Cortically Inspired Sensor Fusion for Portable Devices

eingereichte
PRACTICAL COURSE
von

Markus Steppberger
Hela Tajouri

Lehrstuhl für
STEUERUNGS- und REGELUNGSTECHNIK
Technische Universität München

Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

Betreuer: Cristian Axenie
Beginn: 14.04.2015
Abgabe: 07.07.2015

TECHNISCHE UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR STEUERUNGS- UND REGELUNGSTECHNIK
ORDINARIUS: UNIV.-PROF. DR.-ING./UNIV. TOKIO MARTIN BUSS

April 20, 2015

P R A C T I C A L   C O U R S E
for
Hela Tajouri, Mat.-Nr. 03661568
Markus Steppberger, Mat.-Nr. 03604841

Cortically Inspired Sensor Fusion for Portable Devices

Problem description:

Todays wearable and mobile devices incorporate various sensors to capture information about a users behaviour, movement and location. And all of these sensor signals and raw data need to be translated and processed into meaningful insights. Sensor fusion is the process of combining multiple sensory streams to provide a more precise estimate of a sensed feature than any of individual sensors. Yet, smart phones and tablets are far from ideal sensing platforms. Their manufacturers need to keep the product compact and inexpensive, which can compromise sensor reliability and increase coupling to the electronic environment they operate in (RF signals, electrical noise). In addition, portable devices are subjected to non-ideal environments such as magnetic anomalies, temperature variations, and shock and vibration. This adds noise to the sensor measurements. Although various sensor fusion mechanisms were developed to cope with these problems in mobile sensing, they are dedicated to the hardware setup, lack flexibility and need tedious parametrization. Using a recently developed cortically inspired processing scheme for sensor fusion [1], the project aims to provide a mobile instantiation on an Android mobile device. The model is a distributed network in which independent neural computing nodes obtain and represent sensory information, while processing and exchanging exclusively local data, to infer an estimate of the device's orientation. In this practical project an orientation filter will be implemented using measurements from gyroscope, accelerometer and magnetometer in a smartphone / tablet, using the proposed neural model and the Android API [2].

Tasks:

- Get familiar with the neurally inspired sensor fusion network.
- Get familiar with the Android sensors API for sensory data acquisition.
- Design and implement the neural sensor fusion for orientation estimation on the Android system.
- Test and evaluate the application.

This practical requires programming in Java and knowledge of Android development.

Bibliography:
[1] Axenie C., Conradt J.: Cortically inspired sensor fusion network for mobile robot egomotion estimation. Robotics and Autonomous Systems (2014)
[2] Android sensors interface API, https://source.android.com/devices/sensors/sensor-stack.html

Supervisor: Cristian Axenie

(J. Conradt)
Professor

Abstract

Smartphones develop in a rapid speed in recent years, and they are helping and changing the way of our living. The sensors embedded in them provide various possibilities for mobile applications and these sensors are the key factors of developing more and more interesting smart-phone applications. Some applications require precise measurements from sensors to capture information about device movement and orientation. In this project we present a implementation of a cortically inspired sensor fusion model, which given various sensory inputs, relaxes into a solution which provides a good estimation of the orientation of a mobile device.

Contents

Chapter 1

Introduction

Smartphones and tablets are far from ideal sensing platforms. The sensors capabilities are compromised by commercial constraints such as size and low price. The sensors need to be small, inexpensive and consume low power. Theses sensors being subjects to non-ideal environment, they cannot provide accurate data due to noise, interferences and systematic errors. For the requirements of more accurate and more precise of sensory data, sensor fusion is introduced to improve the accuracy and provide accurate, complete and reliable data. Sensor fusion,is the process of combining sensory data from disparate sources such that the resulting information is in some sense better than would be possible when these sources were used individually[**?**].The combination process allows to balance contributions from disparate sources, resulting in a global consistent representation. In this project, we present an orientation filter for mobile using measurements from gyroscope, accelerometer and magnetometer follows a bio inspired approach to minimize the drift and noise in input orientation. This report is organized as follows. The next section will give a review about fusion algorithms for Android devices. We introduce the model in section 2 and finally, the results are shown in section 3

1.1 Problem Statement

In this project, we follow a bio inspired approach for sensor fusion, proposed in[**?**]. This paper proposes a cortically inspired approach for heading estimation and also presents an implementation of a four dimensional sensor fusion scenario on an autonomous mobile robot and demonstrate its performance using the same framework we instantiate a network for mobile devices.

1.2 Related Work

In[**?**]the design of multi-sensor attitude determination systems is discussed using quaternion and Euler based algorithms.[**?**] introduces a Direction Cosine Matrix

(DCM) based method for attitude and orientation estimation. DCM method has some advantages over the popular methods such as Euler Angle, Quaternion in light of reliability, accuracy and computational efforts.The still limitation of this method is that measurement noise covariance of magnetic compass could not be accessed directly. An extended Kalman filter (EKF)-based data fusion algorithm is discussed in[**?**]. A systematic approach based on wavelet decomposition is utilized to estimate noise covariances used in the Kalman filter formulation. All these approaches provide good results in fixed scenarios bud need judicious parametrization. Furthermore, in order to handle noise and uncertainty, special core must be taken.

Chapter 2

Neural model for orientation estimation on mobile devices

2.1 Android sensors

Most Android devices have sensors that measure motion, orientation, and various environmental conditions. The android platform provides two hardware-based sensors for motion estimation: accelerometer and gyroscope and two hardware-based sensors for position estimation:proximity sensor and magnetometer.

Accelerometer

The accelerometer measures proper acceleration, which is the acceleration it experiences relative to free fall and is the acceleration felt by people and objects. The gravity force affect the measurement of accelerometer for measuring speed or displacement of an object in three-dimensions. The gravity force must be subtracted before any measurement. However, the gravity force can be taken as an advantage of detecting the rotation of a device. When a user rotates his smart-phone, the content he is watching will switch between portrait and landscape. When the screen of smart-phone is in a portrait condition, y-axis will sense the gravity; when the screen of smartphone is in a landscape condition, x-axis will sense the gravity. According to this, users can rotate their screens without affecting their reading experiences.

Gyroscope

The gyroscope is a primary source of extracting orientation.The gyroscope measures the angular rate of the body with respect to body frame coordinates.It can therefore be used to derive orientation by integrating over time if initial orientation is known. All Orientation determination systems that use gyros suffer from bias or drift, which is a low frequency noise component.[**?**] The heading calculated from the magnetometer is not accurate. Moreover, roll and pitch calculated from accelerometer sensor are only accurate when mobile is stationary or its acceleration is zero.[?]

Magnetometer

The magnetometer measures the strength and the direction of magnetic fields. The accelerometer and the gyroscope are able to detect the direction of a movement which is a relative and obeys the coordinate system that a smart-phone uses. Therefore, a magnetometer is needed to get an absolute direction (the direction that obeys the coordinate system of earth). The main source of measurement errors are magnetic interference in the surrounding environment and in the device.

2.2 Dynamics of the network

The architecture of the model is based on the model given in [?] . The network contains nodes which mutually influence each other. Values in each node take small steps towards minimizing the mismatch with the values given by the other nodes. The update rule for node i is given by :

$$m_i(t+1) = m_i(t) - \eta(t).E_{m_i,m_j}(t)$$

$$E_{m_i,m_j} = m_i(t) - m_j(t)$$

These steps are controlled by a confidence factor that quantifies the mismatch of mi from the network and modulates the influence of each unit

$$m_i$$

.The confidence factor for a node i when receiving influence from a node j is given by

$$\eta_{i,j}(t+1) = \eta_{i,j}(0)\frac{\sum_{k=1}^{N-i} E_{m_i,m_k}}{N.E_{m_i,m_j}(t)}$$

The division emphasize the mismatch between the value given by the feeding unit and the network belief.

2.3 Network structure

As mentioned in the section Android sensors, the sensors do not provide accurate values for all the angles. In our proposed approach, the gyroscope and the magnetometer are used to estimate the yaw angle. The accelerometer and the gyroscope are used to estimate the roll and pitch angles.
The fusion process described in section 2 is implemented for each angle .The raw sensor data is fed to each node in the network and then each node implements the identity relationship with the other concerned node and the sensory input.
The implementation of the model in our scenario is described in figure **??**.

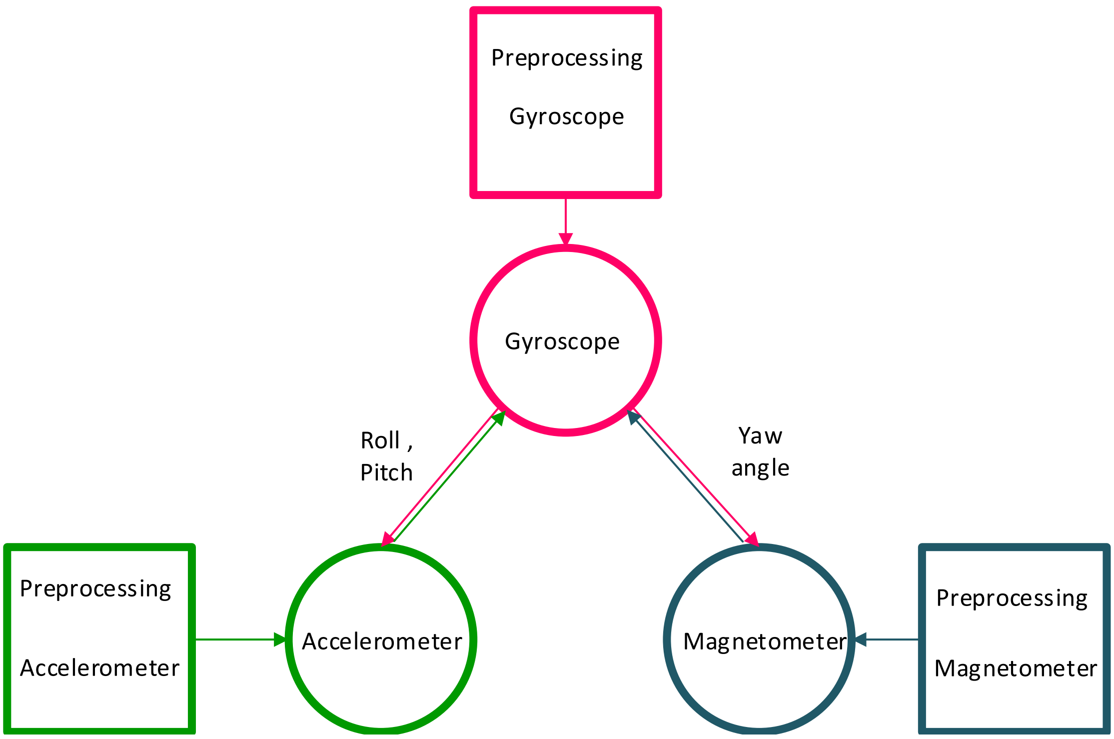

Figure 2.1: Orientation Network Architecture

Chapter 3

Model implementation and evaluation on Android devices

3.1 Android implementation

3.1.1 Data Acquisition

Raw sensor data is obtained by using the Android Sensor API[1]. The raw data acquisition is mainly done by the SensorReader class, which implements the SensorEventListener interface[2]. The class provides a function that is called when sensor values have changed. By reading out the changed values from a SensorEvent[3] the network gets new data whenever the device is moved. The raw data is given for the accelerometer in m/s^2, for the gyroscope in rad/s and for the magnetic field sensor in μT. Therefore the data has to be preprocessed in order to get the same unit for all values. In this case the Euler angles are used as the input unit for the sensor fusion network. The preprocessing is done in the SensorReader class before the values are used for the network computation.
As the Android application provides two different modes, the PrecisionMode and the NormalMode, there are two different implementations for the preprocessing of the raw data. The PrecisionMode aims to increase the precision of the orientation acquired from the Androi API itself. The Android API uses the accelerometer and the magnetic field sensor to provide a rotation matrix from which the Euler angles can easily be computed. The sensor fusion network, explained in this report, increases the quality of the data by also using the data from the gyroscope. The preprocessing in this mode is done by integrating the gyroscope data over time in order to get angle values in degrees.
The NormalMode uses the Android API values acquired from the accelerometer and the megnetic field sensor only for the yaw orientation. The angles for pitch and roll

[1]https://developer.android.com/reference/android/hardware/Sensor.html
[2]https://developer.android.com/reference/android/hardware/SensorEventListener.html
[3]https://developer.android.com/reference/android/hardware/SensorEvent.html

are taken from the preprocessed data from the gyroscope, as descriped above, and the preprocessed data from the accelerometer. The raw data from the accelerometer is preprocessed by computing simple formulas that provide angles out of acceleration values (m/s^2).

3.1.2 Evaluation

The results were compared to the rotation matrix provided by the sensor API. Figure **??** shows the sensor fusion output values, values given by android rotation matrix and the relative error for values for pitch, roll and yaw angles. In order to

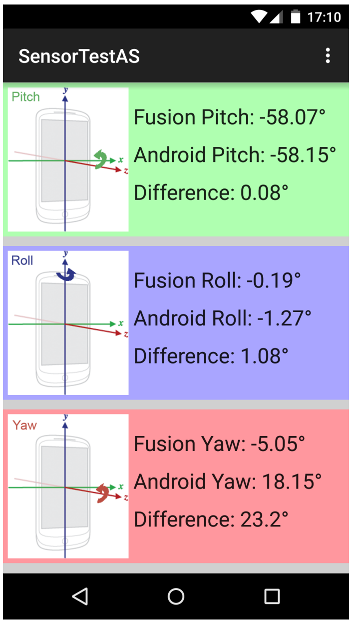

Figure 3.1: GUI of the android application

measure the performance of our model, we compare the output with raw sensor data obtained using the Android API. The model output is much smoother as compared to orientation calculated from the sensors. For the evaluation the device perfomed a 90° roll movement, starting from a flat position on the table. The steplike structure of the graphs is based on the fact that the fusion network computation is triggered independently from the change rate of the sensor values. Therefore some values might not have changed, while others have.
Figures **??**, **??** and **??** show the output angles of the network and the used raw data compared to each other. All shown data samples were recorded at the same time. It can be seen that the network values (green lines) are much smoother than the raw

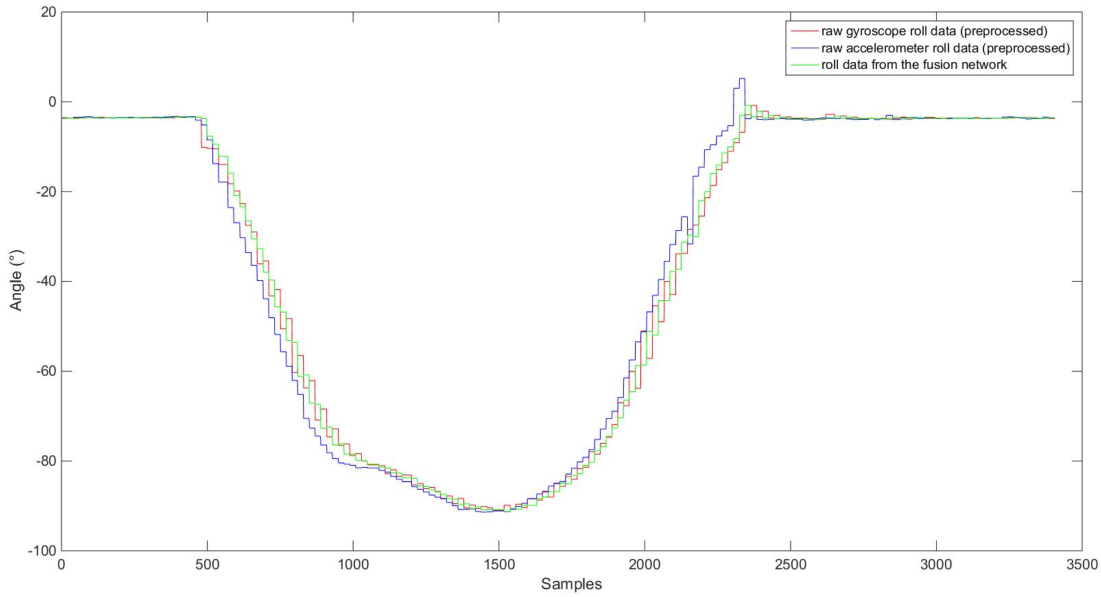

Figure 3.2: Fused and raw data for the roll angle while performing a 90° roll movement

data. Between samples 2000 and 2500 in figure **??** the network flattens the peaks to nearly half the values that were computed from the accelerometer alone. The plots show that the network pulls the represented data from each sensor towards a common value and eliminates distorsions caused by eventual noise and errors.
Figures **??**, **??** and **??** were created out of the data which is shown on the GUI of the application. The blue line represents the angle value from the android api, the green line shows the output of the network and the red line shows the difference between these values. It can be seen that the difference is highest when the device leaves the 0° roll position and when it comes back to 0°. As figure **??** shows around sample 2450, the android api overshoots a bit but the network flattens this behaviour. This leads to the fact, that the sensor fusion network used in this course can increase the precision of the sensor data. The same increased qualitiy of the data can be seen in figure **??** between samples 2000 and 2500. The performed movement was a 90° roll movement only, so there should be no pitch angle at all. The accelerometer gives values up to 7°, whereas the network values are much smaller. Figure **??** shows some fluctuations in the yaw angle, although there was no yaw movement. As can be seen in the figure the used sensors are very noisy and unstable in calculating yaw angles. Nevertheless the network shows a much smoother behaviour than the android api.

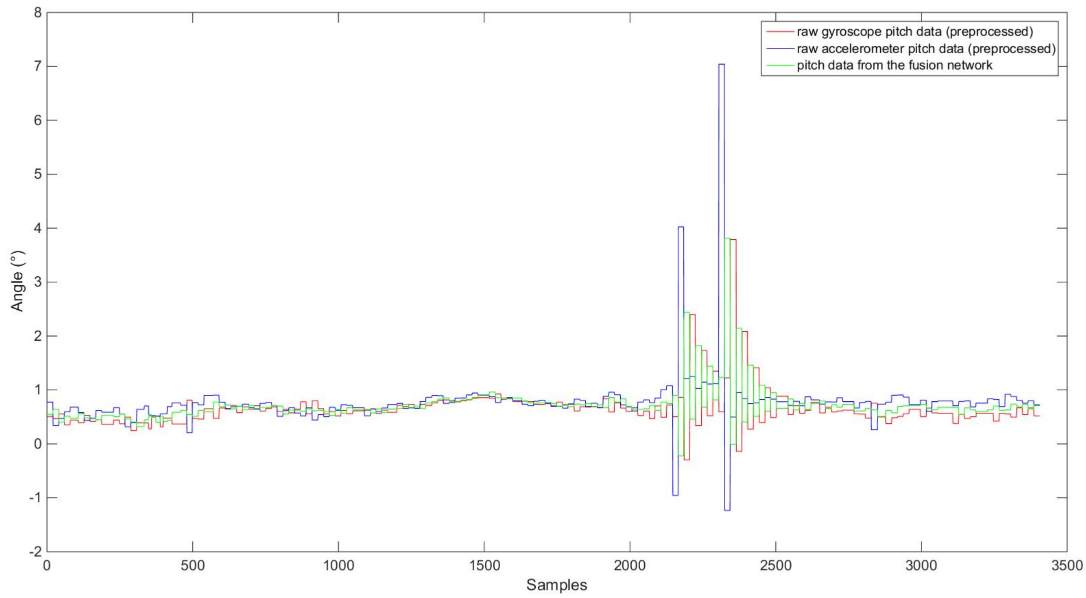

Figure 3.3: Fused and raw data for the pitch angle while performing a 90° roll movement

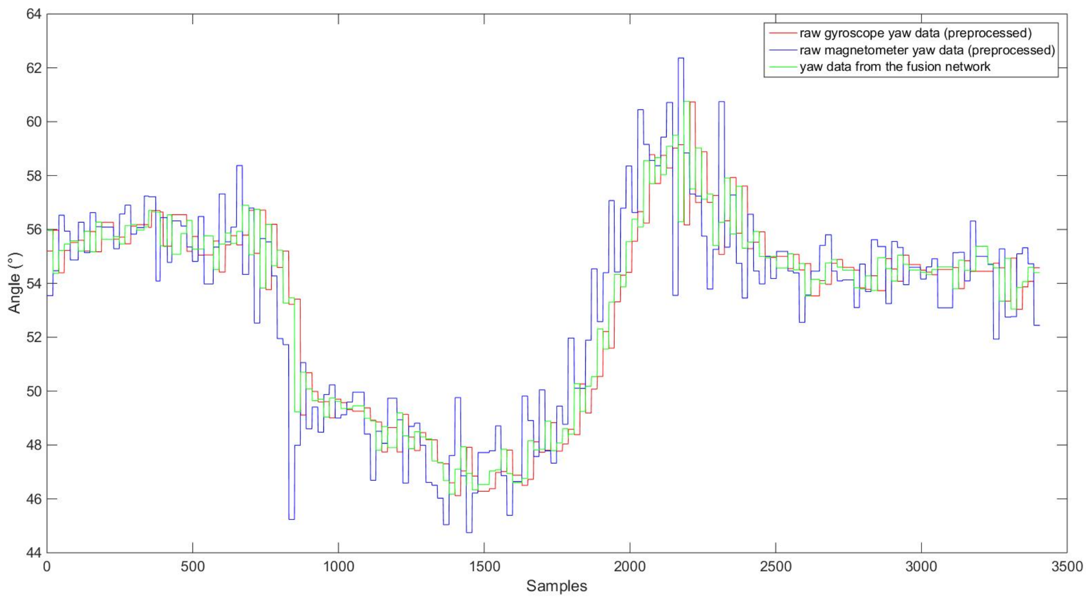

Figure 3.4: Fused and raw data for the yaw angle while performing a 90° roll movement

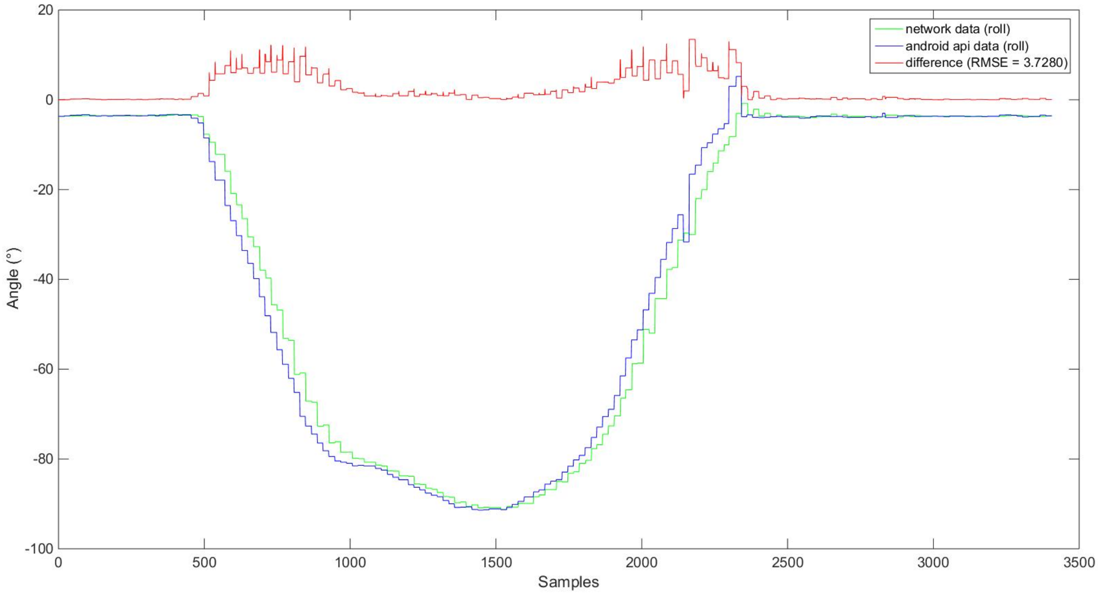

Figure 3.5: Fused data and Android API data for the roll angle while performing a 90° roll movement

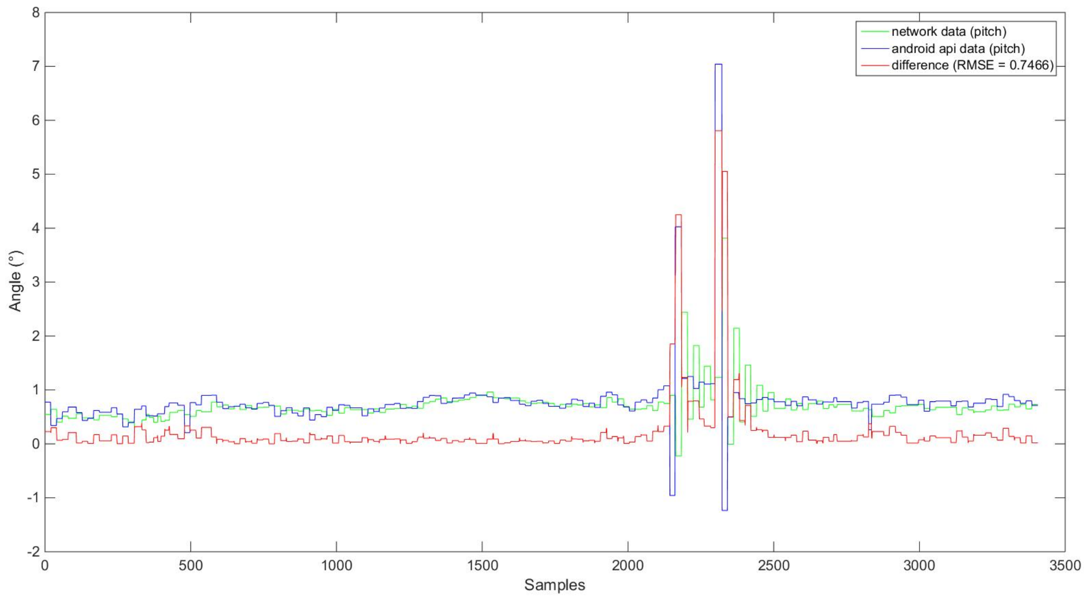

Figure 3.6: Fused data and Android API data for the pitch angle while performing a 90° roll movement

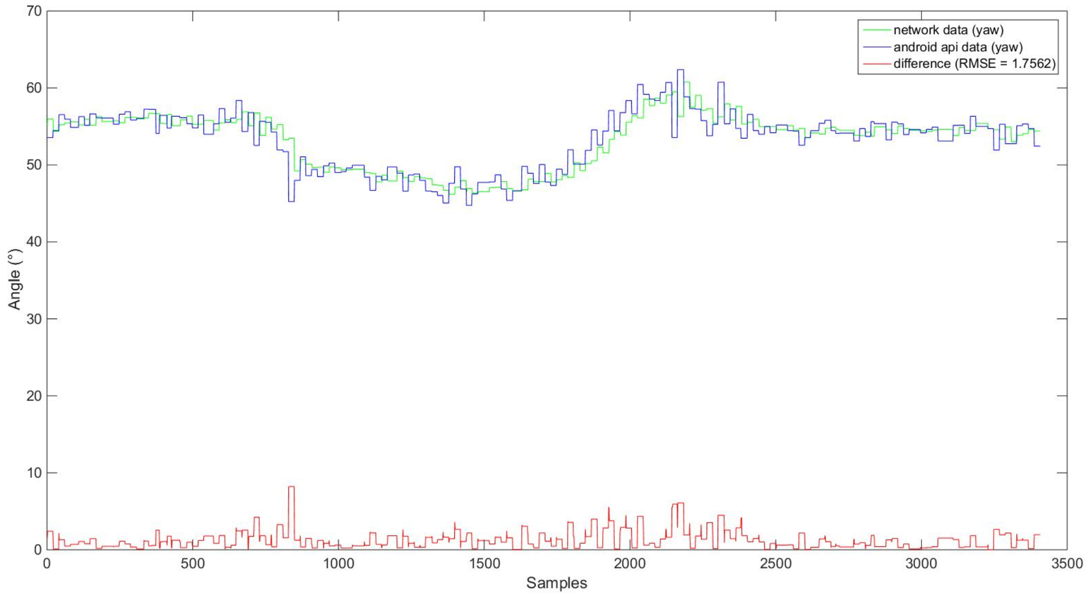

Figure 3.7: Fused data and Android API data for the yaw angle while performing a 90° roll movement

Chapter 4

Conclusion

In this work an novel and accurate orientation estimation method is described and implemented using a cortically inspired sensor fusion algorithm. The model uses a network in which nodes obtain and exchange data, to obtain an estimate of mobile orientation using data from accelerometer, magnetometer and gyroscope in a mobile device.

The still limitation of this method is that the model cannot handle fast variation of orientation.The network needs time to relax to accommodate incoming samples. However it gives another advantage is that the algorithm can good estimation for orientation without any parametrization and with less computation complexity compared to state-of-the-art algorithms .

List of Figures

Bibliography

[1] J. D. Powell D. E. Gebre-Egziabher and R. C. Hayward. Design of multi-sensor attitude determination systems. IEEE Transaction on Aerospace and Electronics Systems, 40(2):627649, 2004.

[2] Cristian Axenie, Jorg Conradt. Cortically inspired sensor fusion network for mobile robot egomotion estimation. Robotics and Autonomous Systems, 71:69-82, 2015.

[3] Hee-Jun Kang Y.-S. S. Nguyen Ho Quoc Phuong and Y.-S. Ro. A dcm based orientation estimation algorithm with an inertial measurement unit and a magnetic compass. Journal of Universal Computer Science, 15(4):859-876, Feb 2009.

[4] B.Honary S.Ayub, A.Bahraminisaab. A Sensor Fusion Method for Smart phone Orientation Estimation. 13th Annual Post Graduate Symposium on the Convergence of Telecommunication, Networking and Broadcasting, 2000.

[5] A. Khayatian P. Setoodeh and E. Farjah. Attitude estimation by separate-bias kalman filter-based data fusion. The Journal of Navigation, The Royal Institute of Navigation, 57:261-273, 2004.

[6] P. D. Groves. Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems. Artech House BostonLondon, 2008.

[7] J. Conradt, P. Simon, M. Pescatore, and PFMJ Verschure. Saliency Maps Operating on Stereo Images Detect Landmarks and their Distance, Int. Conference on Artificial Neural Networks (ICANN2002), p. 795-800, Madrid, Spain, 2008.

[8] C. Denk, F. Llobet-Blandino, F. Galluppi, LA. Plana, S. Furber, and J. Conradt. Real-Time Interface Board for Closed-Loop Robotic Tasks on the SpiNNaker Neural Computing System, International Conf. on Artificial Neural Networks (ICANN), p. 467-74, Sofia, Bulgaria (2013).