POLITECNICO DI TORINO

Master's Degree in Communications and Computer Networks Engineering

Master's Degree Thesis

SINS/GNSS Tighty Coupled Integration based on a Radial Basis Function Neural Network

---

Candidate:
Giovanni Prestigiacomo

Supervisor:
Prof. Fabio Dovis

Supervisor from Beihang University:
Prof. Falin Wu

ACADEMIC YEAR 2019/2020

万事开头难。
"*All things are difficult at the start.*"

I would like to thank my chinese supervisor, lǎoshī 发林吴, all the SNARS Research Group and professor Fabio Dovis for their support, their guidance and for helping me undertaking this research.

Beijing has truly given me much more than what I ever asked in my whole life, professionally and mostly personally. If it is true that people leave footprints wherever they go, the ones I left in China belong to a person who found a purpose in his life, who found the courage to boundlessly love and to respect. I will always feel grateful to my boyfriend, Martin, for spotting me among millions of people and for being at my side when we had no coffee bar to go to. Thank you too, Manelle, Georgina and Mariella for this amazing experience.

Thank you, Bens, for your unconditional support, for all the sleepless nights passed in dormitory and for all the times you luckily agreed to go to Ivan's. Thank you too, Gerlando and Ilenia, for your true friendship.

The last big mention is to my family. Thank you from the bottom of my heart for supporting me and loving me no matter what. Thank you Claudia, Stefania for your beautiful mind and for being my partner in crime. Thank you, mom and dad, because you taught me the true meaning of sacrifice and love.

Giovanni

Table of Contents

List of Figures

Acronyms

GNSS
Global Navigation Satellite System

GPS
Global Positioning System

INS
Inertial Navigation System

IMU
Inertial Measurement Unit

DOF
Degree Of Freedom

V2X
Vehicular To Everything

IoT
Internet of Things

NN
Neural Network

RBNN
Radial Basis Neural Network

SIS

Signal In Space

CDMA

Code Division Multiple Access

Chapter 1

Introduction

1.1 INS and GNSS systems

The technologic development achieved during the twentieth century brought massive improvement in the field of the navigation systems. Nowadays precise and accurate navigation is indeed required also for civil applications. Even if several navigation systems exist, the most common ones are Inertial Navigation Systems (INS) and Global Navigation Satellite Systems (GNSS). Both of them have different characteristics and peculiarities and so they can be employed depending on the requirements of the application. A INS can be useful, for example, because it does not require any supporting network: all the informations and measurements are performed and elaborated in loco. On the other hand, even if GNSS is a network based navigation system, it provides much better accurate position approximations on the long run with respect to INS.

1.1.1 Inertial Navigation System (INS)

This system is based on the employment of IMUs (Inertial Measurement Unit), a measurement instrument mounted on the vehicle characterized by having in its core inertial sensors. Even if nowadays INS are evolving their performance with the ading of several different types of sensors, the most common version is the one which is supported by gyroscopes and acceloremeters. These types of INS mount IMU that have six DOF (Degree Of Freedoms), i.e. three accelerometers and three gyroscopes (one for each Cartesian axis)

in order to have three dimensioned measurements. There are lots of different IMUs in commerce. Their performance vary with respect to the technology of the sensors and with respect to the costs.

- **Platform based:** IMUs implement the sensors in such a way that the attitude of the body is kept constant. This is achieved thanks to the employment of a platform which is supposed to mount the sensors and adapt the relative system to the surroudings.
- **Strap-down based:** IMUs does not mount its sensors over a platform: the body keep its own relative reference system. This kind of system is better with respect to the platform based in terms of power consumption but needs some more refined post-processing of the data, which must be converted to the inertial frame.

Practically speaking every INS suffers from error accumulation, though : this cause the performed measurements to drift on the long run and even the most expensive INS carries a non negligible error. A stand-alone INS can be indeed considered as a dead-reckoning system. The cheapest IMU sensors that can be found in market are the MEMS (Micro-Electro-Mechanical System) IMU sensors, which are characterized by their extreme compactness. Sensors are implemented in silicon chips as miniaturized mechanical and electro-mechanical elements, while the electronics are fabricated as integrated circuits [1].

1.1.2 Global Navigation Satellite System (GNSS)

It exploits satellite communications to retrieve the position of user at the ground. Each satellite based sub-system (as GPS, Galileo, BeiDou, GLONASS, etc.) owns a constellation of satellites which continuously orbit around the earth. Generally these orbits are built in such a way that they are able to provide coverage maximization, in order to increase the reliability of the system as much as possible.

User is therefore positioned thanks to triangulation: he converts each of the signal received by the visible satellites (which tipycally have different

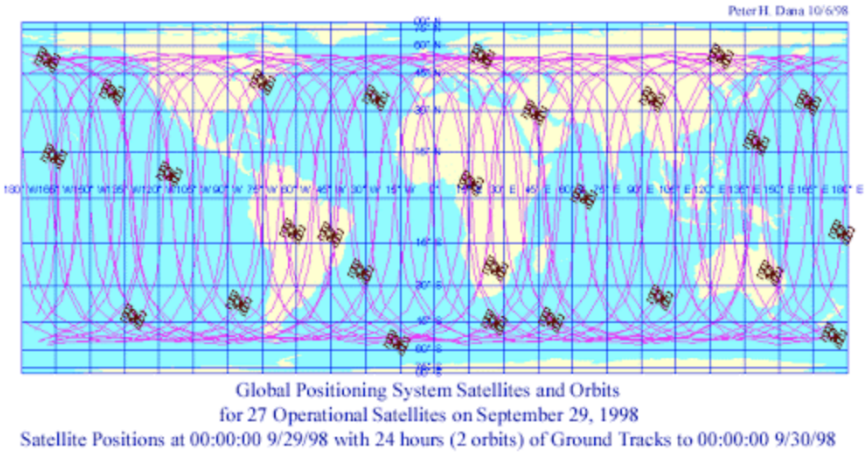

Figure 1.1: GPS coverage

range codes) to get informations about position and velocity by solving mathematical systems made up by collecting the pseudo-range equations related to each visible satellite m.

$$\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + b_{ut} + \hat{\epsilon}$$
$$i = 1, ..., m$$

The previous set of equations are essential to retrieve the position of the user u at a given time t'. $[x_i, y_i, z_i]$ and $[x_u, y_u, z_u]$ are respectively the i-th satellite's and the user's positions. On the other hand b_{ut} represents the clock bias $c\delta t_u$ and the $\hat{\epsilon}$ represents the remaining pseudo-range errors caused by the ionospheric and troposphere effects [2]. The objective of this research is moreover to work on increasing the reliability of the position retrieval process by attenuating those additive errors. This will be done with the integration with INS sensors and with the aiding of a Neural Network.

In GNSS systems attitude can be computed employing multiple antennas at the receiver side (at least three antennas are required to fully retrieve roll, pitch and yaw). Unfortunately satellite communications in GNSS involve really complex assumptions. Since transmissions are based on code division schemes, i.e. each satellite is assigned to a given code, making end to end synchronization one of the main issues, receiver and transmitter should compensate clock bias, clock error and Doppler frequency. Therefore in order to properly retrieve the three dimensioned position of the user and

compensate the clock bias at the same time, there must be at least four satellites in visibility, i.e. a system made up by four equations and four variables must be solved.

1.2 INS/GNSS Integration

Navigation systems are widely used in lots of applications and for lots of purposes. For example, the ETSI standards for V2X communications totally rely on the positioning capabilities of the nodes: positioning is exploited to perform packet addressing (Geo-Networking) [3]. This technology allows the vehicular system not to need a coordinating infrastructure but instead allows vehicles to spread packets (with different methodologies) at the network layer based on the geographical position of the nodes. Therefore packets are basically forwarded and addressed to certain geographical areas. Though it seems to be very hard to understand, this technology allows V2X communications to be more secure, accurate and efficient.

Positioning and navigation are key features also in the IoT, where there exist some strict requirements in terms of costs and in power consumptions. When cloud-based positioning [4] is not employed these systems are forced to employ very low cost and sometimes very inaccurate sensors and positioning systems, because nodes are supposed to elaborate their sensors informations and not to transfer the signal processing to a centralized cloud server.

Whatever the application would be, it is quite clear that more and more exigent requirements are pushing the navigation technologies to a point where the performance of such systems should be optimized as much as possible. Unluckily it is difficult to make a single navigation system meet high requirements on different main characteristics (reliability, accuracy, availability, real-time performances and costs). A stand-alone INS indeed is ideal in terms of availability, has good anti-jamming capabilities, real-time performances and costs. Still it suffers error accumulation and its outcome drifts as time elapses regardless of the error model. Therefore INS has lack of reliability and accuracy on the long run. On the other side a stand-alone GNSS can reach good results in terms of accuracy and reliability but when used in some environments (like in urban canyons) it experiences lack of

availability and real-time performances.

In order further improve the capabilities of a position estimator, integrated navigation becomes the direction where to lead the development and the research on the navigation systems. Integrated navigation is based on combining the informations of two or more systems in order to better refine the position approximation. This is achieved by mutual error compensation. The perk of exploiting such a system is to combine together the advantages of different navigation systems and satisfy more strictly requirements. For instance, an integrated system gives better results at the same costs with respect to a stand-alone one. Integration requires lot of signal and data processing, though. Therefore different level of accuracy can be reached depending on the grade of integration of the systems involved.

It is difficult to give a detailed classification of integrated navigation systems. Indeed, they are implemented with respect to the requirements of the applications for which they are built. The most common and widely used integrated navigation systems is the one obtained by the combination of INS and GNSS, because both the systems can be very low cost and available to be applied much easily, especially for civil applications. As it will be shown in the next section, GNSS/INS integration can be implemented in mainly two ways: loosely-coupled one and tightly-coupled one. The difference depends on the level of integration of the systems.

1.3 Neural Network

Neural networks are one of the most widely used tool in nowadays technologies. Their concept is based on the reproduction of the human neurons. Since it has become a general purpose technology, it can be used in a lot of applications, not only for Artificial Intelligence (AI) or image recognition. The core component of Neural Networks is the neuron itself : logical entities that receive inputs and produce outputs according to a given activation function [5]. Neurons form a networks thanks to connections (as shown in figure 1.2), which are responsible to provide the output of one neuron as input of another neuron. Connections have given weights, that assign a

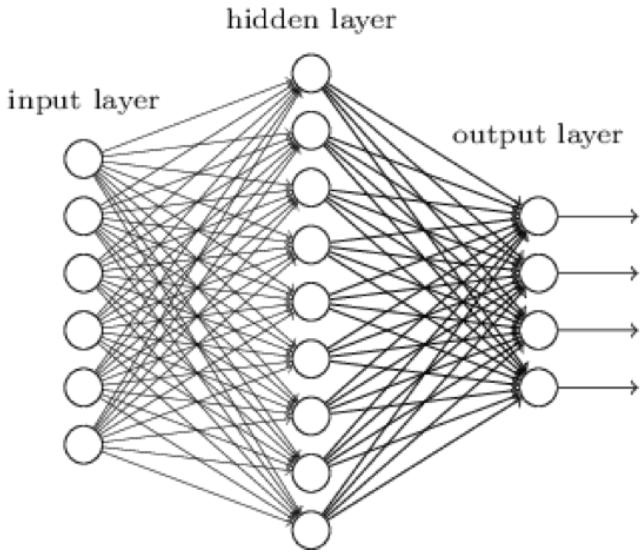

Figure 1.2: Example of a fully-connected Neural Network

certain level of importance to some outputs with respect to other ones. The combination of thousands of neurons, all connected with each other, lead to the generation of a very complex network which can classify of approximate data, as a human brain. Different types of NN exists and different for their topology and from the activation function employed. Each Neural Network have to be trained, though. Training is the process employed to adjust weights and biases associated to the activation functions of each neuron using a given data-set. Once the NN is properly trained, it may be ready to receive inputs and elaborate them in order to give them a complex classification/approximation.

1.4 Related research

INS/GNSS Integration has become one of the main issues in the navigation technologies. Mass products indeed need more and more reliable and accurate systems, according sometimes to very low cost requirements. The benefits of integration has moreover been proven for lots of applications, because it can help the overall system to enjoy the advantages of both the navigation

systems. But there are still some gaps to fill. Since there are some scenarios where there is a lack of performance, the related research needs to improve and study new technologies. The limits of INS/GNSS integrated systems are shown when for example the reliability of one of those system decreases like when there is a GNSS outage when in indoor scenarios (or more likely when a car is approaching a tunnel in the highway). In these case all the responsibility is left on the INS and as it is known this can be a problem in the long run, because the errors on the position retrieval increase more than linearly.

It is proven that tightly-coupled INS/GNSS Integrated systems give more accurate position approximation, even if the supporting algorithms are more complicated than the loosely-coupled ones. In this scenario, research tried to improve the system with the aiding of Neural Networks and results show that they are successfully able to compensate GNSS outages by emulating GNSS pseudo-ranges and pseudo-range rates [6] and attenuating a lot the error increases on the measurements. Since NN require to be trained with proper data, and since this training may require non-negligible amount of time, it is still difficult to understand how to make the system to work in an efficient way minimizing the computational cost of the supporting algoritm.

1.5 Research objective

This work addresses a new approach on integrated navigation combining INS, GNSS with a RBNN (Radial Basis Neural Network). RBNN is a special kind of NN which has a very simple topology with only one hidden layer. The purpose of this research is to build a system able to compensate in a better and lighter way the overall position approximation during GNSS outages, which is achieved by aiding the INS informations to a RBNN. If the NN is properly trained, it will be able to predict and substitute with a certain accuracy the GNSS informations needed to approximate the position of the user keeping a good accuracy. RBNN is a type of neural network characterized by having a very simple topology which is fast to be trained with respect to traditional NN. Moreover it is used also for approximating functions, which is what the system requires since it must approximate the

pseudo-range and pseudo-range rate difference evolutions.

The next section will give an overview of the working principles of GNSS, INS and their combined integration (tightly and loosely-coupled). The section 3 will moreover give a detailed overview of the RBNN and it will also give explanations about why it is convenient to use this kind of NN in navigation systems. Furthermore, the purpose of the section 4 is to go in detail on the actual implementation of the navigation system, showing how the NN is actually employed in it. In section 5 experimental results will be shown. At the end, section 6 will give tips and suggestions about future implementation of the proposed navigation system.

Chapter 2

INS/GNSS Integration

2.1 Overview of GNSS

The “Global Navigation Satellite Systems” refers to all the navigation systems that are based on satellite communications. GPS, Galileo, GLONASS and BeiDou are just examples of different employments of the navigation system. Each of them exploits different signal transmission technologies and owns its own satellite constellation. The position of the user at the receiving end is provided thanks to what is called trilateration: a computational process which is performed in order to retrieve the pseudo distances ρ_s (commonly known as “pseudo-ranges") between the user and the current visible satellites. A receiver theoretically needs just three satellites to compute its position, which is retrieved as solution of an analytic system geometrically represented by the feasible intersection point between spheres which have the position of the satellites as their center, as shown in figure 2.1. In practice, though, every GNSS receiver needs at least four satellites in view in order to perform timing corrections and compensation.

Receiving end knows the positions of the visible satellites because the position retrieval process is based on the assumption that satellites continuously broadcast their orbit informations using the navigation message, usually denoted with Signal-In-Space (SIS). This allows the receiver to easily obtain the pseudo-ranges needed to perform the position estimation. The distances between receiving end and satellites are furthermore referred with “pseudo” since they are affected by the clock biases that exist both at the receiver and at the transmitter sides because of a lack of clock synchronization. The

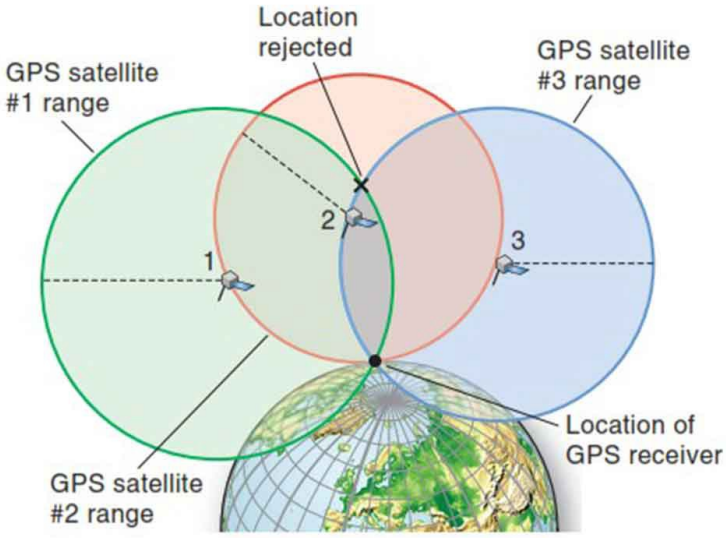

Figure 2.1: GNSS trilateration

pseudorange equation can furthermore be written as stated below.

$$\rho^s = r^s + c(\delta t_u - \delta t^s) + I + T + \epsilon$$

- ρ^s is the pseudo-range related to satellite s.
- r^s is the effective distance between receiver and satellite s. Assuming $P_u = [x_u, y_u, z_u]$ as the position of the receiver u and $P^s = [x^s, y^s, z^s]$ as the position of satellite s, the distance r^s can be expressed using a simple geometrical distance formula.

  $$- \; r^s = \sqrt{(x^s - x_u)^2 + (y^s - y_u)^2 + (z^s - z_u)^2}$$

- δt_u and δt are respectively the receiver and the satellite clock biases. Since these values affect the distance measurements performed at the receiver, they need to be multiplied with the light constant c. In most of the textbooks, biases both at receiver and transmitter side are usually referred with a unique value δt.
- I and T are respectively quantities affected by the ionospheric and the tropospheric delays. The ionospheric effects are quite peculiar because they strongly depend on the frequency of the transmitted signal.

When it goes through a ionospheric region where the signal is affected by ionospheric scintillations, the delay contribution would be bigger. On the other side the tropospheric delays depend on the temperature, humidity, pressure of the troposphere. If the ionospheric delay can be somehow removed thanks to multi-frequency measurements, the tropospheric delay can be more complicated to manage because it needs accurate error modeling. Athmospheric error attenuation is still one of the biggest deals with satellite communications because it is not possible to predict in an accurate way the weather conditions of both the ionosphere and the troposphere.

- ϵ is a quantity that collects all the unmodeled effects such as multipath and measurement errors.

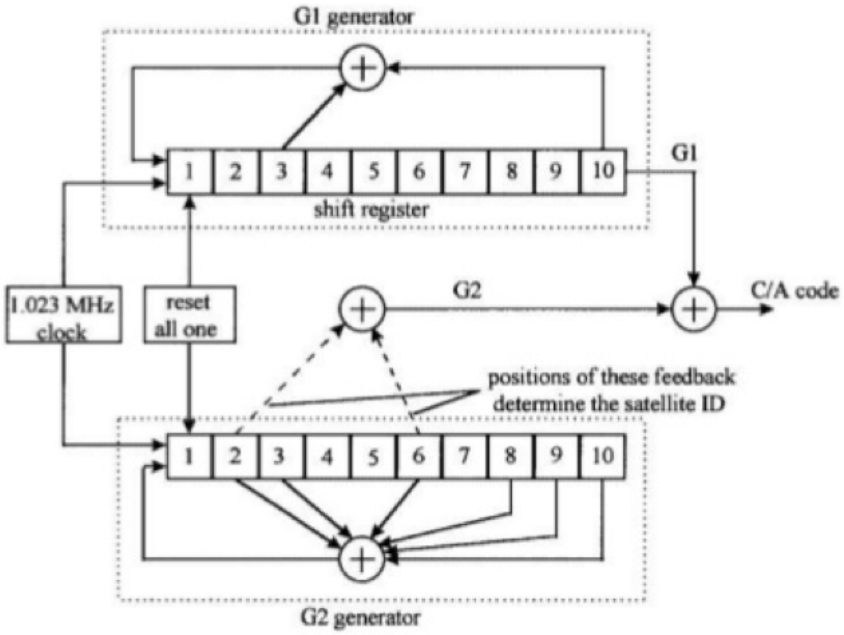

Figure 2.2: GPS C/A range code shift registers

In all the satellite navigation systems multiple access problem is solved thanks to CDMA (Code Division Multiple Access). Each satellite is assigned with a given range code taken from a set of orthogononal codes, which let the receiver proper demultiplex the transmitted signals and identify all the

satellites in visibility. Range codes are usually generated using a BPSK modulation (i.e. the transmitted values are sequence of [0,1] bits). GPS for example uses convolutional C/A codes, which are generated thanks to the combination of two linear shift registers.

Each satellite is assigned to a set phase selectors, which can be spotted in figure 2.2 as the feedbacks of the shift register below. The selection of different feedbacks leads to the generation of different range codes. Thanks to this procedure orthogonality and unique satellite identification is guaranteed. Acquisition of the signal is indeed performed by computing the autocorrelation functions (which ideally have fixed known values) and after by seeking peaks in them. Peak in the autocorrelation function means that the SIS contains a signal transmitted by the satellite which is assigned with the range code by which the autocorrelation function is computed with.

At the end, the receiver is able to properly identify the satellites in visibility and demodulate their SIS, which carries their ephemeris. These data is collected in the navigation data, the binary frame continuously broadcasted by each satellite. These message contains all the informations that are useful for the position retrieval (ephemeris, clock bias parameters, relativistic corrections, etc.).

2.2 Overview of Inertial Navigation Systems

Inertial Navigation Systems, together with GNSS, are the most commonly used navigations systems. These systems are typically equipped with IMU sensors, able to obtain informations about the motion of the vehicle. Using INS may be convenient because these systems are self-contained: they do not need any supporting network to let them work since all the data is provided and processed in loco. Stand-alone INS performs dead-reckoning thanks to the employment of accelerometers and gyroscopes. In order to get three-dimensional informations, a INS needs a triade of both accelerometers and gyroscopes, which are typically put alongside the Cartesian axes. Accelerometers measure the specific forces along the three axes, which can be easily translated into accelerations, velocities and then positions. Gyroscopes

on the other hand provide attitude informations of the body with respect to the navigation frame. This is denoted with ω_{ib}^{b}, angular velocity of the body frame b with respect to the inertial frame i.

2.2.1 Mechanization

A common problem with strap-down INS (which is the most commonly used ones) is that the measured data needs to be transformed with respect to the proper reference system and also needs to be processed in order to be understandable and useful to the navigation process. The specific forces measured by the accelerometers need to be compensated with the gravitational model. On the other hand angular rates needs to be compensated too, because they are measured with respect to the navigation frame, which is influenced by the effects of the rotation of the earth. However, it is understandable that a stand-alone IMU itself can not provide coherent data, because the measured informations need to be properly translated into attitude, position and velocity exploiting severe mathematical assumptions. This is why the IMU that are mounted in the INS always need to be implemented along a proper navigation algorithm which is also commonly called mechanization process.

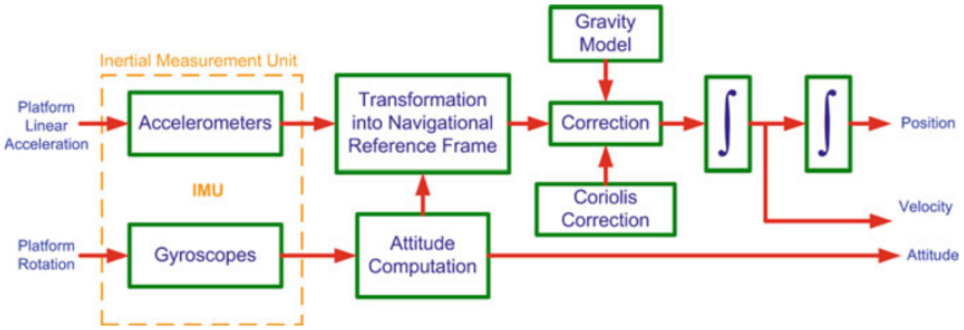

Figure 2.3: Block diagram of the INS mechanization

Mechanization uses kinematic models to transform the data into coherent position, velocity and attitude frames. Kinematic models are more useful and easy to implement than dynamic ones because dynamic modeling requires

specific force and mass models (i.e. in order to be effective, one should model all the forces acting upon the object moving). Starting from the specific forces measured by the IMU, it is possible indeed to retrieve the position and the velocity of the object just by integrating, following the scheme shown in figure 2.3. Summarizing, it is possible to express a relation between these components as shown in the following equations.

$$r(t) = [x(t), y(t), z(t)]$$

$$v(t) = [\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}] = [v_x(t), v_y(t), v_z(t)]$$

$$a(t) = [\frac{d^2x}{dt^2}, \frac{d^2y}{dt^2}, \frac{d^2z}{dt^2}] = [\frac{dv_x}{dt}, \frac{dv_y}{dt}, \frac{dv_z}{dt}] = [a_x(t), a_y(t), a_z(t)]$$

2.2.2 Quaternion and rotation matrix

Accelerometer and gyroscope measurements are typically taken from the body frame, which in principle does not coincide with the inertial frame of the object. Because of this, the specific forces measured by the accelerometer need to be transformed in order to be coherent by multiplying the measurement vector with a given rotation matrix. On the other hand, in order to convert the angular rates measured by the gyroscope to the proper frame, the rotation matrix R is multiplied with the skew-symmetric matrix Ω associated to ω_{ib}^b. In order to parametrize the rotation matrix R several methods exist and the quaternion approach is the most used one. A quaternion is a four-dimensioned representation of the rotation matrix which is related to a rotation angle $\theta = \sqrt{\theta_x^2 + \theta_y^2 + \theta_z^2}$.

A quaternion is said to be consistent only if the following condition holds.

$$q_1 + q_2 + q_3 + q_4 = 1$$

Furthermore, the rotation matrix is computed using the following relationship, which directly associates the quaternion to it:

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix} =$$

$$= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 - q_4^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}$$

2.2.3 Inertial sensors errors

Errors affect the position, velocity and attitude approximations even in the most costly INS. As it is shown in the first section, a stand-alone INS performs indeed dead-reckoning : a navigation technique that approximates the position in a iterative manner by calculating the distances from the previous estimated one. Since these algorithms do not have position compensation feedbacks, it is common that errors accumulate on the long run. Errors cause a drifting phenomena of the measurements made at the mechanization level that lead to a misleading results even for the most accurate INS systems.

In order to build a system able to navigate in the most accurate way, error compensation feedbacks must be implemented. In order to do so, INS error modeling must be performed.

Gyroscope constant bias	0.01 deg/h
Angular random walk	0.001 deg/$\sqrt{h}$
Accelerometer constant bias	50 μg
Velocity random walk	$10\mu g/\sqrt{Hz}$
Scale factor error	10 ppm
Askew installation error	10 arcsec

The table above shows common errors observed using a typical INS system. These values are used to perform error compensation in the model of the

system of this work. On a mathematical point of view, error is modeled as a gaussian random variable, with its mean value and its variance.

- The gyroscope and accelerometer **constant biases** represent the values that are measured on average when both the sensors are not being used (i.e. when they are still). These average biases are always present in any measurement performed by the inertial system. Moreover they can be represented as the mean values of gaussian random variables.
- The gyroscope and accelerometer **random walks** represent on the other hand the error variances that affect each measurement.
- The **scale factor error** of a system denotes how the output is related to its input. Generally speaking the output is the result of a linear effect, in which the output is proportional to the input and scaled [7]. Mathematically speaking it can be considered as the slope of a function that represents a straight line.
- The **askew installation error** is a quantity related to the non-orthogonality of the sensors mounted in the INS. For example, this problem cause the gravity effects to be present in all the components of the measured values by the IMU. Careful manifacturing can somehow attenuate this problem, at the expense of an increase of the costs.

2.3 Navigation Integration

As stated in the previous chapter, the optimization of INS/GNSS navigation integration systems has nowadays become one of the main purposes of scientific research. But how actually integration is performed? How to merge data informations in a coherent manner in order to let those systems cooperate? The answer to this question is hidden in the structure of these systems. INS/GNSS navigation systems are indeed classified in two categories : loosely-coupled and tightly-coupled. The main difference between those implementation is the level of cooperation. Previous research show that as the indipendence level of those systems increase, as the accuracy of

the navigation algorithm decrease. But the accuracy increase comes with a cost, which is the complexity of the system and the computational complexity of the algorithm that implements it. The next pair of sub-sections will give an overview of the pair of navigation integrated system mentioned above.

2.3.1 Loosely-coupled implementation

The loosely-coupled implementation of INS/GNSS navigation system is the simplest one. INS and GNSS systems indeed cooperate together to approximate the position as different and separate systems. In order to improve the position estimation a optimal estimator (i.e. a Kalman Filter) is used. Loosely-coupled integrated systems are robust and simple. In their open-loop form (i.e. without INS feedbacks) they are indeed able to provide three indipendent navigation solutions : the GNSS one, the INS one and the GNSS/INS integrated one.

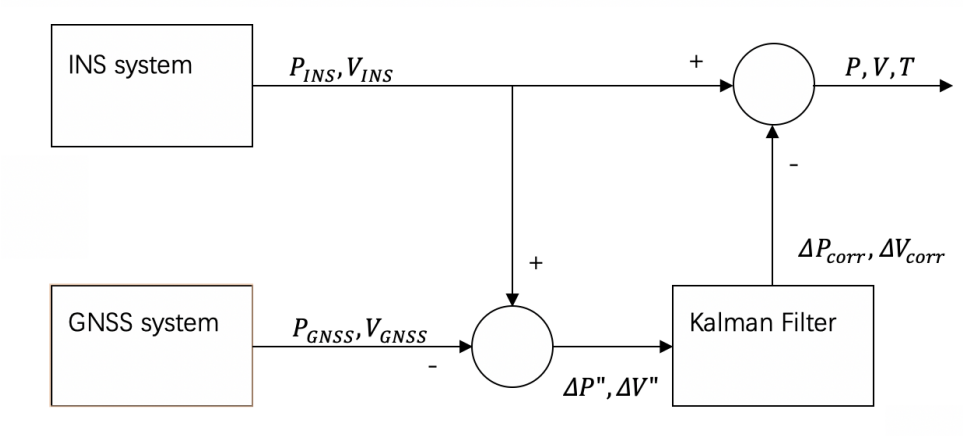

Figure 2.4: Block diagram of an open loop loosely-coupled GNSS/INS integrated system

The block diagram shown in figure 2.4 resumes the working principle of such a system. GNSS and INS provide indipendent navigation solutions. These are then used to approximate a compensation ΔP_{corr} and ΔV_{corr} through the Kalman Filter by means of differences ΔP" and ΔV". The compensation is then subtracted from the INS navigation solution in order

to compute the integrated navigation solution. As it is possible to see, the level of integration is very high : INS system do not interfere in the GNSS receiver. Moreover this system is more simple and robust. Since GNSS still needs at least four satellites in visibility to navigate, the GNSS outages are more frequent for these kind of systems.

2.3.2 Tightly-coupled implementation

The tighly-coupled implementation on the other hand is more complex, because INS and GNSS cooperate in one merged system. Integration is indeed performed at a pseudo-range level since the GNSS and INS collaborate more deeply with each other to improve the error compensation estimation and predict the position in a more accurate way.

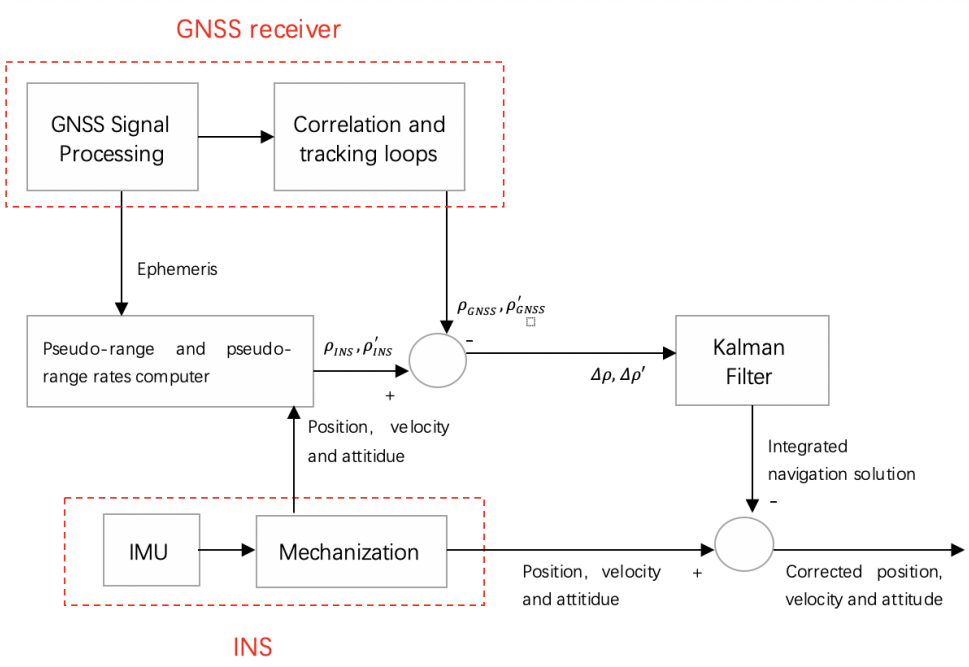

Figure 2.5: Block diagram of a tightly-coupled GNSS/INS integrated system

As it is possible to see in figure 2.5, the block diagram of a tightly-coupled

GNSS/INS integrated system is much more complex than the loosely-coupled one. In this case the level of cooperation of the INS and the GNSS system is definetely higher, because the integration is done at a pseudo-range and at a pseudo-range rate level. What is interesting is that the optimal estimator approximates no more the differences of the navigation solutions of both the systems (as the loosely-coupled version do). On the other hand it estimates the navigation solution from the difference $[\Delta\rho, \Delta\rho']$ between the pseudo-ranges retrieved by the GNSS receiver $[\rho_{GNSS}, \rho'_{GNSS}]$ and the approximated ones that are computed from both the INS measurements and the satellite ephemeris obtained after the GNSS signal processing $[\rho_{INS}, \rho'_{INS}]$.

This approach is furthermore more convenient rather than the loosely-coupled one because it can provide an accurate navigation solution even if the number of visible satellites is below four. This is the reason why this work focuses only on the tightly-coupled implementation. GNSS outages are a big problem in urban environments, urban canyons and in all the situation where the satellite availability is low. This is the reason why using a system that works also with few satellites is a big improvement in navigation.

Chapter 3

Neural Network

3.1 Radial Basis Function Neural Network

As stated in the previous chapters, NN are nowadays source of interests for a wider and wider class of applications. This work focuses on the implementation of a Radial Basis Function NN in the GNSS/INS Integrated Navigation system. This NN has been chosen for two main reasons. First of all because RBNN is a special kind of NN that is characterized by its fast learning. Second, this NN is mostly used for function approximation purposes and this is one of the aim of the research explained in this paper. During GNSS outages, tightly-coupled integrated system is indeed no more able to provide an integrated solution because it lacks of the evolution of the $\Delta\rho$ and $\Delta\rho'$ values. In this work the RBFF is employed to give a coherent approximation of the evolution of such a function when GNSS can no more give a navigation solution.

This type of NN has been thought with a totally different approach (with respect to the standard NN) because its design is viewed as a curve-fitting approximation problem in a high dimensional space. The topology of such a NN is moreover much more simpler than the normal ones. The layer architecture, which is showed in figure 3.1, is indeed classified as follows.

- **Input Layer**: this layer is composed by as many nodes as the dimension of the vector to approximate or classify. Its duty is to connect the input data to the hidden layers.

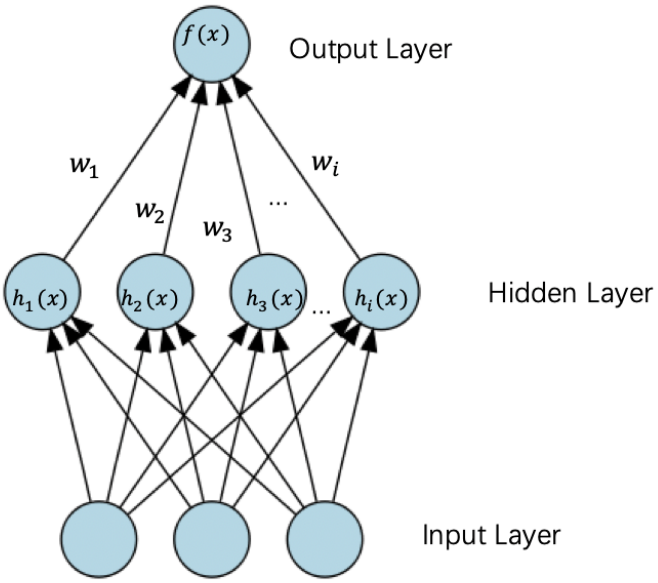

Figure 3.1: RBNN topology

- **Hidden Layer**: it is characterized by its high dimensionality (i.e. it may have a lot of nodes). The duty of its nodes is to provide a set of basis function that act over the input passed from ther input layer. The output of such a computation is then passed to the output layer.

- **Output Layer**: this layer provides the output approximation after having computed the linear combination of the values passed by the hidden layer. The dimension of the output layer (i.e. the number of nodes) depends on the requirements.

3.2 Radial Basis Function characterization

Hidden and output layer of such a Neural Network form together what is known as Radial Basis Function. The linear combination of the hidden layer outputs is weighted by the w_i values associated to each connection going in the output layer. Each node of the hidden layer is moreover associated with

a bell shaped function (i.e. Gaussian function) $h_i(x)$. The output $f(x)$ of the NN can be furthermore computed as follows.

$$\sum_{i=1}^{m} w_i h_i(x)$$

$$h_i(x) = exp(\frac{-(x - c_i)^2}{r_i^2})$$

The Gaussian function associated to each node i of the hidden layer is characterized univoquely by its c_i and r_i values. These values are respectively the center and the radius (in stochastic processes this value is also known as standard deviation) of the $h_i(x)$ function. Known this, the design of a Radial Basis Function Neural Network is reduced to understanding how many neurons the hidden layers should have and compute the optimal parameters w_i, c_i and r_i associated to each neuron.

3.3 Learning strategies

As stated in the previous sections, RBNNs are characterized by their strong fast learning processes. But how can the learning process be optimized? What are the best strategies to train such a NN? The answer to these question is brought by three main learning strategies, that are explained in the next sub-sections.

3.3.1 Random selections of the centers

This learning strategy is quite simple to implement. The hidden nodes number is fixed. Their centers are picked up from random values taken from the data-set used to train the network. The only values that should be parameterized are the output weights (the coefficients which multiply each $h_i(x)$ function). The only drawback of such a strategy is that in order to obtain a good learning, the network should be trained with a quite large data-set.

3.3.2 Cluster-based learning

This strategy is also known as "self-organized learning" and it is considered a hybrid learning technique. This strategy is based on applying a k-mean clustering process to the training data-set in order to understand the number of hidden nodes of the hidden layer and to retrieve its centers. The k-means is a clustering algorithm which splits data by means of its different attributes in an iterative way. At first it identifies K partitions that can be determined randomly. After computing the centroid for each of them, the algorithm identifies again new partitions by associating nearest centroids. This algorithm is convenient to use because it converges quickly but it does not guarantee that the global optimum will be reached. Since it can be applied multiple times, k-means is usually applied several times using different initial partitions and the best solution is chosen afterwards. But still it is not the best solution to use because this process will add some delay to the learning process.

3.3.3 Supervised selection of centers

This is also a hybrid learning strategy based on the use of a Least Mean Squared algorithm which is employed to determine in a unique and supervised way all the free parameters of the network. Supervised learning is a learning technique which is based on the adaptation of the variables of the network on several input/output samples (which are also known as training samples). During the training, the algorithm will search for given patterns in the data correlated to the desired output [8]. Even if this technique is the most use one also for machine learning purposes, for this kind of applications some drawbacks may be spotted. For example it is suggested to change the supervised learning algorithm depending on the level of heterogeneity of the data-set. If the features of the data-set are indeed a lot different with each other, some learning algorithms may be more advised than others because the converge more quickly. Moreover if the data-set contains some redundant informations, the performance of some learning algorithms may be worse than others.

Chapter 4

Tightly-coupled Integration with Neural Network aiding

4.1 The system model

As stated in the previous chapters, this work focuses on the implementation of a Radial Basis Function Neural Network on a GNSS/INS tightly-coupled integrated navigation system. The objective of such an employment is to compensate in the best way possible the navigation errors during GNSS outages. While GNSS is not available because there is no satellite visibility, a typical integrated navigation system would only rely its approximations on the INS measurements. No matter if the GNSS/INS system is tightly-coupled or loosely-coupled. The NN implementation may be very important because if properly trained, it can substitute the GNSS performances during those outages.

This research focuses only on the tightly-coupled implementation for several reasons. One of these is because it is known that tightly-coupled systems are more accurate than loosely-coupled ones. Another reason is that tightly-coupled GNSS/INS integrated systems are characterized by having an outage probability much lower than the loosely-coupled one. Loosely-coupled integrated system is indeed composed by indipendent GNSS and INS receivers. This means that the GNSS receiver needs as usual at least four

satellites in visibility, as explained in chapter 2.1. A tightly-coupled system on the other hand is characterized by a different working structure. GNSS and INS does not work indipendently but cooperate in an active and deep way. This means that a tightly-coupled system is able to approximate the navigation position of the user even if the number of satellites is below four [9]. A tightly-coupled system is furthermore able to retrieve the position of the user with cooperation between GNSS and INS when the loosely-coupled one may be not.

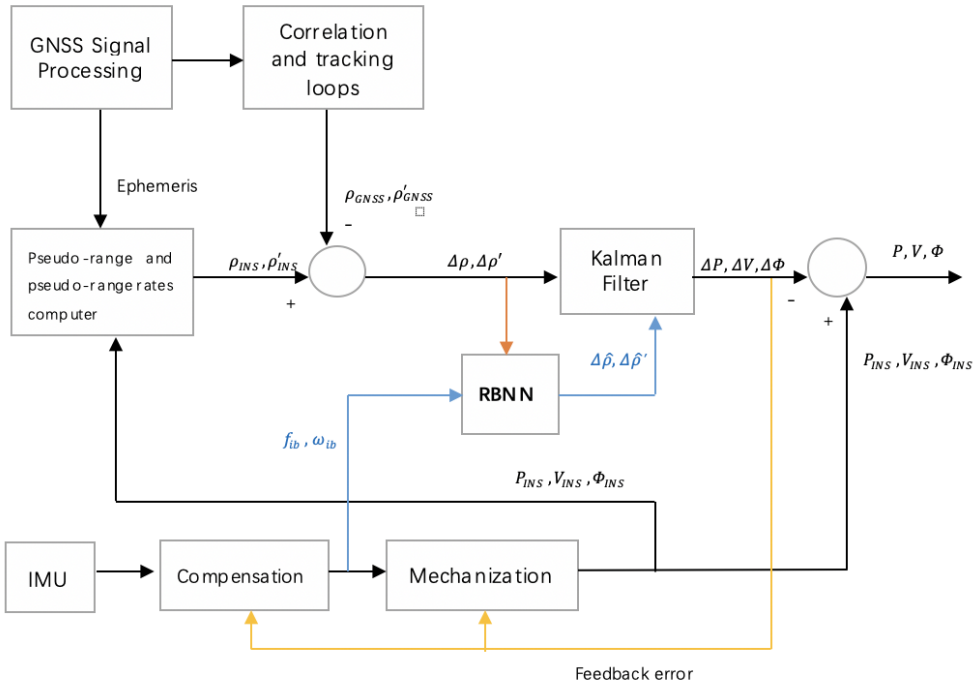

Figure 4.1: System model

Figure 4.1 shows the system model in its general scheme. It is possible to observe that it is very similar with the standard tightly-coupled integrated navigation system, shown in chapter 2.3, but with a slightly different structure. A RBNN (Radial Basis Neural Network) is added. This block takes useful data only when GNSS has an outage (blue arrows), otherwise it does not predict any output but it only takes the training data $\Delta\rho$ and $\Delta\rho'$ that is necessary to the learning process (red arrow). The output of the Kalman

Filter is furthermore used to perform a feedback error correction needed by the INS to improve its accuracy. This is proven by the previous research conducted on the field [10] and it is highlighted by the yellow arrow.

4.1.1 First generalized scheme

In order to properly understand the working principles of the system model, it should be shown in its two generalized schemes. The first scheme represents the system model when the GNSS is working as it should (i.e. when there are enough satellite in visibility). Figure 4.2 shows the block diagram of the system on its first generalized scheme. In this case the RBNN is not needed because GNSS is able to retrieve the position of the user properly. At the end the scheme of the integrated navigation system is very similar to the classical tightly-coupled implementation, shown in the previous sections.

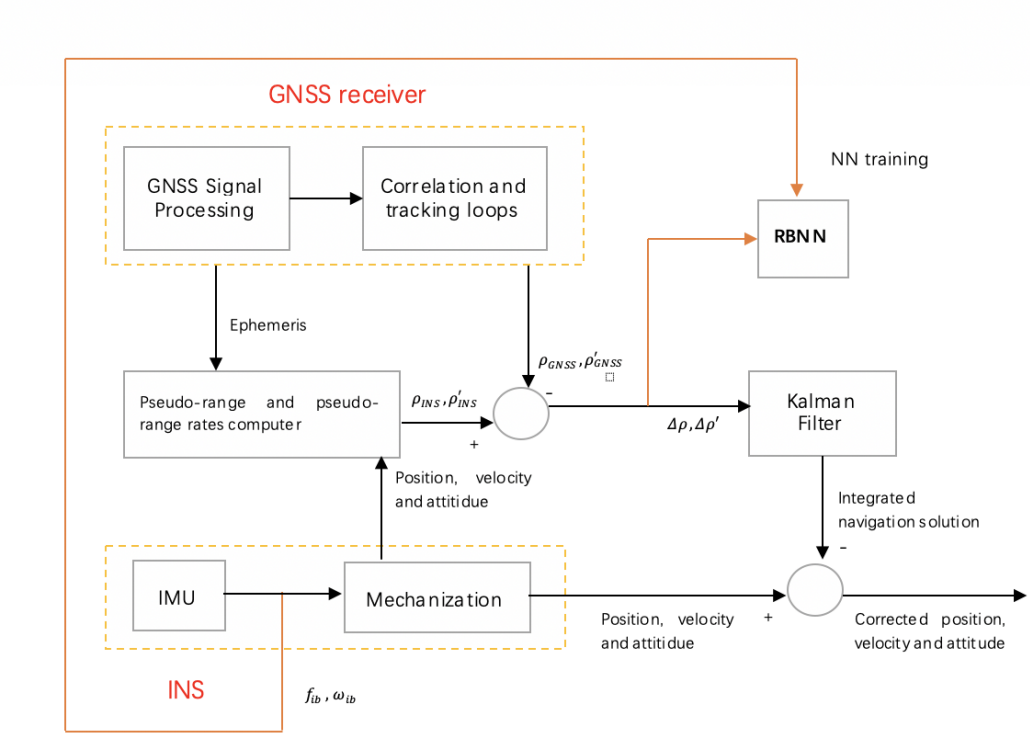

Figure 4.2: First generalized scheme (GNSS working properly)

It can be moreover noticed that the pseudo-range differences computed at the first stage of the integration are used to feed the RBNN. Training can be done in a offline and online way. The offline training is also called « supervised training ». It consist in training the NN when the system is not operating in an active way. Data is collected in bunches and fed to the NN in a supervised way. Online training consist in training the NN while the GNSS is working and the system is operating. In other words, the network is fed with data as long as there is no outage and as long as the system is working properly.

4.1.2 Second generalized scheme

The second generalized scheme shown in figure 4.3 represents the system while there is a GNSS outage. In this case the system model is simplified because the GNSS receiver is no more considered. Its performances are indeed substituted by the RBNN block, which is fed with the error compensated measurements performed by the IMU block.

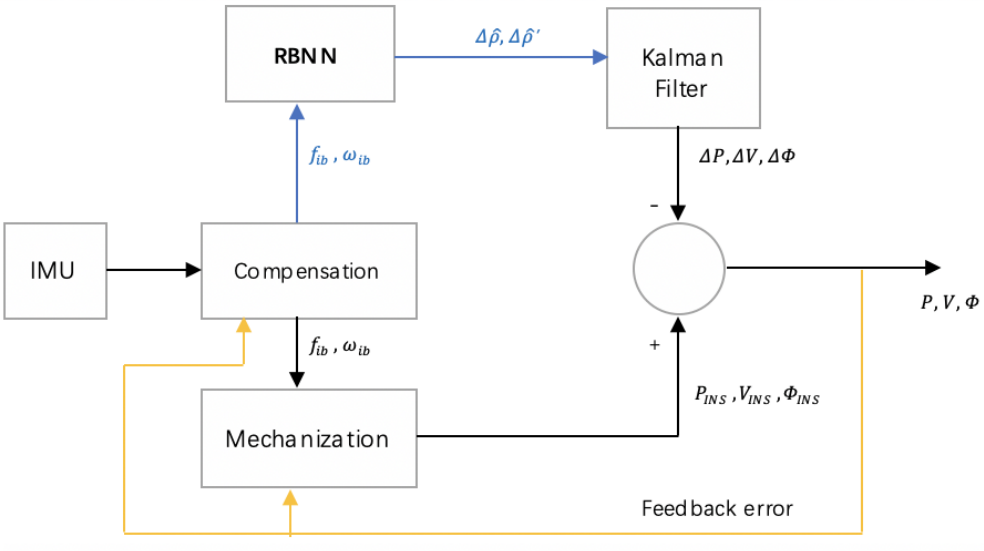

Figure 4.3: Second generalized scheme (during a GNSS outage)

4.2 Working principles of the RBNN

As proposed in the schemes shown in the previous section, it is possible to observe that the RBNN has a different behavior depending on its generalized scheme (i.e. depending if it is on its training or on its active modality). When the GNSS is working properly the RBNN has the possibility to be trained, but the system shown in this research works with offline training, so it simply goes in stand-by. When there is a GNSS outage caused by a lack of satellite visibility, the RBNN switches into its active mode and predicts the pseudo-ranges $\Delta\hat{\rho}$ and the pseudo-range rates $\Delta\hat{\rho}'$ needed to estimate the position using the Kalman Filter. As stated in the previous sections, the RBFNN is moreover characterized by having a pretty quick training. This means that in order to have good performances not a lot of pair samples are needed.

Offline training is achieved using the supervised selection of centers, shown in section 3.3. This algorithm adapts the number of centers and all the NN parameters depending on pairs of input and output. Since the objective is to be somehow able to predict $\Delta\hat{\rho}$ and $\Delta\hat{\rho}'$, the offline training is achieved feeding the NN with the following pairs:

$$[(f_{ib}, \omega_{ib}), (\Delta\hat{\rho}, \Delta\hat{\rho}')]^T$$

Where f_{ib} are the accelerometer measurements before the mechanization still denoted with the body frame and ω_{ib} the gyroscope ones. The correlation between the pairs is proven by the fact that the differences $\Delta\hat{\rho}$ and $\Delta\hat{\rho}'$ are computed thanks to the approximated pseudo-ranges ρ_{INS} and $\hat{\rho}'_{INS}$.

$$\Delta\rho = \rho_{GNSS} - \rho_{INS}$$
$$\Delta\rho' = \rho'_{GNSS} - \rho'_{INS}$$

Those approximated pseudo-ranges are computed in the pseudo-range and pseudo-range rate computer block, shown in the previous chapters. These values are directly related to the IMU measurement performed at each iteration of the algorithm. In the pseudo-range rate formula, f_D^s denote the doppler

frequency measured in the GNSS receiver, and it is related to the difference of the velocities of both the satellite and of the velocity measured by the IMU.

$$\rho_{INS} = \sqrt{(x_{sat} - x_{INS})^2 + (y_{sat} - y_{INS})^2 + (z_{sat} + z_{INS})^2}$$
$$\rho'_{INS} = -\lambda f_D^s$$

Chapter 5

Eperimental results

5.1 Initialization of the system

In order to properly initialize the simulation of the system model shown in the previous section, it will be needed to fix univoquely the IMU measurements related to the motion of the user. Setting the accelerometer and gyroscope measurements means to determine in a unique way the motion of the user. The figures 5.1 and 5.2 show the accelerometer and gyroscope measurements that this work used to perform the simulation of the system. These measurements are not ideal ones. They are taken from a real trajectory and so they already are subjected by the errors. This is the reason why their evolution is noisy.

5.2 Testing of the system without RBNN

After having set the IMU measurements, the system has been tested with its standard tightly-coupled solution. For simplicity reasons, since it is not the purpose of this work, the satellite constellation is supposed to be fixed. This means that in this particular ideal case the satellites have a steady evolution and they do not move along their orbits. The GNSS system taken into account is moreover the GPS one. IMU measurements are supposed to give sensor outputs with a rate of $t_s = 0.01\ s$, while GPS gives approximatively coherent data each second. Implementing the system using one IMU output sample per time leads to poor resistance from the circular

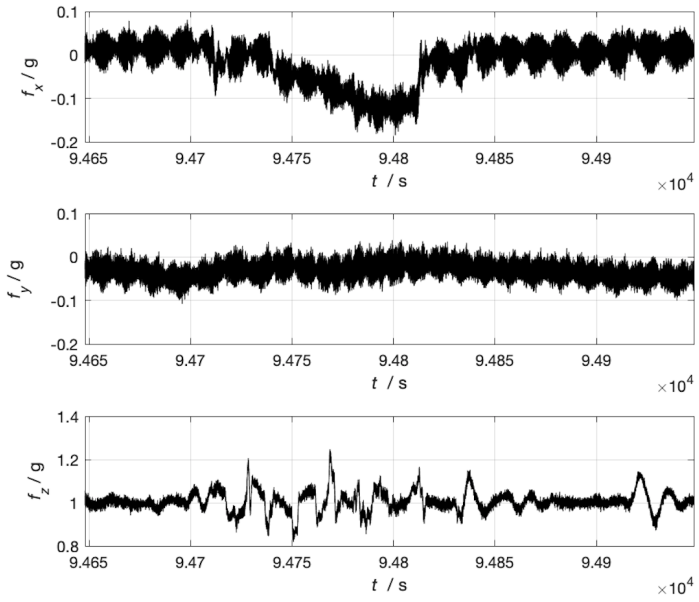

Figure 5.1: Noisy accelerometer IMU measurements

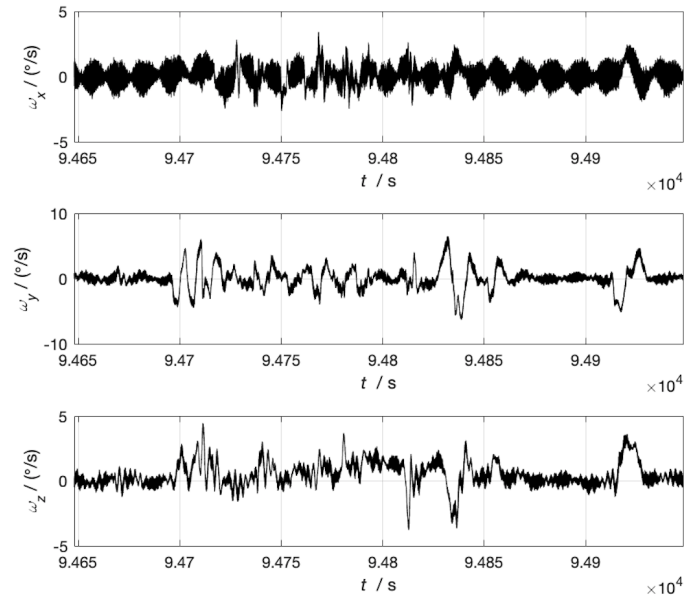

Figure 5.2: Noisy gyroscope IMU measurements

motion error. This problem leads to very misleading results. In order to get good performances at least three or four samples should be used to update INS during the mechanization. In this work 5 samples are taken each time. This means that INS gives an update of its measurements each $t_{mech} = 0.05\ s$.

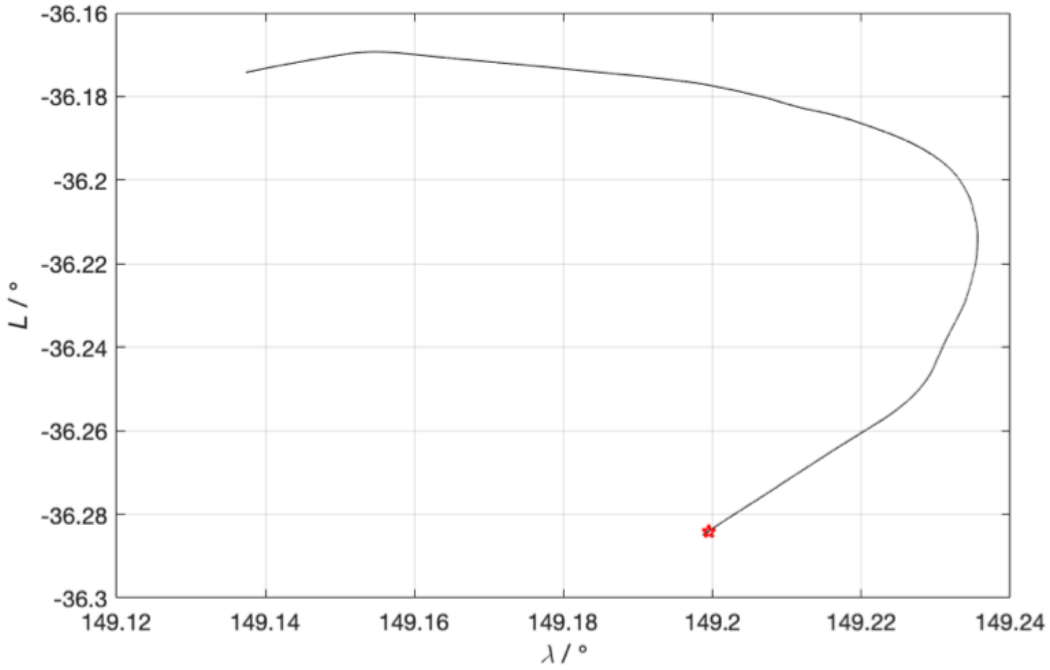

Figure 5.3: Approximated trajectory

Figure 5.3 shows the approximated trajectory of the vehicle in motion. As expected, the algorithm is able to evaluate the position evolution with the standard results that have been shown in the previous research. The position and velocity errors, whose evolution is shown in figures 5.4 and 5.5, are indeed coherent to the standard errors expected using a simple tightly-coupled integrated system.

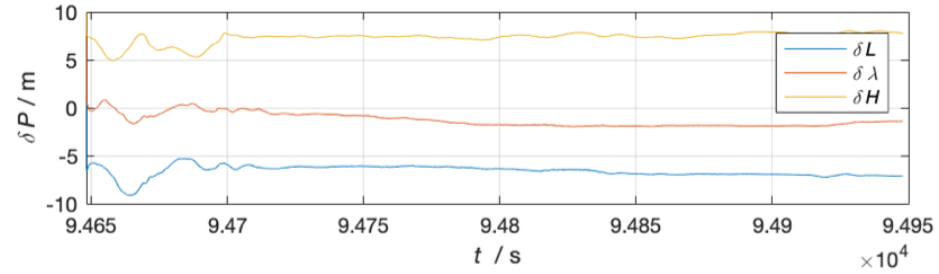

Figure 5.4: Position error evolution

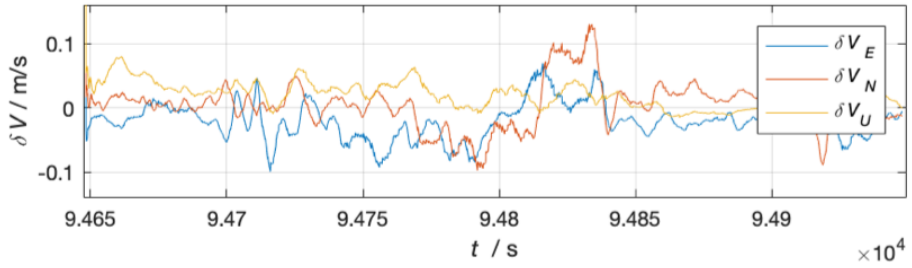

Figure 5.5: Velocity error evolution

5.3 Tightly-coupled GNSS/INS integrated system with RBNN

With the Radial Basis Function Neural Network, the system model reduces to the one shown in section 4.1. Furthermore the algorithm adopted for this work is based on the offline of NN. In order to do a proper simulation additional data has been generated to train it. This data has been approximated with the following form:

$$[(f_{ib}, \omega_{ib}), (\Delta\hat{\rho}, \Delta\hat{\rho}')]^T$$

Where, as shown in section 4.2, the outputs of the IMU sensors are directly associated to the pseudo-ranges and pseudo-range rates differences $(\Delta\hat{\rho}, \Delta\hat{\rho}')$.

The process of association between these functions permits the realization of the system model shown in figure 4.3.

5.3.1 Offline training of the RBFNN

At a first stage, the algorithm adopted is based on the offline training of the NN. The RBFNN is characterized by its fast training and by the fact that it does not need too much samples to give good performances. The NN has been nevertheless trained with 50 samples of pairs, which correspond to 4.9 s of additional trajectory data. The simulation is moreover runned supposing that the GNSS suffers of about 30 s of outage. This outage starts at about 79 s from the beginning of the motion of the vehicle. The simulation lead to the results shown in figure 5.6 and 5.7.

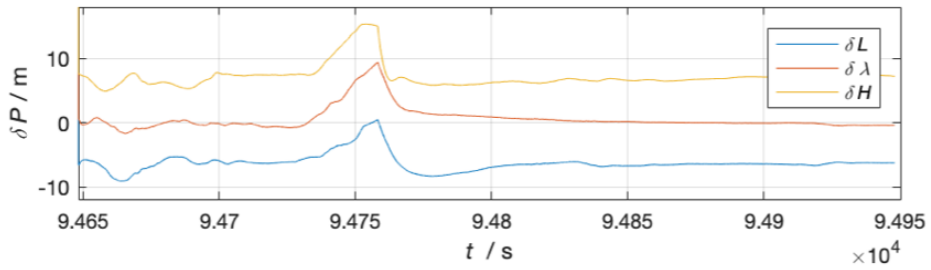

Figure 5.6: Position error evolution with offline training

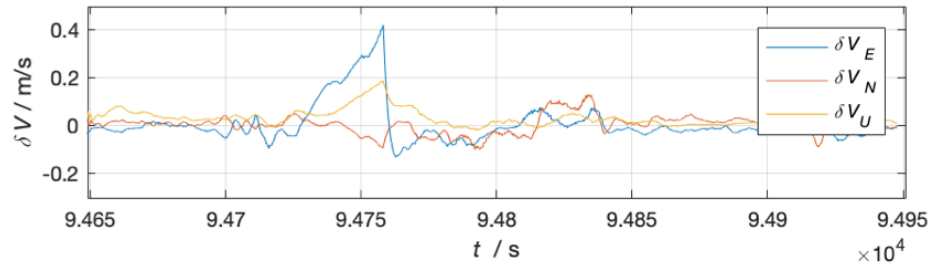

Figure 5.7: Velocity error evolution with offline training

As it is possible to observe, both of the error evolutions starts to grow

almost linearly at the beginning of the GNSS outage. Both of the evolutions present indeed a peak located exacty in the GNSS outage interval. This means that, as the GNSS stops to work and the system starts to work with its second generalized scheme (shown in section 4.1), the accuracy starts to slightly decrease. The performances of the position estimator starts to be normal when the GNSS starts to work properly again.

5.3.2 Online training of the RBFNN

In the second simulation stage the algorithm has been modified a bit in order to let it perform online training. Online training is based on the assumption that, as long as the GNSS is working properly, the RBFNN is contemporarily trained with the samples retrieved during the iterations of the algorithm. In order to obtain better results for the comparison, the GNSS is planned to have an outage at almost 79 s from the beginning of the motion of the vehicle, as done in the previous sub-section. The outage duration is 30 s. This time, of course no additional data is used to train the NN. As stated before the samples of pairs computed iteratively are indeed used to train the network.

In figures 5.8 and 5.9 it is possible to observe the same phenomena that occurred with offline training. The error evolution of both the position and the velocity present peaks that are caused by the lack of accuracy that becomes more ingent as long as the GNSS has an outage. The situation then restores to normality after the GNSS starts to work properly again.

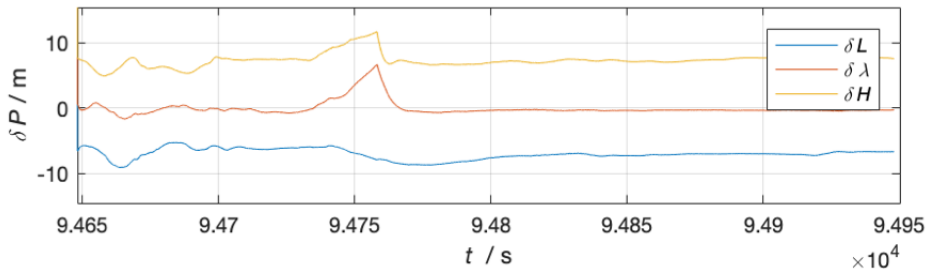

Figure 5.8: Position error evolution with online training

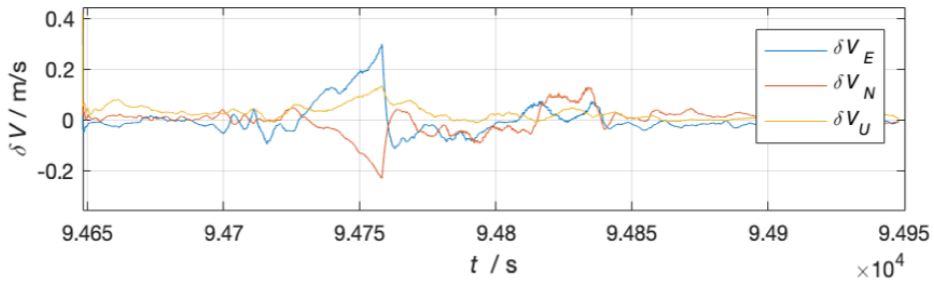

Figure 5.9: Velocity error evolution with online training

5.3.3 Comparison

Whether if the RBFNN is trained with an offline or an online approach, this work show that it is possible to implement successfully a NN in any tightly-coupled GNSS/INS integrated navigation system. Giving a more closer look to the error evolutions shown in the previous sub-sections, it is moreover clear that online training provides slightly more accurate position solution during GNSS outages. An offline trained RBFNN can produce less accurate approximations due to the fact that the error peak is slightly higher, as it is possible to observe comparing figures 5.8 and 5.9 with figures 5.6 and 5.7.

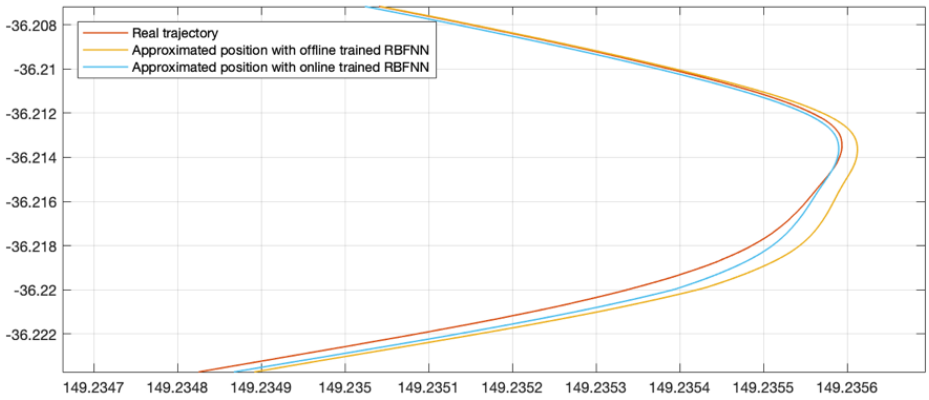

Figure 5.10: Trajectory comparison between the two different approaches

Figure 5.10 gives a direct proof of the these assumptions. It shows a zoomed overview of the trajectory of the vehicle during the GNSS outage. It is clearly possible to observe that the online trained RBFNN provides a GNSS/INS integrated solution much more closer to the real trajectory of the vehicle in motion. The offline trained RBFNN on the other hand provides a less accurate navigation solution, which is more far from the real trajectory.

Chapter 6

Conclusions

Offline training of RBFNN can be a very powerful approach. It let the integrated system be more efficient and fast, because the training of the NN is done una tantum and in a supervised approach. A drawback of this training procedure is that the supervisor should be able to generate as much eterogeneous data as possible, in order to let the RBFNN be ready to respond to many different kinds of inputs. Data can be generated artificially, providing the same INS error modeling, or can be collected directly from the vehicle motion.

Online training on the other hand gives much more accurate solutions because the learning of the RBFNN is done contemporarily to the GNSS/INS standard working routine (when GNSS is also working properly). During the training the NN is being fed with more eterogeneous data and is made more efficient and ready to give accurate outputs to many kinds of different input pairs. One drawback of online training is that is done in real-time. This may cause computational problems because at each iteration of the algorithm the NN must be trained and updated.

6.1 Future research

This work proposes one possible approach to the mitigation of the GNSS outages in a GNSS/INS tightly-coupled integrated navigation system. In order to further analyze and study how to exploit in the best way NNs, it can be interesting to further deepen some other aspects that may be helpful to this research. For example:

- Analyzing and comparing the performances of a GNSS/INS integrated navigation system with RBFNN aiding using both a loosely-coupled and a tightly-coupled approach.
- Comparing the performances on the tightly-coupled GNSS/INS integrated system using a standard NN.
- Analyzing the performances of the system shown in this work when the supervised offline trained data-set varies (for example using less eterogeneous data or increasing the pair samples).
- Analyzing the NN overfitting phenomena in the online training case.
- Analyzing and comparing the computational costs of the algorithms adopted in this research.

Bibliography

[1] MEMS and Nanotechnology Exchange. «What is MEMS Technology?; Online: https://www.mems-exchange.org/MEMS/what-is.html». In: () (cit. on p. 8).

[2] F. Dovis. «Slides from Satellite Navigation Systems; Basic Principles of Positioning». In: () (cit. on p. 9).

[3] ETSI. «Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking;» in: (2013) (cit. on p. 10).

[4] Inside GNSS. «GNSS IoT Positioning: From Conventional Sensors to a Cloud-Based Solution; Online: https://insidegnss.com/gnss-iot-positioning-from-conventional-sensors-to-a-cloud-based-solution/.» In: (2013) (cit. on p. 10).

[5] TechTalks. «What are Artificial Neural Networks (ANN)?» In: (2019) (cit. on p. 11).

[6] J.-Y. Tourneret J.-R. De Boer V. Calmettes and B. Lesot. «Outage mitigation for GNSS/MEMS navigation using Neural Networks». In: (2009) (cit. on p. 13).

[7] Novatel. «IMU Errors and Their Effects». In: () (cit. on p. 22).

[8] A. Wilson. «A Brief Introduction to Supervised Learning». In: () (cit. on p. 29).

[9] T. B. Karamat A. Noureldin and J. Georgy. «Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration». In: (2013) (cit. on p. 31).

[10] Earth New Mexico Tech and Environmental Science. «ERTH 455 / GEOP 555 - Geodetic Methods for Understanding Earth's Surface Deformation - Lecture 05: Position Estimation with Pseudoranges». In: () (cit. on p. 32).