time-tags and scan angles within the ALS frame. The configuration for the RIEGL LMS-Q280i ALS was provided in

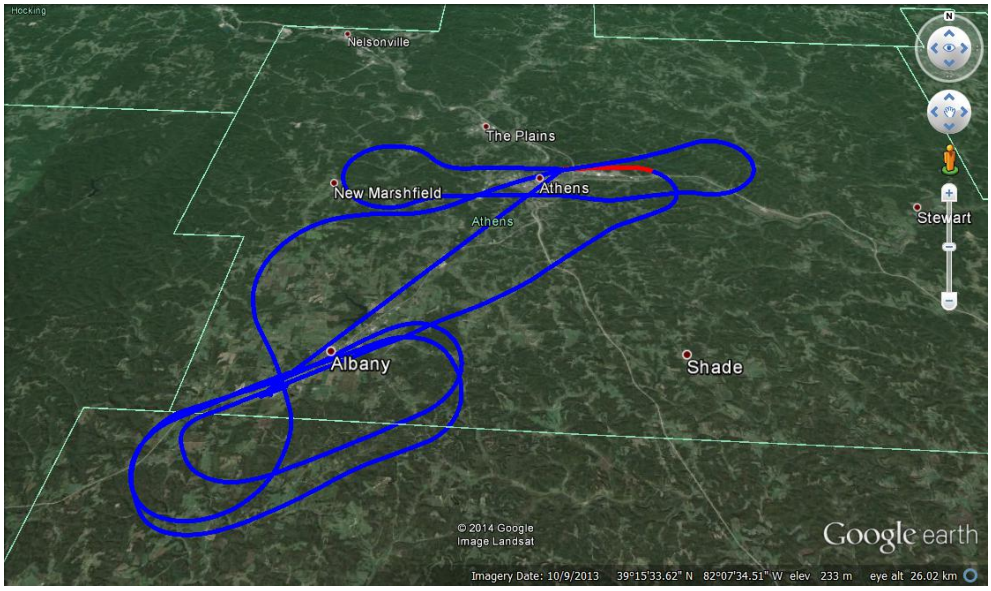


Figure 5.22: ALS Flight trajectory

table 5.4. In this configuration, assuming the altitude of flight and a GS of 60 m/s, the distance between scans in the along-track direction is approximately 2 m and the distance between cross-track scans is approximately 1 m, as illustrated in figure 5.26.

The only form of TRN signal supported by the current implementation of the tracking algorithm is an array of terrain measurements taken in the along-track direction. Given the ALS scanning profile, data had to be processed to form a terrain signal that would fit the needs of the tracking algorithm. The ALS measured profile for the TRN algorithm was formed by taking each point with the coordinates specified by the GPS and averaging all ALS measurements that lay within a circle having its center in the GPS provided position and the radius R. Figure 5.27 illustrates the ALS measurement profile obtained when using different values for R. A radius of 2 m proved to be the minimum possible, as going below this value resulted into sometimes having no ALS measurements within the search area.

The structure of the data in the original DSM is of scattered type, meaning that the points (latitude, longitude, altitude) have no structure or order between their relative locations. The scattered data could be used directly in the TRN algorithm. This would result into a larger level of details. However, to match the ALS measurements created resolution, we have decreased the database resolution. To obtain this, the

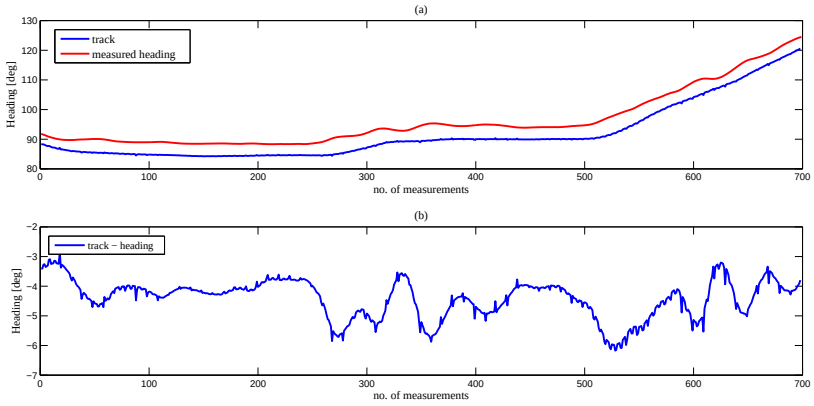


Figure 5.23: Comparison between track and INS measured heading: (a)absolute values, (b)difference

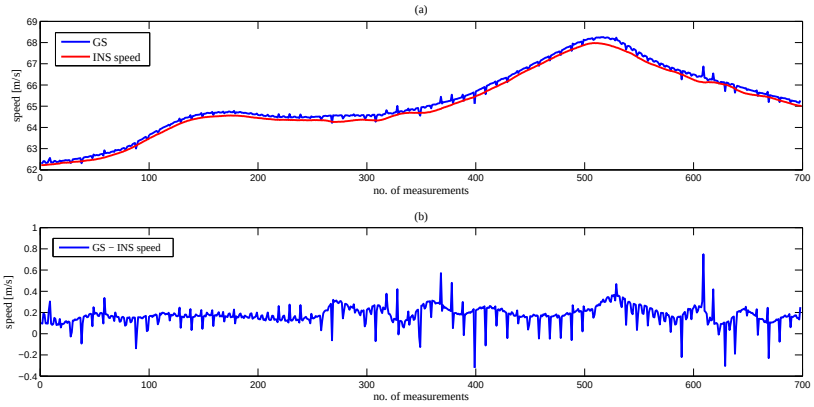


Figure 5.24: Comparison between GS and INS measured speed: (a)absolute values, (b)difference

scattered database has been transformed into a gridded database (ordered in an axis aligned grid) with the resolution equal to the radius used in processing the ALS measurements. Linear interpolation was used in this process. Figure 5.28 illustrates the terrain profile extracted from the scattered database versus the one extracted from the gridded database. As a result of the double interpolation, the level of details is slightly reduced.

Finally, in figure 5.29, the agreement between the ALS measurement profile and

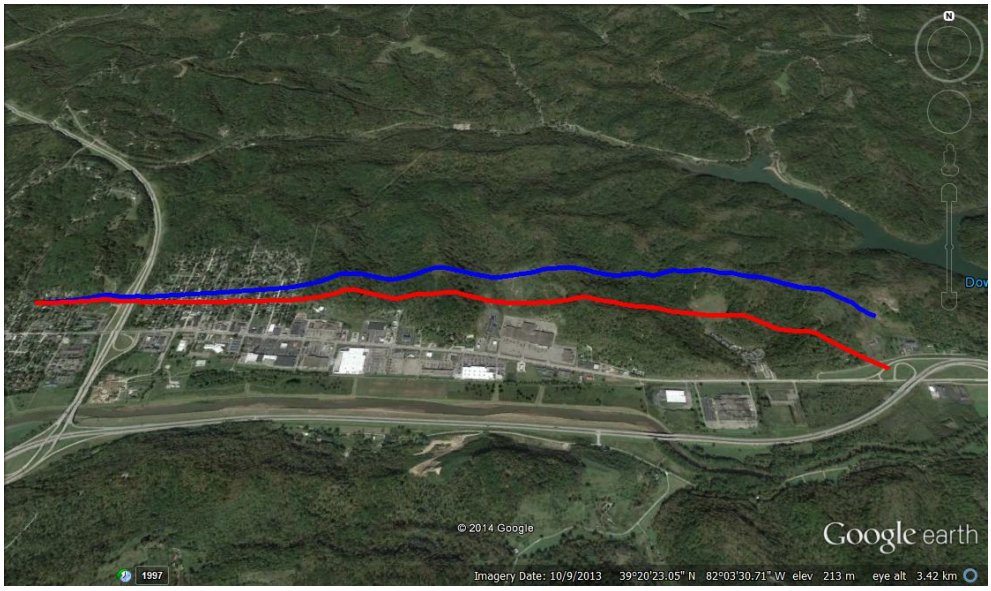


Figure 5.25: Comparison between GPS (in blue) and INS (in red) obtained trajectories

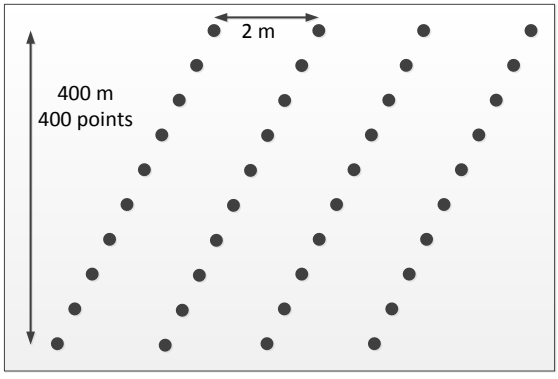


Figure 5.26: ALS scanning pattern and parameters

the database extracted profile is illustrated. Figure 5.30 illustrates the distribution of the terrain measurement errors per tile. A noticeable difference in the agreement can be observed over the course of the entire trajectory. In figure 5.21 it was shown that the first part of the trajectory is from a populated area, preponderantly with buildings

and only few trees, whereas the rest is from a sparsely populated area, with rolling hills and dense forests. It is important to notice that the ALS measured data is very different from the profile extracted in the second and third tile and this is attributed to vegetation. The vegetation can be penetrated to different depths. In dense vegetation conditions, the last return can come from tree tops, tree branches or tree trunks. The laser pulse can be reflected from any of them and this contributes to the high noise seen on the profile. Moreover, the two sets of measurements haven't been performed at the same time and vegetation is season dependable. It is anticipated that this phenomena will negatively impact the correlation algorithm.

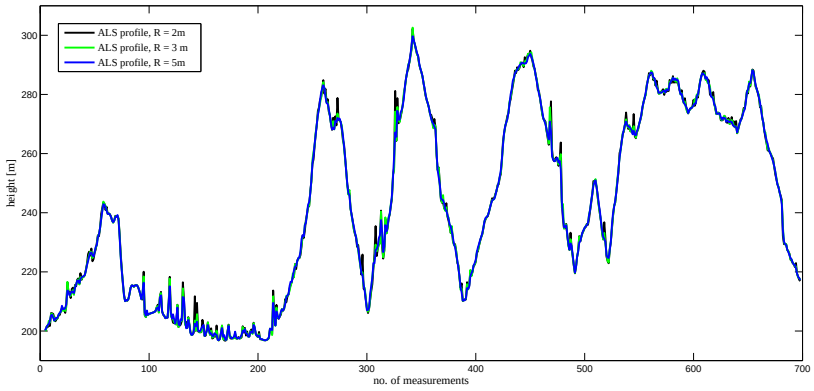
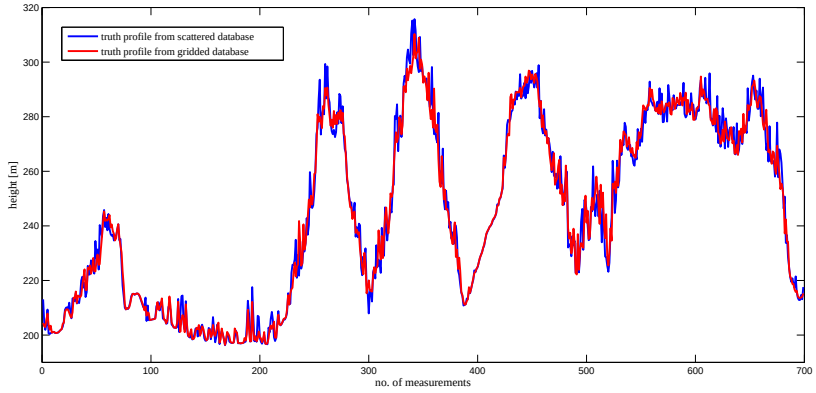


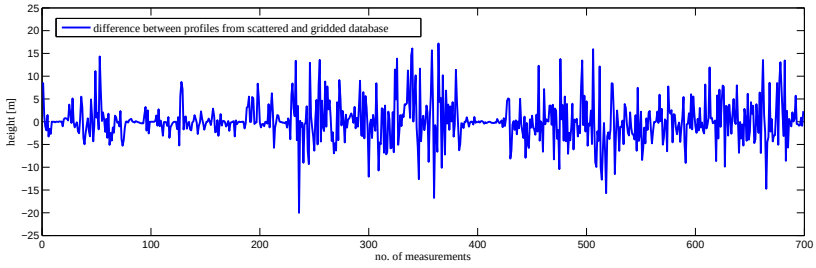
Figure 5.27: ALS measurement profiles, for different R

5.3.4 Results

The performance of the TRN tracking algorithm using the ALS flight test data was evaluated following the same setup as with the radalt data. No acquisition phase was implemented and it was considered that the starting position together with the starting arrays of speed and heading are accurately known. In the simulation, this data is provided from the truth references. The INS speed data, INS heading data and the ALS measured profile are provided as input to the algorithm. A radius of 5 m was used in the computation of the measured profile. The DSM tiles were concatenated and from the scattered data a gridded database was formed using a 5 m resolution. In accordance with the theory from chapter 3, the spacing in the speed/heading tracking loop was continuously adapted to the properties of the speed/heading correlation function. Table 5.5 provides the values assigned to other



(a)



(b)

Figure 5.28: Comparison of ALS database extracted truth profiles (a)absolute value, (b)difference

parameters of the algorithm. The TRN position estimates are shown in figure 5.31. Figure 5.32 illustrates the position error with respect to the GPS truth trajectory. The error was computed as the geodesic arclength between the truth geographical coordinates and the corresponding estimated coordinates. The speed and heading estimates are shown in figure 5.33 and 5.34.

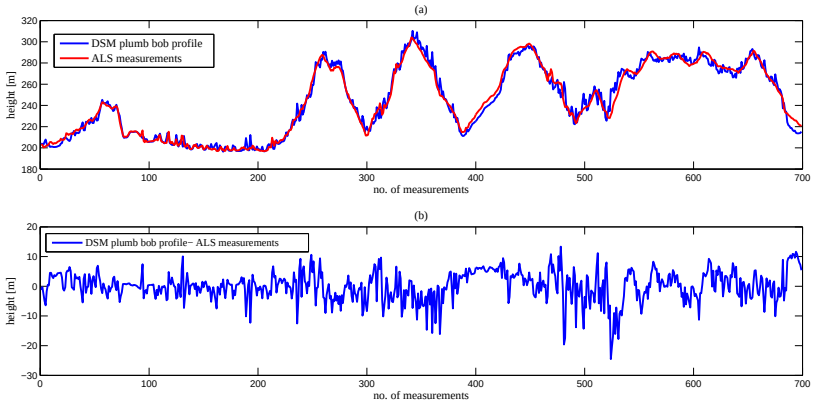


Figure 5.29: Comparison between DSM extracted profile and the ALS measurements: (a)absolute values, (b)difference

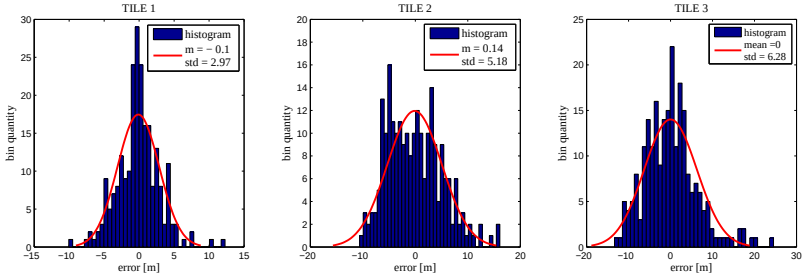


Figure 5.30: Statistics of difference between DSM extracted profile and the ALS measurements

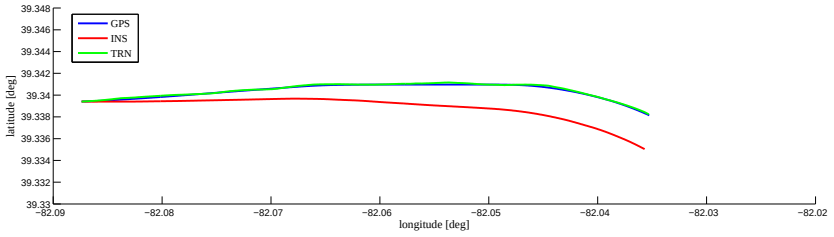


Figure 5.31: TRN position estimates using ALS data

The ALS TRN signal reproduces the bare-Earth shape on top of which real world spatial data (such as trees, buildings and other objects) is overlaid. The latter is a high frequency component, meanwhile the Earth surface has a low frequency of change (compared to the real spatial data). The level of agreement between the processed ALS sensor measurements and the DSM synthetized profiles varies, as shown in figure 5.29 and 5.30. The mismatches occur preponderantly on the level of the high frequency components. The errors are attributed to three sources: firstly, any un-calibrated orientation errors between the ALS and INS lead to a position bias which will propagate through the rest of the geo-referencing process. Secondly, a portion of the error is attributed to the presence of thick vegetation at the flight test location and finally, averaging the ALS measurements might lead to a decrease of the high fidelity reproduction of the terrain features. The errors in the height dimension affect the correlation. What we see is that tracking is based on the high frequency terrain signal rather than the Earth shape, as this is the dominant component in the over-all

Table 5.5: TRN Tracking parameters

Parameter		**Value**	**Comments**
N(transect length)		50 samples	On an average the TRN signal has 325 m and contains about 65 database grid points
1st speed tracking loop	spacing	adaptive	Multiple of the SCF resolution.It can take one of the following values: [0.5; 1; 1.5; 2; 2.5] m/s
	SCF resolution	0.5[m/s] 10 points	The minimum spacing (0.5 m/s) is equal to a shift in the test profile of database resolution points The maximum spacing (2.5 m/s) is equal to a shift in the test profile of 2 and database resolution points.
	threshold	0.002	Fixed, break at 101th iteration
2nd speed tracking loop	spacing	0.1 [m/s]	Fixed
	threshold	0.001	Fixed, break at 101th iteration
1st heading tracking loop	spacing	adaptive	Multiple of the SCF resolution. It can take one of the following values: [0.5; 1; 1.5; 2; 2.5] °
	HCF resolution	0.5[°] 10 points	The minimum spacing (0.5°) is, with the average speed, equal to a shift of about 3 database resolution points The maximum spacing (2.5°) is, with the average speed, equal to a shift of about 13 database resolution points.
	threshold	0.002	Fixed, break at 101th iteration
2nd heading tracking loop	spacing	0.1 °	Fixed
	threshold	0.001	Fixed, break at 101th iteration

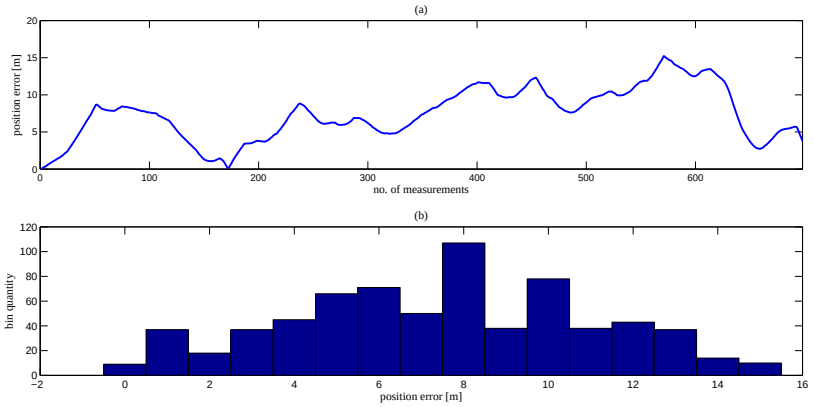


Figure 5.32: TRN position error: (a)absolute value, (b)histogram

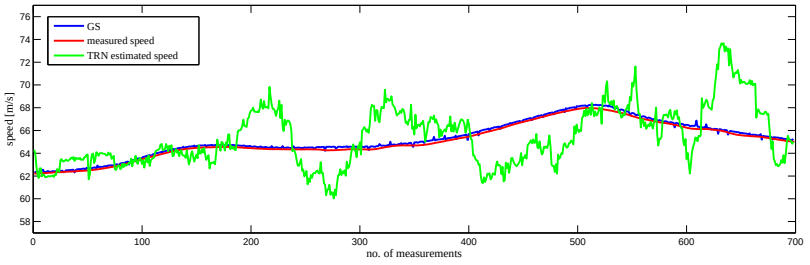


Figure 5.33: TRN speed estimation using ALS data

signal. The estimations are more accurate for the first part of the trajectory because of the relief: buildings are less susceptible to change and the ALS measurements have greater fidelity with respect to the terrain in the DSM. The trajectory corresponding to the second and third tile is located in more natural areas. Interestingly, note that the estimations of the TRN algorithm approximate better the truth values for the end part of the trajectory, despite that the route still covers a forest-like area. This demonstrates that the high frequency part of the terrain signal is rather unpredictable. Figure 5.35 emphasizes the impact of the high frequency modulation terrain signal on the correlation algorithm. In this case, the TRN algorithm was ran using a different spacing for the heading tracker. All parameters were preserved with the exception of the HCF resolution, which has been doubled to 1°. Therefore the spacing used

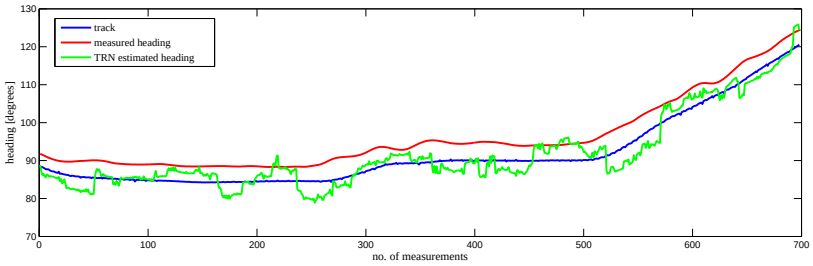


Figure 5.34: TRN heading estimation using ALS data

in the heading tracking loop is increased. The results obtained differ significantly, showing that even a slight heading change can give a completely new high frequency modulation component (although on the same low frequency terrain shape) which, in turn, will affect the correlation.

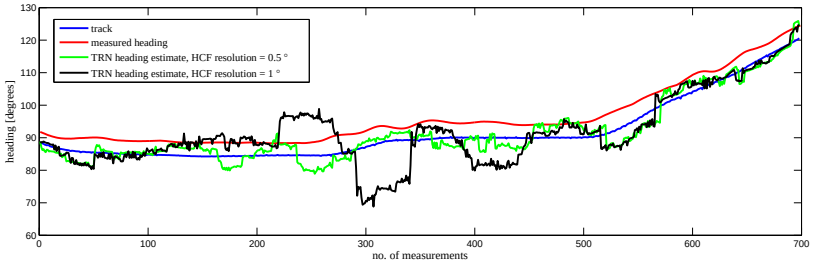


Figure 5.35: Comparison between TRN heading estimates when using different HCF resolution

5.4 Discussion and results

In this chapter the performance of the developed TRN tracking algorithm has been evaluated using two different sets of recorded sensor data: RADALT data and ALS data. The main goals of the chapter were:

- to validate the conclusions drawn in Chapter 4, where simulated sensor data was used to assess algorithm performance;
- to explore if and to what extent performance can be improved with better quality data.

The basic TRN signal needed in the developed algorithm is an array of terrain elevation samples, taken in the along-track direction. A minimum length is required (in terms of distance, not just of number of samples) to have an unique signal as the basis for the correlation process. In the context of the adaptable tracking loop, a (fixed) 25 samples length was first chosen. On an average this provided a signal length of 2 km. The resolution of the database was 30 m. The IRS used during the flight campaign provided an almost unbiased speed estimate, meanwhile the measured heading suffered from a rather consistent error which would cause for the position to drift up to 3 km by the end of the trajectory. As discussed in the previous chapter, the TRN tracking algorithm is able to detect and compensate for a constant or a low frequency change bias in speed or heading. This claim could not be fully validated due to the large errors in terrain measurements. The errors did not have a Gaussian distribution, but the std of the normal best fit was computed to 33 m, about two times larger than the expected std. Moreover the SNR (for a signal length of 25 samples) was low, dropping under the 5db threshold for most of the trajectory. Due to these issues, speed and heading TRN estimates showed fluctuations with a large magnitude. Besides a low accuracy, this also translated into the impossibility of detecting small biases. The errors in terrain measurements were attributed to the poor resolution of the RADALT measurements and, mostly, to the slant range errors. This statement was proven to be correct with the implementation of a roll correction algorithm. When roll correction was applied (compensating for slant range errors in turnings), the position accuracy was improved by 20% to 65%. To verify the processing gain, the performance of the algorithm was analysed using a double length TRN signal (50 samples). The main improvement was noticed in the speed and heading estimates, where the magnitude of fluctuations in the output was decreased. The position accuracy was (only) modestly improved. The explanation lies in the fact that, despite the increased signal length, the new SNR would sometimes still drop under the 5 db threshold.

Concluding, the analysis performed in section 5.2 showed that the poor resolution of the RADALT measurements together with the slant range errors limited the maximum achievable accuracy. The correlation is enough to keep the tracking loops in lock, estimate position and avoid inertial drifting, however small biases in speed and heading pass undetected. Thus, given the results, the question arising was: "How much could the performance of the algorithm be improved, if better data was available?"

The laser data had indeed better quality. Because data was not in an optimal format (one-dimensional signal in the along-track direction), further processing was required. This resulted in a 5m resolution for both the measurement profile and the terrain database. The length of the ALS TRN signal was chosen to encompass, on average, the same amount of sample points as used in the case of RADALT

measurements. The std deviation of the error distribution of the ALS terrain measurements was found to be 5 m on average, but varying from tile to tile (depending on the vegetation). The INS performance was similar to the one of the IRS used in the radar altimeter campaign: the INS provided speed was almost unbiased with respect to the GS, meanwhile the measured heading introduced a significant heading error with respect to the track. As expected, the TRN algorithm provided more accurate results when higher quality data was used. The following improvements were noted: 1) improvement in position accuracy (figure 5.32); 2) decrease in magnitude of fluctuations in the speed/heading estimate.

However, despite being far more accurate than RADALT measurements, laser data has one major issue: the terrain signal does not reproduce only the shape of the Earth surface itself, but has great fidelity with respect to the environment capturing all details like vegetation, buildings, power lines etc. Given the rate of change of these two components, the ALS measured TRN signal can be viewed as a composite signal: a low frequency component determined by the Earth surface on top of which a high frequency component (buildings, vegetation) is overlaid. As seen from the analysis in section 5.3, it is not desirable to achieve correlation based on the match between the high frequency features. Because it is less predictable, this part of the signal acts more as a disturbance, rather than as a matching feature. It is the Earth shape that provides the 'uniqueness' of the terrain signal and not the environment features. Their values can change as a function of the scanning angle or even of the season. Thus, the TRN concept works best if these high frequency components are removed or if their influence is suppressed to the point where the contribution of the slow varying Earth shape is by far dominant in the overall signal. The future design challenge lies in finding the appropriate filtering technique that will allow us to obtain, from the ALS signal, a 'new' TRN signal which, in terms of properties, is better than the RADALT provided one.

Chapter 6

Conclusions

6.1 Results

In this thesis the potential of improving TRN systems by applying digital signal processing concepts for signal acquisition and tracking has been explored.

In order to identify opportunities in the field of TRN and to understand how to utilize the existing knowledge from the related signal processing applications, an in-depth study on the past and existing TRN systems was performed. The study was presented in Chapter 2 and resulted in the following conclusions:

- The present DSP capabilities, the increase in real-time processing power, the existence of hi-resolution elevation maps and the developments in the area of terrain sensors are important enablers for TRN enhancements.

- Sequential TRN is faced with a rather conceptual limitation. Batch processing methods can make the more complete use of the latest technological developments. There is ample potential to benefit from the advances developed in related signal processing applications.

- Traditionally batch TRN techniques have been used for position aiding because of the constraints in storage capabilities and processing power. Modifications need to be made to allow a batch processing terrain system to handle an increased amount of data and to provide continuous navigation.

- Terrain features are a fundamental limitation for any TRN system. It is important to be aware of the 'quality' of information available in the surroundings and how it can be used to obtain the best possible navigation

solution. This implies detecting the operational context and adapting the algorithm according to it.

Based on these observations, a novel correlation algorithm for terrain navigation was developed. In the proposed approach, DSP techniques that have been originally developed for GPS code acquisition and tracking were adapted to TRN. The basic rationale for the proposed algorithm is to use terrain correlation to "acquire and track" the speed and heading of the host vehicle, while the position advances are calculated using these estimates together with the previously determined position. Whereas the method of location estimation derives from conventional batch TRN, the novelty of the approach consists in the implementation of a tracking scheme based on the DLL concept. Furthermore, to optimally benefit from the available information, the proposed architecture adapts the tracking of the terrain signal to the quality of the signal content. Throughout the development, several achievements can be highlighted. Among them:

- *Extension of the DLL tracking concept from deterministic sequences with pre-defined properties to random sequences*

  The deterministic sequences used as C/A codes in GPS are characterized by specific correlation properties and the implementation of the DLL is based on these properties. In this thesis, it has been proved that this type of feedback loop can be used also for any random sequences, as long as a reference for the code is known and a suitable measure for the correlation can be deduced. Mitigation strategies were proposed and applied successfully to solve for the issues caused by the lack of identical correlation properties between the C/A codes and random codes. (Chapter 3)

- *Identification of terrain signal specific properties for TRN signal characterization*

  Similar to RF signals, properties like bandwidth and energy of the signal were defined for a terrain signal. (Section 3.4.2)

- *A method to analyse the information content in the terrain signal and an algorithm that adapts its operation to the quality of the signal*

  For optimal performance, the configuration of the tracking loops must be made in accordance to the characteristics of the signal. Based on the analysis of the correlation function, the tracking parameters are set. The analysis is performed using the reference signal (extracted from the database) and not the measured signal. The analysis is performed continuously because the terrain signal varies with time. (Section 3.5.3)

- *A method to predict the performance of the algorithm based on the information content in the terrain signal and on the expected performance of the supportive sensors*

  The impact of speed and heading errors on the performance of the algorithm has been analysed. The total introduced error was modelled as a combined effect of measurement noise, sensor specific bias and data bias. A method to estimate and compensate the data bias introduced by flow-fields was implemented. It was shown that sensitivity to speed and heading errors is dependent on terrain features. The degree of sensitivity is assessed by analysing the terrain autocorrelation function. By combining this information with the expected error characteristic of the navigation sensors, the performance of the algorithm is predicted. (Section 4.2)

- *A method to predict and enhance the performance of the algorithm based on the information content in the terrain signal and on the expected performance of the primary sensor*

  The impact of altimeter measurement errors on the performance of the algorithm has been analysed. Two types of errors were identified: altitude dependent noise and slant range errors. Similar to RF signals, a signal to noise measure was introduced as the ratio between the variance of the TRN signal and the variance of the measurement noise. It was shown that sensitivity to terrain measurement errors is dependent on the SNR. Based on the expected performance of the radar altimeter and on the properties of the terrain, the expected SNR is predicted. If not sufficient, SNR is improved through processing gain. (Section 4.3)

- *Implementation of the algorithm in Matlab*

  The proposed TRN algorithm, with all features described in the thesis, has been implemented in Matlab environment. Additionally, a graphical user interface(GUI) has been designed to visualize a cycle estimation of the tracking loop: the user interactively sets the parameters of travel and of algorithm and the interface graphically illustrates how the estimation process is performed.

  The algorithm developed has been successfully verified based on experimental data with terrain signals coming from a radar altimeter and a laser altimeter (Chapter 5). Implementations of the algorithm with both heading and speed sensors as well as with the heading sensor only have been compared in the case of terrain signal coming from the radar altimeter sensor. Impact of the sensor data accuracies on the algorithm performance has been shown.

Compared to traditional batch processing TRN methods, the benefits of the proposed technique are:

- *Continuous navigation*

  TRN batch techniques have mostly been used for position fixes, rather than continuous navigation. This was partly because of the limited storage capabilities, but also because traditional correlation methods are time-wise inefficient to be performed continuously. While the traditional methods correlate the incoming signal with a large number of different profiles and base their decision on the correlation matrix, the presented method uses only two profiles for correlation. This considerably reduces the running time.

- *Adaptivity*

  'Terrain suitability' is a generally acknowledged issue for terrain navigation. The meaning is different, depending on the used techniques. For SITAN and other sequential methods it translates to clearly defined terrain gradients. For TERCOM and correlation based methods a rough, not self-similar topography yields the best results and a good performance is assured with a thorough pre-planning process. The proposed algorithm has a different approach: the information content in the signal is analysed and the operation is adapted correspondingly to this analysis. Due to the adaptive configuration, the approach has an increased versatility. Additionally, the prediction capability of the TRN algorithm can be used to provide navigation guidance.

- *Robustness*

  Robustness is an important asset to a navigation system, because (as the saying goes) although a method might work excellent it is useless if you do not know when this is the case. The integrity prediction feature contributes to the increased robustness of the presented algorithm.

6.2 Future work

Further refinements of the developed algorithm are certainly possible. Listed below are a few recommendations that would further improve the proposed TRN algorithm:

- The TRN tracking algorithm is said to be adaptive, meaning that the tracking loops are continuously configured for optimal performance based on the properties of the terrain and the expected performance of the primary and supportive sensors. Such an evaluation could also provide the basis for the guidance needed for choosing routes that assure the 'best performance' of the tracking algorithm. Depending on the context, 'best performance' will have different significations such as: higher accuracy (i.e. earlier bias detection), maximum tracking bandwidth (i.e. the ability to deal with larger errors), faster

convergence etc. Clearly, if information can be conveyed that indicates to the operator whether a certain manoeuvre can provide enhanced accuracy or better options to prevent loss of lock, this would be a valuable addition.

- In section 4.3.3, the impact of slant range errors on the performance of the TRN algorithm was analysed. Although it was shown that the impact is similar to the one of measurement noise errors, a quantifiable measure (analogous to the terrain SNR) could not be identified. In a next step, the relation between terrain characteristics and the likelihood of slant range errors should be studied more in depth. Alternatively, a compensation method for these errors may be applied. The 'spot' algorithm could be a good starting point.

- In section 4.3.4 it was observed that database inaccuracy causes discrepancies between the terrain sensor measurements and the database extracted elevations. Although not stated directly, it is implied that a poor accuracy of the database will negatively impact the algorithm's performance. However, an analysis on the contribution of the database accuracy to the system's achieved accuracy should be performed.

- In section 4.3.5 it was discussed how the presence of errors in the terrain measurements make it more difficult to estimate the errors in speed and heading, in particular limiting the values of the minimum detectable errors. A quantified relationship between the amount of uncertainty in the elevation errors and the amount of detectable errors in speed/heading has not yet been found. In the future this aspect should be further researched.

- In chapter 5, the experimental validation of the TRN tracking algorithm using laser terrain measurements was described. The only form of TRN signal supported by the current implementation of the tracking algorithm is an array of terrain measurements taken in the along-track direction, meanwhile an ALS uses a certain scan profile, providing measurements in both cross-track and along-track directions. Modifying the TRN tracking algorithm in order to fully benefit from the high 2-dimensional resolution of the ALS measurements would definitely be a challenging act worth researching. Also, the raw ALS TRN signal reproduces the bare-Earth shape(low-frequency component) on top of which real world spatial data(high-frequency component), such as trees, buildings and other objects, is overlaid. It was seen that the latter component of the signal acts more as a disturbance, rather than as a matching feature. The high frequency component should be removed or its influence suppressed. A future design challenge is to identify the appropriate filter technology to tackle this issue.

- The work presented in this thesis has focused mainly on aerial applications. This

algorithm is suitable for being used for underwater navigation too. However, this statement should be validated using experimental data.

The research presented in this thesis is part of a cooperative research program between Delft University of Technology and the Netherlands Defence Academy

Appendix A

Primary and supportive sensor data

The proposed TRN algorithm requires primary(terrain elevation measurements) and supportive(with a minimum of speed and heading) information. When designing a system that needs to combine information coming from different types of sensors, it is important to account for their characteristics. This section offers a general introduction on navigation sensors, with particular emphasis on the ones used in avionics. Spanning from technological characteristics to performance aspects, the information presented here serves as a background for Chapter 4, in which the implications of sensor integration within the TRN algorithm are discussed.

There are many criteria that can be used to classify the various types of navigation sensors. Depending on the sensing technologies, they can be divided into four main categories: air data, magnetic, inertial and radar sensors. Each technology will be briefly described, with a focus on the sensors that are of interest in our research.

A.1 Air data instruments

Almost from the beginning of powered flight, height and speed information has been provided by air data instruments. They work by measuring static pressure, pitot pressure and air temperature and then deriving a string of parameters, like: barometric altitude, indicated airspeed(IAS), calibrated airspeed(CAS), TAS, vertical speed, mach number, total air temperature. Simplest systems would provide ALT and IAS as a minimum, but modern aircrafts require the entire set. The advent of digital computing and data buses, starting with the late 70s, meant air data

computers(ADCs) could be introduced, to solve for some of the inherent issues of these instruments, such as: conversions, computation of parameters, correction(reduction) of pressure errors, (cold) temperature errors and compressibility effects. Later in the 80s, air data modules(ADM) were introduced to take over some of the ADC's tasks, with the benefit that they were physically placed in the vicinity of the pitot-sensing points. In [72] it is mentioned that virtually all transport aircrafts designed within the last 25 years have adopted the ADM implementation. For example, the Airbus system is rather complex and robust, containing three pitot, six static sensors and seven ADMs to distribute the air-data to the display, flight control and navigation systems [72]. However, it is important to underline that, despite the improvements made to this type of instruments over the years, they essentially work by measuring air pressure and the measurements will be incorrect if the airflow is turbulent, if there are crosswinds or the aircraft is side-slipping. Change in atmospheric pressure will also lead to some inevitable delays in readings. Countermeasures are taken to minimize the effects of such situations, but they cannot be completely solved. As "air data is generally regarded as accurate, but in the longer term rather than the short term" [72], integrating their measurements to the TRN tracking module raises the question in which ways these delays will impact the TRN algorithm. Another issue to consider when working with pressure instruments, common for all mechanical devices, is a (small) error of measurement due to the manufacturing tolerances. Besides sensing the measured subject, they can also be affected by temperature changes, vibration, shock, humidity etc. It was found from [73] that a typical value for airspeed indicator tolerance is about $\pm$3knots and for an ALT between 60-80 ft. It may not seem as high values from a navigational perspective, but when the task is to replicate a terrain profile the introduced error might be significant, as it is a function of both speed and flight altitude. Irrespective of any errors, the IAS (as well as all derived parameters: CAS, TAS) provides an indication of the aircraft's speed relative to the surrounding air and, as such, is a poor indicator of the speed with respect to the ground (GS). In a perfect still weather the two (GS and TAS) may be equal, but generally en-route winds will cause a difference between GS and TAS. Moreover, a change in the actual speed of the aircraft or a change of the wind current will both yield a change in the measurements of the air-data instrument. The TRN algorithm produces a navigation solution that locates the aircraft relative to the terrain and therefore it requires the aircraft's speed relative to the ground. A flow-based sensor will typically provide biased information of the ground speed. How much bias caused by the air-flows could be generally expected? How much bias could the TRN algorithm accept in order to keep performance at high standards? Is there a tolerance limit for the bias, with respect to the algorithm's performance? It is important to understand how such an error will influence the functionality of the tracking module and how the integration with the sensor should be handled in order to detect and correct the bias.

A.2 Magnetic instruments

The magnetic compass is one of oldest and simplest way of providing a reference for direction. The requirements for flights under VFR and IFR state that it is mandatory for all civil aircrafts to carry such an instrument on board. Although normally it is used as a standby compass, in general aviation aircraft of limited range and performance it is likely to be the only heading reference available [73]. Main type of errors affecting the magnetic instruments are: variation, deviation and dip errors. Variation is caused by the difference between true and magnetic direction. The Earth's magnetic variation has been accurately measured and charted (although long term changes may occur) and this allows for the variation error to be corrected within the system's computer. Deviation is due to the aircraft's own magnetic field and it can be minimized by the pilot through a maintenance task known as "swinging the compass". In maritime navigation, deviation can cause important errors and, especially on large ships, a different instrument is preferred to measure heading (e.g. gyrocompasses, described in a later paragraph). Dip errors appear because the Earth's magnetic field is inclined at a certain angle to the horizontal and the measured heading will consist of both a horizontal and a vertical component. The horizontal component represents the correct reading, meanwhile the vertical one will introduce errors, which can become significant at large dip angles. Another two noticeable errors caused by the dip are severe displacements of the compass reading during turns and accelerations/decelerations. Main types of magnetic compasses used in air navigation are: the vertical card compass, the flux gate system and the remote indicating compass. The fluxgate compass is also widely used on small vessels. On aircrafts, a magnetic compass is generally used for alignment or monitoring of other heading sensors. Otherwise, it is important to integrate it with an accurate vertical reference. In [74] is said that "modern magnetic sensors with computer compensation together with an accurate vertical reference enables the heading errors to be constrained generally to less than 0.7° for latitudes up to about 60°". If the only source of providing heading to the TRN algorithm is a magnetic sensor, the tracking module should be adapted to the particularities of the instrument. Some questions regarding such an implementation would be: what is the effect of the large heading errors during turns or acceleration on the TRN system's integrity? How much will that decrease the quality of the correlation process?

A.3 Inertial instruments

Gyroscopes and accelerometers form the inertial sensor category. As the name suggests, they detect motion by exploiting the property of inertia, namely the resistance to changes in linear motion in case of the accelerometer and to changes in

momentum to sense angular motion in case of the gyro [74]. Accelerometers measure acceleration on a particular axis and, by integration, the velocity (and position) of the vehicle can be derived. Note that the velocity information coming from an accelerometer is a measure of the GS (unlike measurements from air data sensors) and therefore would be preferred as input for the TRN tracking module. Gyros provide an attitude reference - typically aircraft pitch position, roll position and yaw position(heading) - or information relating to the aircraft body rates - pitch rate, roll rate and yaw rate. The principal flight instruments that use the properties of the gyroscope are: directional gyro(DG) or heading indicator(HI), the artificial horizon or attitude indicator and the turn and bank indicator. All gyroscopic instruments are of great importance in piloting, but within the frame of the present work, the interest is mainly in the HI. As pointed out in a previous paragraph, the magnetic compass has many inherent errors and therefore, in most aircrafts, is supplemented with a HI device. Main error for this instrument is the gyro drift, i.e. any deviation of the horizontally spin axis from its point of reference. Real drift is caused by friction or wear and the actual error will largely depend upon the condition of the instrument. The drift may be excessive if the bearings are worn, improperly lubricated or dirty, but in a properly maintained DG is said to be insignificant [25]. The Earth constantly rotates at about 15°/h while the gyro is maintaining alignment with a fixed point in space, thus causing another drift (the apparent drift) in the displayed heading of as much as 15° per every hour of operation. Another sort of apparent drift exists in the form of transport wander, where aircraft movement will essentially add or subtract to the effect of the Earth's rotation upon a gyroscope [73]. It is possible to predict the drift and a compensation device exists, known as latitude nut. However, it must be set on the ground and provides compensation only at the latitude for which it was set. A more common procedure is to compare the HI with the magnetic compass about every 10 - 15 minutes and reset the heading as necessary to find agreement. Providing heading information to the TRN module from a HI assures improved precision and reliability compared to the magnetic readings. However, it is difficult to make an exact assessment because the limitations of the HI vary with the particular design and make of the instrument. A simple heading indicator requires readjustments every 10 to 20 minutes. How would this short-term accuracy impact the TRN performance? Another design limitation example: for some HI, generally the ones found on light airplanes, there are limits for bank and pitch, approximately of 55°. That is, when either of these attitude limits is exceeded, the HI will tumble and will no longer provide a correct indication until reset. In this particular case (also considering the dip errors of the magnetic compass), the TRN algorithm will be left without heading information. Could the TRN module be designed in such way to be able to work without external heading info for a short period? This would be a rather exceptional situation, as many of the modern flight instruments used are designed

in such a manner that they will not tumble. As for the general performance of an HI device, this will vary with the used gyro technology. A brief overview of these manufacturing technologies will follow, after completing the list of main applications for inertial sensors in the avionics industry.

Besides stand-alone instruments, inertial sensors may also come in packages as inertial measurement units. An IMU combines multiple accelerometers and gyros to provide three dimensional measurements of body rates and acceleration. An IMU also includes a processor, a temperature sensor and a calibration-parameters store. This way the IMU is able to perform unit conversions, compensate for the known errors of the sensors and, in some cases, perform range checks to detect sensor failure [25]. IMUs are essential elements of the attitude heading reference system (AHRS) and the INS. The two share common technology and operating principles. Both AHRS and INS are obtained by pairing the IMU with a navigational processor, which integrates the IMU's output to provide attitude, position and/or velocity. The difference between them is largely determined by the grade of the included inertial sensors and consequently by the applications. AHRSs are normally used as secondary sources of heading information or are aided with air data velocity, Doppler radar data or magnetic heading data because they comprise lower grade IMUs. At the upper end of the performance spectrum we find INSs, accurately enough to be used even as sole means of navigation for significant portions of flight. The term inertial reference system (IRS) may also be encountered in aircraft navigation and there is no agreed differentiation between an INS and an IRS. For example, some sources say that the difference is that an INS can work as an independent navigator, meanwhile an IRS can only be used to feed data to a flight management system (FMS). Others relate it to performance, an IRS generally being less accurate, "with an accuracy of up to 4nm/hour error compared with 1 to 2 nm/hour for a typical INS" [74]. Essentially they work on the same principle, with the difference being only in the age of development and the physical engineering.

There is a great diversity of technologies applied to inertial navigation, as illustrated in Table A.1. It is beyond the scope of this book to explain how these devices function, the attempt here is to review only some of the inertial technologies, as found relevant for the proposed topic.

Table A.1: Technologies applied to inertial navigation sensors

Sensor	Gyroscope					Accelerometer	
Type	spinning mass		optical		vibratory	pendulous	vibratory
Phenomenon	angular momentum	gyroscopic precession	Sagnac effect		Coriolis effect	force feedback	open loop
Type of implementation	momentum wheel bearing technologies		ring laser gyro: mechanical ditter multi frequency	fiber optic gyro: RFOG IFOG	tuning fork HRG vibrating cylinder	simple pendulous EMA pulse integrating	MEMS

The spinning mass gyros were the first to be introduced. Due to their mechanical

complexity, power consumption and cost of ownership, mechanical gyros have started to be gradually replaced by the solid-state implementations. The dynamically tuned gyro(DTG) is one of the few spinning mass gyros remaining widely in use because of its fast reaction time, small size and relatively low cost. From an application perspective, the DTG can cover a wide performance spectrum: from 0.01°/hour to 30°/hour rate uncertainty [74]. Ring laser gyros(RLG) dominate the high performance application market, with characteristics as: 0.01°/hour or less bias uncertainty, insensitivity to acceleration, negligible warm up time and mean time between failures in excess of 60.000 hours. On the down side we find size, weight, high power consumption and price as limiting factors.The other class of optical gyros, the fiber-optic gyros (FOG), are much less demanding in their fabrication techniques and much more flexible in their design and packaging [75] and as a consequence are more price accessible. Currently FOGs are considered viable rate sensors for the medium performance range. With the advent of micro electromechanical technology in the 90s, quartz and silicon sensors began to be produced. Vibrating mass gyros exploit this technology. The performance of MEMS instruments is continually improving and currently they cover the tactical performance end of the application spectrum.

Most common accelerometers used today in navigation are pendulous, vibrating beam and the pendulous integrating gyro (PIGA) accelerometers. Sensors belonging to the first two categories are either mechanical or implemented using MEMS technology and they have lower performance and cost. Quartz resonators may also be found in lower grade tactical and consumer applications. PIGA is a mechanical instrument and is considered the highest performance accelerometer existent. Its applications include strategic missile guidance and high grade INSs. Generally, the range of performance for accelerometers spans of only about three orders of magnitude, in contrast with the one of gyros which is more than the double.

Up to this point, terminology like high, medium or low grade has been used when referring to the performance of the inertial sensors, without defining any clear boundaries between the three categories. This is because there is no universally agreed definition of these terms and it may depend on the author of the cited reference. A more precise classification could be found in [25], where IMUs and inertial sensors are divided into five groups according to the application: marine, navigation, intermediate, tactical and consumer. The marine-grade INS (used on ships and submarines) tops the performance scale with a navigation solution drift of less than 1,8 km per day. Navigation-grade INSs have specification requirements of a drift of 1,5 km in the first hour of operation and are normally used in military and commercial aircrafts, meanwhile intermediate IMUs are used in small aircrafts and helicopters. At the end of the performance spectrum we find the tactical and consumer grade inertial sensors which are sold as IMUs or stand-alone sensors, as opposed to the marine, navigation and intermediate grades, which tend to be sold as part of

an INS. Tactical sensors are used in guided weapons, UAVs and AHRSs. Finally, consumer-grade sensors have little application in navigation, their market being mostly the automotive industry. The parameters normally taken into consideration when categorizing or discussing performance levels of IMU/INS/IS are: bias, scale factor and random noise because these are the most critical error sources for both accelerometers and gyros. These errors will be briefly discussed below, but for more information [25], [74], [76] are great readings to start with.

Bias is the dominant term in the overall error of an inertial instrument. It comprises a static and a dynamic component. The static bias comprises a run-to run variation plus the residual fixed bias remaining after sensor calibration. It is deterministic by nature, but varies from run to run and can be addressed by calibration. The dynamic bias, also known as bias instability, changes during the course of a run and is typically modelled as random noise. As a conclusion, it is important to note that the bias evolves in time in an unpredictable manner and there is no such thing as a perfect bias calibration. Typical accelerometer and gyro biases for different grades of IMUs are shown in table A.2. The scale factor represents the linear approximation of the sensor's response over a given input range. In practice, the response might have some non-linear characteristics. Similar errors are cross-coupling and misalignment errors and normally they are treated together. The range of scale factor errors, as a function of the used sensor technology, can be read from figures A.1 and A.2. Random noise originates from a number of sources. For example: MEMS are affected by electrical noise, pendulous accelerometers and spinning mass gyros are subject to mechanical instabilities, in the RLG the residual lock-in effects manifest as noise, while vibratory gyros exhibit high-frequency resonances [25]. Noise problems may also arise when the IMU is rotating or is placed in a highly dynamic/vibration environment. The spectrum of noise for frequencies below 1Hz is approximately white. By integrating white random noise on the specific force/angular rate measurements the following parameters are obtained: velocity random walk/angle random walk. Some typical values for these parameters for different grades of IMU can be read in table A.2. Further sources of error for inertial instruments are the so called 'specific errors', because they are dependant of the sensor design.

Table A.2: Typical values for biases and random walks for different grades of IMUs

IMU grade	Accelerometer		Gyroscope	
	Bias [mg]	Random walk $[\mu/\sqrt{Hz}]$	Bias [°/h]	Random walk $[\mu/\sqrt{Hz}]$
marine	0.01	20 - 100	0.001	0.002 - 0.02
aviation	0.03 - 0.01		0.01	
intermediate	0.1 - 1		0.1	0.03 - 0.07
tactical	1 - 10	100 - 1000	1 - 100	0.07 - 0.1
consumer	> 10		> 100	> 1

Table A.2 illustrated typical numbers for bias and random noise encountered in instruments belonging to the different performance classes, but it is important to underline that those values are pertaining to the raw accelerations and rotation rates, not to the navigational parameters. Integrating acceleration with respect to time will yield the velocity and travelled distance. Therefore any errors present in the sensor's output are numerically integrated and will result into velocity errors, which in turn will be again integrated with time producing errors in position. A constant error in an accelerometer leads to a distance error proportional to the square of the elapsed time. An error in the gyro introduces errors in the orientation of the accelerometer input axes, leading to a distance error proportional to the cube of time. For instance, an accelerometer bias of 0.1 mg will cause a distance error of 0.49 mm after 1 second and of 1.76 m after 1 minute. A gyro with a drift of $0.1°/h$ will cause a distance error of 0.045 mm after 1 second and 9.8 m after 1 minute. The example considered rather large sensor errors, but it was used to emphasize the importance of inertial sensors accuracy, particularly the gyro accuracy. This is a key parameter for any inertial system, as it has the greatest impact on the overall navigation solution drift. Note that, in addition to bias, other sources of errors (as described above) will contribute to the accuracy of the inertial sensor. Some of the errors may be corrected, some may not. Usually, the IMU processor corrects the constant and temperature components. The so-called turn-on component (i.e. different each time the sensor is used, but constant within the run) can be calibrated. The in-run component (i.e. different each time the sensor is used and variable within the run) is random and normally unsolvable. Thus, when incorporating inertial sensor measurements in an application the first question that arises is what kind of errors are we dealing with? Do we really understand them? Can we deal with different errors in the same way? Or are they to be treated differently? Due to the complexity and diversity of existing manufacturing technologies, is hard to find a general accepted answer and eventually it comes down to the make and model of the specific IMU you are working with. Figure A.1 and A.2 illustrate the gyro/accelerometer bias and scale factor requirements, as well as the type of instrument most likely to be found in different applications. The figures have been adapted from a study published in 2011 [5]. In case of using an INS/IRS or an AHRS, the propagation of error to velocity and attitude does not depend solely on the inertial instrument error, but is also a function of the initialization error, the gravity model used in computations and the host-vehicle dynamics.

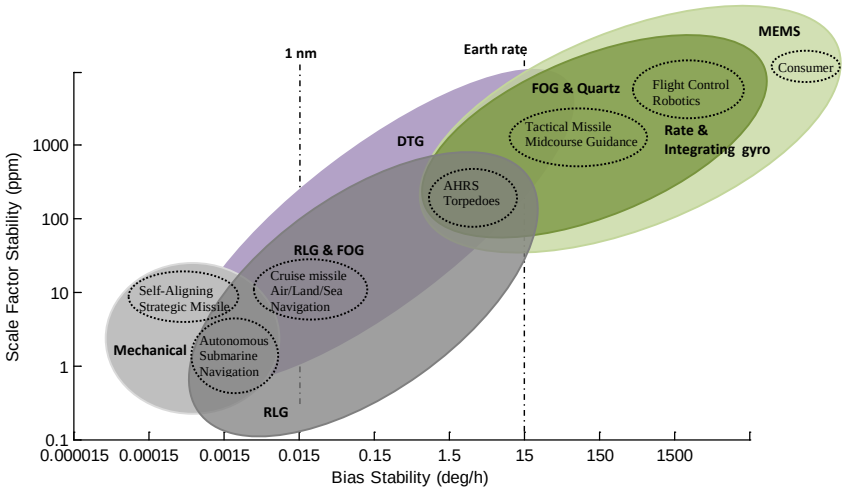


Figure A.1: Current gyro technology applications (illustration adapted from [5])

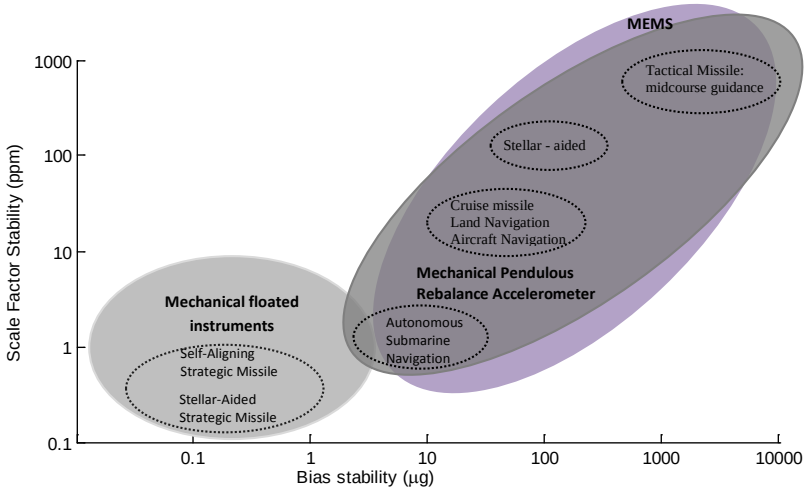


Figure A.2: Current accelerometer technology applications (illustration adapted from [5])

In conclusion, inertial sensors are key state sensors for both air and maritime navigation. They have the advantage of being accurate in the short-term, unlike the air data which needs settling time. On the other hand, they have the tendency of drifting with time (even a high quality INS with an accuracy of 1nm/h will have a position uncertainty of 3nm after 3 hours!). A wide variety of inertial sensors is available, with performance specifications varying by several orders of magnitude. The choice is often dictated by the application. Given all this diversity, it is reasonable to ask whether it is possible to design a TRN algorithm that could accommodate all level of performance in sensors. Can we deal with different performance sensors in the same way? Is it possible to obtain long time navigation with low performance sensors? Are there any minimum sensor performance requirements?

A.4 Radar sensors

Main radar sensors in use in avionics are: radar altimeter, Doppler-velocity radar, forward looking terrain radar and weather radar. Given the interest of the present research, only the altimeter and the Doppler velocity radar will be addressed. Although both instruments are based on the radar principle, the altimeter provides primary information meanwhile the Doppler radar falls (for the proposed algorithm) in the category of supportive sensors.

A.4.1 Radar altimeter

The role of the RADALT is to measure and display the vertical distance between the aircraft and the ground beneath it, thus the AGL. This is crucial information for the pilot during approach and landing and in fact CAT II and III precision approach procedures require the instalment of an altimeter. Alternatively, AGL data is needed by a large range of applications, such as: TRN function, FMS, ground proximity warning system, autopilot etc.

Traditional radalts operate in the C frequency band, with a central frequency at 4.3 GHz. They commonly use separate identical antennas for transmission and reception. There are three main modulation techniques used: FM-CW, pulse and spread spectrum. All three types of radalts use a tracking loop to smooth out the noise from successive measurements and filter out anomalous returns [25]. The operating range is between 0 and 2500ft or 5000ft, depending on the model. The manufacturer accuracy specifications are normally given as noise or noise as a function of height (with typical values of about 1 ft. or 3% of the indicated height, whichever is greater). However, the true performance of the instrument is largely dependent on the antenna's beamwidth, the flight altitude and the roughness of the terrain. Height measurement errors are larger when the aircraft is higher, as this leads to

a larger footprint. For example, a beamwidth of 17° (which is relatively narrow) will have an effective footprint diameter of almost 300m when the airplane flies at 1000m above the terrain and 450m at an altitude of 1500m. The terrain surrounding that below the aircraft will further impact the measurement error: larger numbers will be expected where there is more variation in the terrain captured within the footprint. The beamwidth may be reduced by using a larger antenna aperture or a higher frequency, but this remains an essential problem. In an algorithm based on the correlation between measured and stored data (such as the proposed TRN algorithm), the accuracy of the measurements is essential. What is the effect of the beamwidth size on the correlation algorithm? And what should be the system's requirements regarding this aspect, in order to avoid the decline in quality of the correlation result? Errors may further be caused by pitch and roll values. When the aircraft is not level, the peak of the transmitted beam will not reach the terrain directly beneath. Wider beams are more tolerant than narrow beam designs. However, when using radalt measurements, one must pay careful attention to this fact. Should the measurements be rejected once pitch and roll angles exceed a certain threshold? Or can this be compensated through processing? Lastly, it should be indicated that errors can also be expected due to reflections from parts of the aircraft structure or the receiver might pick up a multi-path component reflected from the ground. To a large extent, these errors can be solved.

As already discussed in chapter 2, other sensing methods can be used to provide primary information. Most promising results in terrain navigation have been noted with the use of interferometric SAR and laser technology.

For ships, submarines and AUVs, a sonar can be used to measure the distance from the vessel to the seabed.

A.4.2 Doppler Radar

A Doppler radar is a specialized tool which makes use of the Doppler effect to calculate the velocity of the host vehicle. The most encountered configuration is one of three or four beams, distributed to the front and rear of the aircraft. The frequency difference between the radiated and reflected energy enables for the GS to be derived. By measuring the lateral frequency difference, cross track velocity can also be calculated. Typical accuracy values found in the literature for velocity over land are of 0.06 m/s ±0.2%, with high performance designs specifications of about a factor or two better [25]. Doppler radar also needs an external attitude reference source (such as an AHRS or INS) and information on the pitch angle. Long-term position accuracy is about 1% of distance travelled with AHRS attitude and 0.15% with INS attitude [25]. For the forward and aft beam geometries (most common ones) the horizontal velocity error is of the order of 0.015% per degree of error in pitch angle. As with the altimeter,

certain types of terrain will not reflect enough energy to allow the Doppler radar to work properly. Such conditions occur while flying over snow-covered or glacial terrain, as well as over water. It should be noted that in this case the velocity is measured with respect to the water and not the Earth's surface. Older Doppler systems provided the user with a land/sea calibration switch, reducing the residual errors to 0.3% - 0.6%. Newer designs typically use a modified beam shape, reducing the velocity errors to within 0.2%, while high performance units measure the variation in scattering coefficient using additional or steerable beams [25].

Noise can also be a problem for this instrument, as the Doppler shift varies across the footprint of each beam while the scatters are distributed randomly. Noise standard deviation varies as the square root of the velocity [25]. This is why the Doppler radar is preferred to be used in lower speed applications, such as helicopters. They provide a good alternative to the more expensive INS. However, information was found that Doppler radars are no longer fitted in the new generation of military helicopters and strike fighters because of the radar energy emissions which increase the risk of detection [74].

For submarines, ships and AUVs the same principle can be applied to measure the velocity of the platform with respect to the seabed. The instrument is known as Doppler velocity log or Doppler sonar. The speed of water varies with temperature, depth and salinity and this is normally corrected through processing methods. Alternatively a correlation velocity log may be used. For both instruments the accuracy is given to be about 0.5% of distance travelled.

Appendix B

Simulation environment and MATLAB code

In this appendix details are given on the simulator architecture and structure of the code that has been used to develop and investigate the performance of the proposed TRN algorithm. The simulator has been developed in MATLAB environment.

B.1 Structure of the code

Figure 3.4 gives an overview of the logical building blocks that make up the TRN algorithm. The developed computer simulation followed this model. The actual data flow and the Matlab functions used are depicted in figure B.1. First, the terrain database is loaded. Next, a small section of terrain data and navigation data are read from file and passed to the **Acquisition** *m*-file. Navigation data comprises the speed, heading and time between consecutive measurements. As discussed in chapter 3, acquisition requests for these parameters to be constant.The **Acquisition** *m*-file estimates the speed and heading. The results are stored in the **AcqOutput** file. Next, the **AcqPosition** reads the **AcqOutput** file and calculates the position solution.

Once acquisition is completed, the tracking can begin. For tracking, a section of the input signal is read from the recorded and passed to the tracking function: **Tracking** *m*-file. The function is fed one sample of terrain data and navigation data (speed and heading of the vehicle). The tracking function calculates the signals (terrain signal, the speed signal and the heading signal), tracks the speed and heading, calculates the correlation functions, sets the tracking parameters(correlator spacing and correlator threshold), checks if the tracker is in lock, analyzes terrain. The main

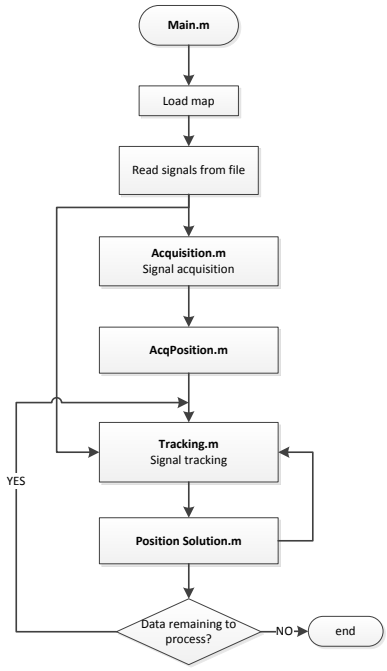


Figure B.1: TRN system flow diagram

tracking results (such as speed and heading) are stored in the **TrackOutput** file. Additional tracking results used for further analysis (such as correlation function, bandwidth, number of iterations) are stored in the **TrackResults** file.

After tracking is finished, the function **Position Solution** is launched to calculate the current estimation of position. The function stores the previously estimated position and together with the estimates of speed and heading computes the current position. The results are also stored in the **Position** file.

The computer simulation continues with tracking as long as there is data remaining to process.

B.2 Acquisition Function

The Acquisition function employs the serial search algorithm described in section 3.3.2. The purpose is to find a rough estimate of speed and heading. The flow

diagram for the actual code is shown in figure B.2.

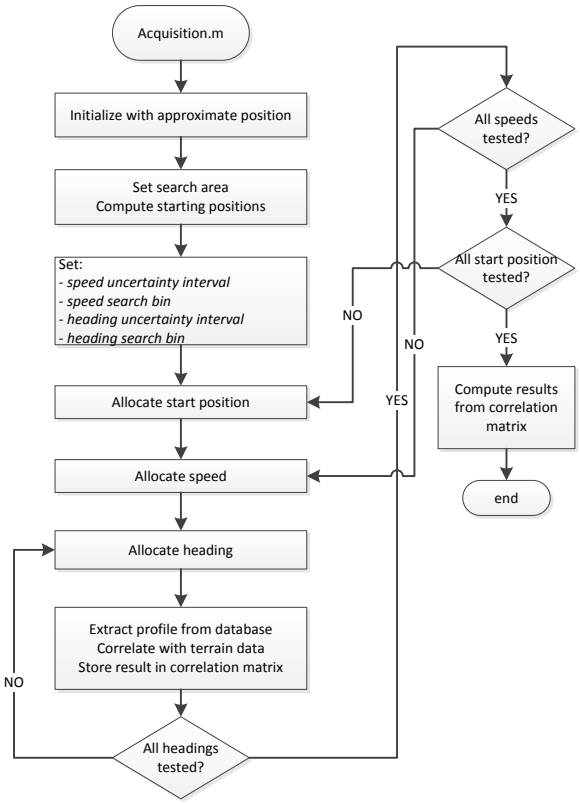


Figure B.2: Flow diagram of acquisition

The acquisition function looks for the speed signal in a step set by the *speed search bin width* (user defined), given the limits provided by the *speed uncertainity interval* (user defined). The acquisition function looks for the heading signal in a step set by the *heading search bin width* (user defined), given the limits provided by the *heading uncertainity interval* (user defined). For each speed, all headings are used and for each speed and heading combination a profile is extracted from the map and a correlated with the (measured) terrain signal. The acquisition correlation matrix is built row by row, according to the figure 3.6(a). A new correlation matrix is created for each starting position.The output is a multidimensional array structure containing search

results for all combinations.

After interpreting this result, the Acquisition m-file delivers the output: the estimated speed and heading.

B.3 Tracking Function

This function tracks the speed and heading of the vehicle. The function takes as input the following parameters: terrain signal, speed signal, heading signal, time signal (time elapsed between consecutive samples), start position. The tracking function tailors the input signals according to the size of the tracking signal (user defined). The TRN tracking flow of diagram for the actual code is shown in figure B.3.

User defined parameters for speed tracking are:

1. 1st speed tracking loop:
   (a) SCF resolution (calculated as described in section 4.2.3)
   (b) Threshold (user defined. Alternatively, the threshold may be assigned predefined values depending on the bandwidth)
2. 2nd speed tracking loop
   (a) Spacing (user defined)
   (b) Threshold (user defined)
3. 1st heading tracking loop:
   (a) HCF resolution (calculated as described in section 4.2.3)
   (b) Threshold (user defined. Alternatively, the threshold may be assigned predefined values depending on the bandwidth)
4. 2nd heading tracking loop
   (a) Spacing (user defined)
   (b) Threshold (user defined)

The SCF analysis and HCF analysis have a similar functionality: determining if the SCF and HCF have a V-shape and detecting the existing type of distortions. When a V-shape is not found, a flag is raised signaling that tracking is out of lock. The flow of diagram of the algorithm used for the V-shape calculation is provided in figure B.4.

The tracking function has other added functionality, which can be enabled by the user:

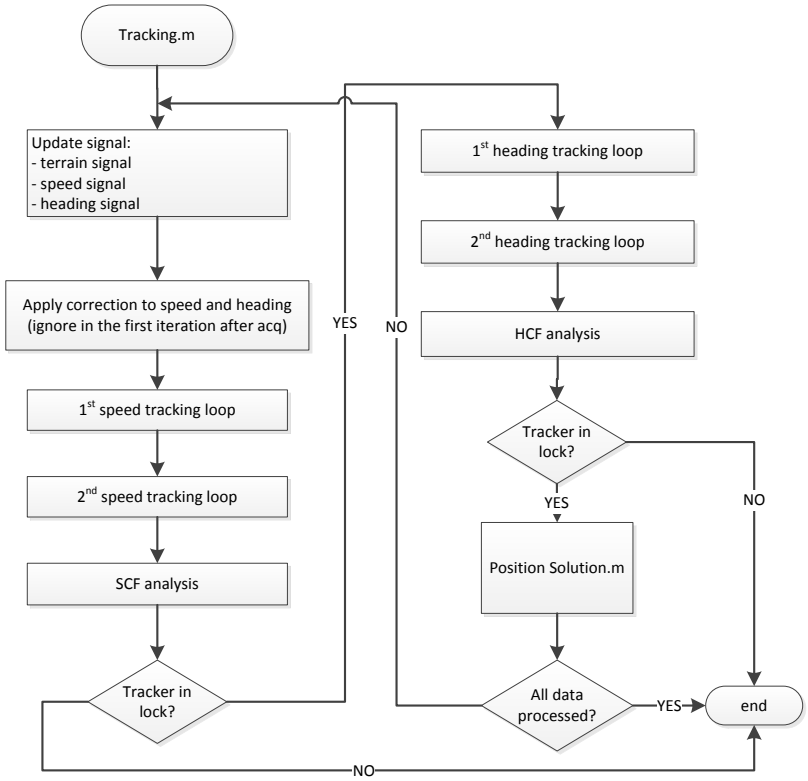


Figure B.3: Flow diagram of tracking

- SNR computation (as described in 4.3.2)
- ACF analysis (as described in 4.2.3)
- Roll compensation
- Graphical user interface to visualize a cycle estimation of the tracking loop

B.4 AcqPosition / Position Solution Function

This function computes the current position using: previously determined position, speed and heading.

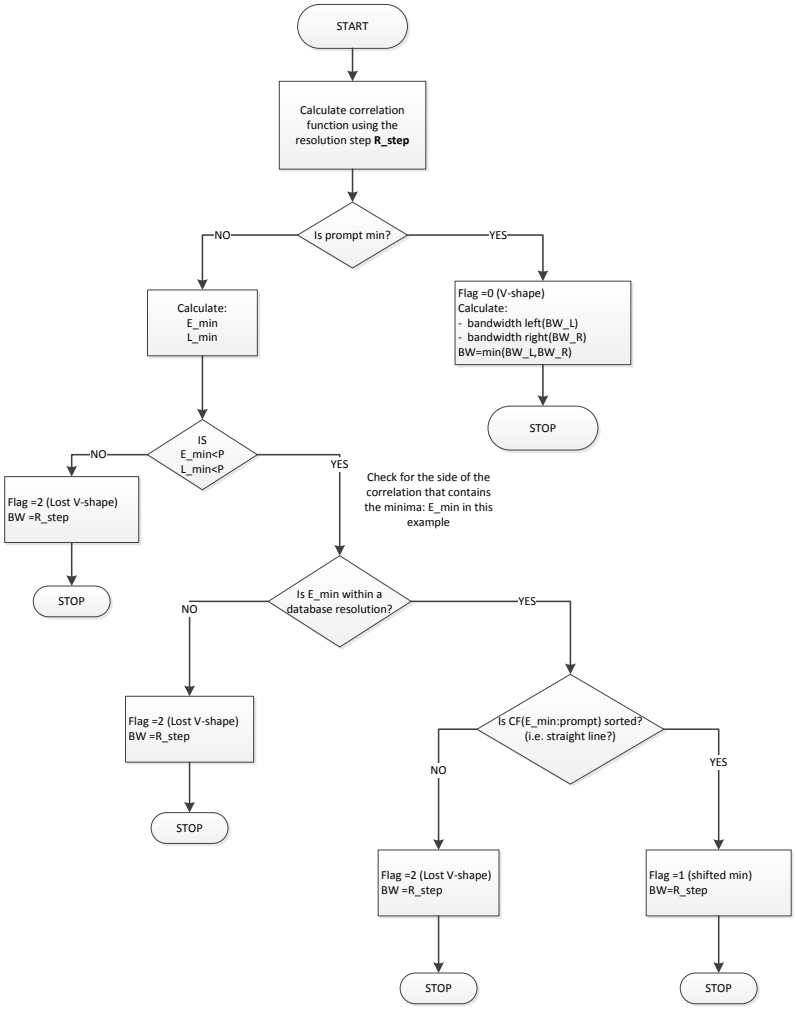


Figure B.4: Flow diagram of Bandwidth algorithm

List of Acronyms and Symbols

Acronyms

ACF	AutoCorrelation Function
ADC	Air Data Computer
ADM	Air Data Module
AFTI	Advanced Fighter Technology Integration
AGL	Above Ground level
AHRS	Attitude Heading Reference System
ALCM	Air Launched Cruise Missile
ALS	Airborne Laser Scanner
ALT	Barometric Altitude
ASD	Aeronautical System Division
ATRAN	Automatic Terrain Recognition and Navigation
BPSK	Binary Phase Shift Keying
C/A code	Coarse Acquisition code
CAROTE	Correlation and Recognition of Terrain Elevation
CAS	Calibrated Airspeed
CEP	Circular Error Probability
DEM	Digital Elevation Map
DG	Directional Gyro

DGPS Differential GPS

DLL Delay Locked Loop

DSMAC Digital Scene Matching Area Correlator

DTG Dynamically Tuned Gyro

EGM Earth Gravitational Model

EKF Extended Kalman Filter

EMA Electromagnetic accelerometer

FMS Flight Management System

FOG Fiber Optic Gyro

GAC Goodyear Automatic Corporation

GNSS Global Navigation Satellite System

GPS Global Positioning System

GS Ground Speed

HCF Heading Correlation Function

HI Heading Indicator

HRG Hemispherical Resonant Gyroscope

IAS Indicated Airspeed

IFOG Interferometric FOG

IFOG Interferometric Fiber Optic Gyro

IFR Instrument Flight Rules

IMU Inertial Measurement Unit

INS Inertial Navigation System

IRS Inertial Reference System

JHU/APL Applied Physics Laboratory of Johns Hopkins University

KF Kalman Filter

LEP	Linear Error Probability
LTV-E	Ling-Temco-Vought Electrosystems
MAD	Mean Absolute Difference
MCMC	Monte Carlo Markov Chain
MDAC	McDonnell-Douglas
MEMS	MicroElectroMechanical System
MMAE	Multi Mode Adaptive Estimation
MSD	Mean Squarred Difference
MSL	Mean Sea Level
NAVAIR	Naval Air Systems Command
P code	Precision code
PDF	Probability Distribution Function
PF	Particle Filter
PGCA	Prediction Ground Collision Avoidance
PIGA	Pendulous Integrating Gyro Accelerometer
PLL	Phase Lock Loop
PMF	Point Mass Filter
POS	Position and Orientation System
PPI	Plan Position Indicator
PRN	Pseudo Random Noise
RFOG	Resonant Fiber Optic Gyro
RLG	Ring Laser Gyro
SAR	Synthetic Aperture Radar
SCF	Speed Correlation Function
SITAN	Sandia Inertial Terrain Aided Navigation

SLAM Standoff Land Attack Missile

SLCM Submarine Launched Cruise Missile

SPARTAN StockPot Algorithm Robust Terrain Aided Navigation

SRTM Shuttle Radar Topography Mission

SVS Synthetic Vision System

TAINS Terrain Aided INS TERCOM

TAN Terrain Aided Navigation

TAS True AirSpeed

TAWS Terrain Awarness and Warning System

TERCOM TERrain COntour Matching

TERPROM Terrain Profile Matching

TF Terrain Following

TRN Terrain Referenced Navigation

VA Viterbi Algorithm

VFR Visual Flight Rules

WGS World Geodetic System

References

[1] G. Goebel, "The Wizard War: WW2 & the origins of radar," http://www.vectorsite.net/ttwiz.html, 2005, [public domain internet book].

[2] Radar, "U.S. Army publication," Issue No.3, 30th of June, 1944.

[3] R. Koch and D. Evans, "ATRAN terrain sensing guidance - the gran-daddy system," in *SPIE Vol. 238 Image Processing for Missile Guidance*, 1980, pp. 2–9.

[4] J. Golden, "Terrain contour matching: a cruise missile guidance aid," in *SPIE Image Processing for Missile Guidance*, vol. 238. SPIE, 1980, pp. 10–18.

[5] G. T. Schimdt, "INS/GPS technology trends," NATO Research and Technology Organization lecture series 232 Advances in Navigation Sensors and Integration Technology, 2003.

[6] "About EGNOS," http://egnos-portal.gsa.europa.eu/discover-egnos/about-egnos.

[7] M. Jones, "The civilian battle field: protecting gnss receivers from interference and jamming," *Inside GNSS*, vol. March/April, pp. 40–49, 2011.

[8] D. Hambling, "Gps chaos: How a $30 box can jam your life," http://www.newscientist.com/article/dn20202-gps-chaos-how-a-30-box-can-jam-your-life.html, 2011.

[9] G. Gibbons, "Inside GNSS news," http://www.insidegnss.com/node/3676, 2013.

[10] D. Shepard, J. Bhatti, and T. Humphreys, "Drone Hack: Spoofing Attack Demonstration on a Civilian Unmanned Aerial Vehicle," http://gpsworld.com/drone-hack/, 2012.

[11] S. Zaragoza, "Spoofing a superyacht at sea," http://www.utexas.edu/know/2013/07/30/spoofing-a-superyacht-at-sea/, 2013.

[12] J. Campbell and M. Uijt de Haag, "Assessment of radar altimeter performance when used for integrity monitoring in a synthetic vision system," in *20th AIAA/IEEE DASC Conference*, 2001, pp. 2C3/1 – 2C3/9.

[13] J. Campbell, M. Uijt de Haag, and F. Van Grass, "Light detection and ranging based terrain navigation," in *ION GPS/GNSS*, Portland, OR, 2003.

[14] J. Campbell, "Characteristics of a real-time digital terrain database integrity monitor for a synthetic vision system," Master thesis, Fritz J. and Dolores H. Russ College of Engineering and Technology Ohio University, 2001.

[15] ——, "Application of Airborne Laser Scanner - aerial navigation," Ph.D. dissertation, The Russ College of Engineering and Technology of Ohio University, 2006.

[16] A. Vadlamani, "Airborne Laser Scanner Aided Inertial for Terrain Referenced Navigation in unknown environments," Ph.D. dissertation, The Russ College of Engineering and Technology of Ohio University, 2010.

[17] D. Jensen, "PTAN's potential," http://www.aviationtoday.com/av/business-and-general-aviation/PTANs-Potential_979.html, 2004.

[18] F. Berger, "Aircraft Navigation System," U.S. Patent 2,847,855, 1958.

[19] G. Petrie, "Airborne topographic laser scanners," *Journal of GeoInformatics*, vol. January/February, pp. 34–44, 2011.

[20] A. Robins, "Recent developments in the TERPROM integrated navigation system," in *44th ION Annual Meeting*, Annapolis, MD, 1998, pp. 58–66.

[21] P. Hinrichs, "Advanced terrain correlation techniques," in *IEEE PLANS*, San-Diego, CA, 1989, pp. 89–96.

[22] Atlantic Inertial, "TERPROM," http://www.atlanticinertial.com/index.php?/products/detail/terprom, 2009.

[23] N. Bergman, "Recursive Bayesian estimation: navigation and tracking applications," Ph.D. dissertation, Linkoping University, 1999.

[24] I. Nygren, "Terrain navigation for underwater vehicles," Ph.D. dissertation, Royal Institute of Technology, Stockholm, 2005.

[25] P. Groves, *Principle of GNSS, Inertial and Multisensor Integrated Navigation Systems*, A. House, Ed., 2008.

[26] V. Ekutekin, "Navigation and control studies on cruise missiles," Ph.D. dissertation, The Graduate School of Natural and Applied Sciences of Middle East Technical University, 2007.

[27] H. Visser, *Array and phased array antenna basics*, Wiley, Ed., 2005.

[28] H. Mulberger and J. Bellitt, "Flight indicating instrument," U.S. Patent 2,526,682, 1950.

[29] F. Berger, "Optical Cross Correlator," U.S. Patent 2,787,188, 1958.

[30] F. Riedel, S. Hall, J. Barton, J. Christ, B. Funk, T. Milnes, and D. Stark, "Guidance and navigation in the global engagement department," *Johns Hopkins APL Technical Digest*, vol. 29, pp. 118–132, 2010.

[31] G. Siouris, *Missile guidance and control systems*, Springer-Verlag, Ed., New York, NY, 2004.

[32] W. Hallmark, "Fix-taking means and methods," U.S. Patent 3,328,795, 1967.

[33] D. Boozer and R. Fellerhoff, "Sitan test results in the AFTI/F-16 aircraft at edwards air force base," in *ION NTM*, Santa Barbara, CA, 1988, pp. 151–157.

[34] C. Baird and M. Abramson, "A comparison of several digital map-aided navigation techniques," in *IEEE PLANS*, 1984, pp. 286–293.

[35] L. Hostetler, "An analysis of a terrain aided inertial navigation system," Sandia Laboratories, Tech. Rep. SAND75-0299, 1975.

[36] L. Hostetler and R. Beckmann, "The SANDIA inertial terrain aided navigation system," Sandia Laboratories, Tech. Rep. SAND77-0521, 1977.

[37] J. Hostetler and R. Andreas, "Nonlinear kalman filtering techniques for terrain aided navigation," *IEEE Trans. Autom. Control*, vol. AC-28, no. 3, pp. 315–323, 1983.

[38] J. Hollowell, "Heli/SITAN: a TRN algorithm for helicopters," in *IEEE PLANS*, Las vegas, CA, 1990, pp. 616–625.

[39] Y. Pei, Z. Chen, and J. Hung, "BITAN II: an improved terrain aided navigation algorithm," in *IEEE IECON*, 1996, pp. 1675–1680.

[40] Z. Long, "An algorithm for terrain-aided inertial navigation based on nonlinear optimal filtering," *Journal Science China: Physics, Mechanics and Astronomy*, vol. 54, no. 6, pp. 1883–1888, Jun. 2011.

[41] R. Enns and D. Morell, "Terrain aided navigation using the viterbi algorithm," *Journal of Guidance, Control and Dynamiccs*, vol. 18, no. 6, pp. 1444–1449, nov-dec 1995.

[42] A. Runnalls, "A bayesian approach to terrain contour navigation," AGARD Guidance and Control Panel 40th Symposium, 1985, paper 43.

[43] A. Henley, "Terrain aided navigation - current status," in *IEEE PLANS*, 1990, pp. 608–615.

[44] A. Runnalls and R. Handley, "The gold standard navigator," in *Eurofusion '98*, 1998, pp. 77–82.

[45] T. Schon, F. Gustafsson, and P. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2279–2289, 2005.

[46] P. Nordlund and F. Gustafsson, "Particle filter for accurate and reliable terrain aided navigation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 4, pp. 1385–1399, 2009.

[47] N. Wilkinson, T. Brookes, A. Price, and M. Godfrey, "Latest development of the TERPROM digital terrain system," in *ION Joint Navigation Conference*, Orlando, FL, 2009.

[48] L. Chan and F. Snyder, "System for correlation and recognition of terrain elevation," U.S. Patent 4,584,646, 1986.

[49] L. Keearns, "Navigation system," U.S. Patent 4,495,580, 1985.

[50] ——, "Method of determining the position and velocity of a vehicle," U.S. Patent 4,520,445, 1985.

[51] C. Baird, "Map-aided navigation system employing TERCOM-SITAN. Signal processing," U.S. Patent 4,829,304, 1989.

[52] R. Goebel, D. Fogel, D. Toretta, P. Panagos, and P. Hefferen, "Terrain correlation system," U.S. Patent 6,218,980, 2001.

[53] K. Raymer and J. McGuffin, "Terrain referenced navigation - schuler cycle error reduction method and aparatus," U.S. Patent 5,450,345, 1995.

[54] N. Priestley, "Terrain referenced navigation," in *IEEE PLANS*, 1990, pp. 482–489.

[55] N. Bergman, "A bayesian approach to terrain-aided navigation," in *11th IFAC Symposium on System Identification*, Fukuoka, Japan, 1997, pp. 1531–1536.

[56] H. Lerche, "Navigation of aircraft by correlation," U.S. Patent 4,910,674, 1988.

[57] O. Dieffenbach, "Apals autonomous precision approach and landing system," in *51st ION Annual Meeting*, Colorado Springs, CO, 1995, pp. 247–251.

[58] J. Nicosia, K. Loss, and G. Taylor, "High resolution autonomous precision positioning system," U.S. Patent 6,865,477, 2005.

[59] M. Morici, "Aircraft position validation using radar and digital terrain elevation database," U.S. Patent 6,233,522, 2001.

[60] J. Hager and L. Almsted, "Methods and systems for enhancing accuracy of terrain aided navigation systems," U.S. Patent 7,409,293, 2008.

[61] W. Metzdorff, P. Lux, and M. Eibert, "Linear method of navigation," U.S. Patent 5,047,777, 1991.

[62] ——, "Method of navigation," U.S. Patent 5,087,916, 1992.

[63] A. Hicks, I. Scaysbrook, and J. Fountain, "Terrain navigation apparatus," U.S. Patent 6,389,354, 2002.

[64] K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. Jensen, *A software-defined GPS and Galileo receiver: a single-frequency approach*, Birkhauser, Ed., Boston, 2007.

[65] P. Misra and P. Enge, *Global System Positioning: signals, measurements and performance*, G. J. Press, Ed., Lincoln, MA, 2006.

[66] R. Gold, "Optimal binary sequences for spread spectrum multiplex," *IEEE Trans. Inf. Theory*, vol. 13, no. 4, pp. 619–621, 1967.

[67] A. Wilde, "The generalized delay locked loop," *Wireless Personal Communications*, vol. 8, pp. 113–130, 1998.

[68] P. Van Dierendonck, A.J.and Fenton and T. Ford, "Theory and performance of narrow correlator spacing in a gps receiver," *Journal of the Institute of Navigation*, vol. Vol.39, pp. 115 – 124, 1992.

[69] I. Irsigler and B. Eissfeller, "Comparison of multipath mitigation techniques with consideration of future signal structures," in *International Technical Meeting of the Satellite Division of The Institute of Navigation*, Portland OR, 2003, pp. 2584–2592.

[70] J. P. L. C. I. of Technology, "Shuttle radar topography mission: the mission to map the world," http://www2.jpl.nasa.gov/srtm/.

[71] O. gov., "Ohio geographically referenced information program," http://gis5.oit.ohio.gov/geodatadownload/.

[72] I. Moir and A. Seabridge, *Civil Avionics Systems*, P. E. Publishing, Ed., 2003.

[73] D. Harris, *Ground studies for pilots: Flight instruments and automatic flight control systems*, 6th ed., B. Science, Ed., Oxford, UK, 2004.

[74] R. Collinson, *Introduction to Avionics Systems*, Springer, Ed., 2006.

[75] J. Juang and R. Radharamanan, "Evaluation of ring laser and fiber optic gyroscope technology," web paper: www.asee.org/documents, 2009.

[76] J. Weston and D. Titterton, "Modern inertial navigation technology and its application," *Electronics and Communication Engineering Journal*, vol. April, pp. 49 – 64, 2000.

Summary

A GPS inspired Terrain Referenced Navigation algorithm

Terrain Referenced Navigation (TRN) refers to a form of localization in which measurements of distances to the terrain surface are matched with a digital elevation map allowing a vehicle to estimate its own position within the map. The main goal of this dissertation is to improve TRN performance through better signal processing. More specifically, the project aims to explore opportunities in the field of TRN by using digital signal processing techniques that were originally developed for the acquisition and tracking of GPS signals.

A typical TRN system uses speed, heading and time to establish the relative horizontal position between subsequent elevation measurements. Thus, any error in speed, heading or time will cause an error in the resulting relative position. If the speed or heading error contains a bias, this will cause a gradual reduction in the correlation. To prevent that a reduction in correlation causes the estimated position to drift away, the idea behind the research described in this thesis is the use of arrays of terrain elevation measurements with intentional (positive and negative) offsets in speed and heading in a tracking-loop configuration. It is well known that such a concept works well for optimized signals such as the ones used in GPS.

To further explore the viability of this idea for a signal defined by a series of terrain elevation measurements, an analysis of similarities and differences with the GPS signal is performed. In accordance to the GPS receiver approach, a novel correlation algorithm for TRN is proposed and implemented. The basic rationale for the algorithm is to use terrain correlation to "acquire and track" the speed and heading of the host vehicle, while the position advances are calculated using these estimates together with the previously determined position. The novelty of the approach consists in the implementation of a tracking scheme based on the DLL concept. To answer feasibility-related questions, the algorithm is first evaluated in a purely theoretical framework. Based on this analysis it is concluded that the concept seems feasible and promising, but additional considerations in the design are required

to compensate for the differences between the GPS and TRN signals. Enhancements are brought to the initial design resulting in the development of an adaptive tracking scheme, in which the tracking loops are configured based on an analysis of the terrain signal.

Next, an in-depth sensitivity analysis is carried out to understand how sensor measurement errors (in speed, heading and terrain height) impact the algorithm performance. The analysis is performed using exclusively simulated data. It is shown that sensitivity to speed and heading errors is dependent on terrain features and it is possible to assess the degree of sensitivity by analyzing the terrain signal. By combining this information with the expected error characteristic of the navigation sensors, the performance of the algorithm can be predicted. The sensitivity to terrain measurement errors depends on the ratio between the terrain signal strength and the measurement errors. It is shown that this ratio can be predicted up to a certain extent and a method to improve the ratio is proposed and discussed.

The developed capabilities are validated with recorded sensor data from flight tests. Two different types of recorded sensor data are used: radar and lidar based data-sets.

Samenvatting

A GPS inspired Terrain Referenced Navigation algoritm

Terrain Referenced Navigation (TRN) refereert aan een vorm van localisatie, waarbij afstandsmetingen tot terreinoppervlakten vergeleken worden met een digitale hoogtekaart. Hierdoor kan een voertuig zijn positie op de kaart schatten. Het belangrijkste doel van dit proefschrift is om de werking van TRN te verbeteren middels betere signaal verwerkingsmethodieken. Specifieker, het promotieproject had als doel de mogelijkheden op het gebied van TRN te onderzoeken door gebruik te maken van technieken uit de signaal verwerking, die in beginsel ontwikkeld zijn voor de acquisitie en tracking van GPS signalen.

Een typisch TRN systeem maakt gebruik van snelheid, koers en tijd om de relatieve horizontale positie te bepalen tussen achtereenvolgende hoogtemetingen. Daarom zal ook elke fout in snelheid, koers en/of tijd een fout veroorzaken in de resulterende relatieve positie. Als snelheid of koers een afwijking bevatten, dan zal dit leiden tot een geleidelijke reductie in correlatie. Om te voorkomen dat deze reductie tot een drift in de bepaalde positie zal leiden bekijkt het onderzoek in dit proefschrift het gebruik van arrays bestaande uit terrein hoogtemetingen, met opzettelijk ingebouwde afwijkingen (positief en negatief) in snelheid en koers, in een zogenaamde tracking-loop configuratie.

Om de levensvatbaarheid van bovenstaand idee verder te onderzoeken voor een signaal, dat gedefinieerd is door een reeks van terrein hoogtemetingen, is een analyse uitgevoerd van overeenkomsten en verschillen met een GPS signaal. Gerelateerd aan de aanpak binnen een GPS ontvanger is een nieuw correlatie algoritme voor TRN ontwikkeld en geïmplementeerd. Het basis beginsel van het beginsel is het gebruik van terrein correlatie voor de “tracking” en “aquisitie” van snelheid en koers van het voertuig, terwijl de verandering in positie bepaald worden door een combinatie van deze schattingen met de eerder bepaalde positie. Het nieuwe van deze aanpak bestaat uit de implementatie van het tracking schema op basis van het DLL concept. Om vragen op het gebied van uitvoerbaarheid te kunnen beantwoorden is de aanpak eerst

geëvalueerd binnen een puur theoretisch raamwerk. Op basis van deze analyse lijkt de aanpak niet alleen haalbaar maar ook veelbelovend, hoewel aanvullende overwegingen in het ontwerp noodzakelijk zijn om de verschillen tussen GPS en TRN signalen te compenseren. Aanpassingen zijn in het initiële ontwerp aangebracht, resulterend in de ontwikkeling van een adaptief tracking schema, waarbij de iteratieslagen zijn samengesteld op basis van een analyse van de terrein data.

Een uitvoerige gevoeligheidsanalyse is uitgevoerd om te begrijpen hoe fouten in de sensormetingen (snelheid, koers, hoogte) de werking van het algoritme beïnvloeden. Deze analyse is uitgevoerd met simulatiedata. Hierbij is aangetoond, dat de gevoeligheid voor fouten in snelheid en koers afhangen van de eigenschappen van het terrein en dat het mogelijk is grip te krijgen op de mate van gevoeligheid door de terrein data te analyseren. Door deze informatie met de verwachte foutkarakteristieken van de navigatiesensoren te combineren kan de werking van het algoritme voorspeld worden. De gevoeligheid t.a.v. fouten in de terrein hoogtemetingen hangt af van de ratio tussen de sterkte van het "terrein" signaal en de meetfouten. Er is aangetoond dat deze ratio tot op zekere hoogte ingeschat kan worden en een methode om deze ratio te verbeteren wordt geïntroduceerd en besproken.

De ontwikkelde mogelijkheden worden gevalideerd met gebruik van opgenomen sensordata van testvluchten. Twee verschillende soorten opgenomen data zijn gebruikt: op radar en lidar gebaseerde datasets.

Author's publications

Journal Papers

1. Oonincx, P., Vaman, D., "An early-late tracking algorithm for terrain referenced navigation," European Journal of Navigation, Vol. 8, 2010, pp. 1-10.

2. Vaman, D., Oonincx, P., "Design of a TRN Tracking Loop: a Study on GPS Multipath Mitigation Strategies," Cambridge Journal of Navigation, Volume 65, Issue 04, October 2012, pp 603-616.

Conference Papers

1. Vaman, D., Oonincx, P., "Exploring a GPS Inspired Acquisition and Tracking Concept for Terrain Referenced Navigation," Proceedings of the 2010 International Technical Meeting of The Institute of Navigation, San Diego, CA, January 2010, pp. 459-466.

2. Vaman, D., Oonincx, P., "Performance analysis of a GPS inspired terrain referenced algorithm," Proceedings of the 2010 European Navigation Conference, Braunschweig, DE., p. 1-11.

3. Vaman, D., Theunissen, E., Oonincx, P., "An Adaptive Early-late Tracker for a GPS Inspired TRN System," Proceedings of the 2011 International Technical Meeting of The Institute of Navigation, San Diego, CA, January 2011, pp. 801-810.

4. Vaman, D., Oonincx, P., "Design of a TRN tracking loop: a study on GPS multipath mitigation strategies," Proceedings of the 2011 European Navigation Conference, London, UK, p. 1-9.

5. Vaman, D., "TRN revisited: A systematic design of a GPS inspired adaptive tracking loop ", Digital Avionics Systems Conference (DASC),IEEE/AIAA

30th, Seattle, WA, October 2011, pp. 4B4-1 - 4B4-11.

6. Vaman, D., Oonincx, P., "Dynamics & Performance of a GPS Inspired TRN Tracking Loop," Proceedings of the 2012 International Technical Meeting of The Institute of Navigation, Newport Beach, CA, January 2012, pp. 262-268.

7. Vaman, D., Theunissen, E., Oonincx, P. "'GPS inspired TRN: tracking loops issues and solutions," Proceedings of the 2012 European Navigation Conference, Gdynia, Poland, p. 1-10.

8. Vaman, D., "Terrain referenced navigation: History, trends and the unused potential," Digital Avionics Systems Conference (DASC), IEEE/AIAA 31st, Williamsburg, VA, October 2012, pp. 1 - 28.

Other Publications

1. Oonincx, P., Vaman, D., "Terrain Referenced Navigation using a GPS approach", NL-ARMS, advances in military navigation technology (NL-ARMS 2010), pp. 55-72.

About the author

Daniela Vaman was born in Botoşani, Romania on August 7th 1984. She obtained her baccalaureate diploma from the Petru Rareş National College in Suceava, Romania in 2003 (mathematics-informatics). Between 2003 and 2008 she studied Telecommunications at the Faculty of Electronics, Telecommunications and Information Technology, at Gheorghe Asachi Technical University of Iaşi, Romania. During the last months of her study (February – July 2008) she was a guest student at KaHo Sint-Lieven Gent, where she worked on her Diploma Project "Indoor Localization using ZigBee networks". She received her engineering degree in September 2008, graduating in the top five percent of her year.

During her university studies, Daniela also completed two summer internships.The first internship was at PSE Siemens Brasov, June – September 2006, where she worked in the Network Security department, researching on methods of detection of network intrusion. From June – September 2007, she worked as an assistant programmer at Asic Ahead International BV, Bucharest. She took part in a project entitled "WiMAX at system level" and her main responsibility consisted in the implementation in Matlab environment of the PHY layer for mobile WiMAX.

In 2009, Daniela joined the Navigation group within the International Research Centre for Telecommunications and Radar (IRCTR) of Delft University of Technology as a PhD student. Her research activity was conducted in cooperation with The Netherlands Defence Academy. She worked under the supervision of prof. dr. ir. Erik Theunissen and prof. dr. P.J. Oonincx.

At present, Daniela works as a technical consultant within the Mobile and Payment Practice department at UL Transaction Security.

Acknowledgements

Completing the PhD and writing this dissertation has been a long journey. At times wonderful, at times overwhelming.. it has been a true learning experience and I am grateful to many people with whom I collaborated and who supported me along the way.

This research would have never been possible without the support and guidance of Erik Theunissen. Erik has been a great role model and an inspirational teacher. He provided me the vision, the encouragement and the advice necessary to proceed, but also gave me great freedom to pursue independent research. Thank you Erik for your commitment to this project, despite the adversities in the last years! I would also like to express my gratitude to Patrick Oonincx, who has been a steady influence throughout my PhD journey. Patrick gave me the opportunity to do this PhD in the first place. He has guided and supported me with promptness throughout the entire project and his ability to select and approach research problems has often led to key insights and improvements.

This thesis was funded by the Netherlands Defence Academy and I would like to thank this organisation for its generous support. As a member of both NLDA and Delft University, I have been surrounded by wonderful colleagues and both communities have provided a pleasant and rich environment to study and explore new ideas. In TU, a special thank you to my colleagues in the Navigation Group: Joris and Richard. I really enjoyed the good times spent together and our interesting conversations on travelling, sports and life in general. I deeply regret the dissolution of our group and I wish you the best of luck in your new jobs.

I would also like to express my gratitude to all people in the MS3 group: professors, fellow PhD students and technical staff. Furthermore, a special thank you to all people that I had met in IRCTR, back in the days when I first came here. What a great environment to start working in! I feel really fortunate that along the way I have met many great scientists and good friends here at TU Delft. Thank you: Yann, Fra, Dima, Alexey, Tobi, Antonio, Wynand, Norbert, Niels, Stojan, Igor, Riccardo, Karolina, Lukas, Eduardo, Simone, Alex, Christian, Adriano, Carolina, Amer. Thank

you guys for all the lunches, beers and fun times we spent together.

Delft is an international city and one can find a mix of multiple cultures here. I've met so many interesting people: Alicja, Assiyeh, Gian-Luca, Roma, Lukasz, Erio, Rob, Mila and many more. Thank you for the thursday nights in Proeflokaal and for the expat parties. Sport is an important part of my life, so I would also like to take the time to thank all the wonderful people I've meet while climbing: Katka, Peter, Anne-Rieke, Lenny, Plamen, Herve, Satomi, Ania, Matej, Cristophe, Jivko, Alex, Erik, Ryan, Balasz.

A very special thank you to a close group of friends from Delft who gave me support and companionship during all these years. Ana, Adolfo, Dhiradj, Joe, Kedar, Nirali, (and now little Anusha!), Thomas, Tom, Vera.. you've been like a family to me! Thank you for just being there for me!

I cannot forget about my friends from Romania. My dear paranymph Irina, Cornelia, Tudor, Alex, Luci.. though we all share the expat status now and we are spread in different corners of Europe, we still remain good friends and do our best to keep in touch. I am deeply grateful for having you in my life. You keep me sane and grounded. To those of you that are here, thank you for coming to support me. You have no idea how much it means to me.

Last, but not least, I would like to thank my family. My mother Viorica, my sister Alina, my brother in law Cristi and their little Elena for their unconditional love and support. My final thought goes out to my late father, to whom I dedicate this thesis. I hope you are proud of me, wherever you may be!