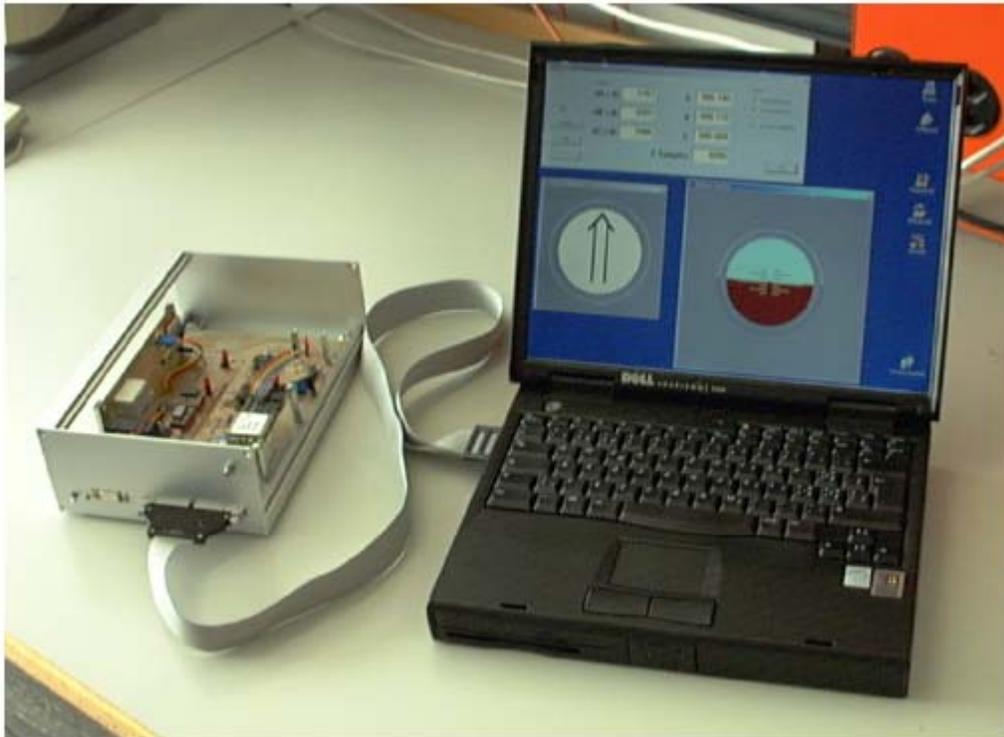


Translator's Notes:

On German page 28, lines 9-10 of section 6.5, the sentence beginning with 'Hier, unsere... [Here, our ...]' is incomplete. It was left as is in the translation (English page 38, line 6 from the bottom).

The sentence beginning with "Eine Verbesserung der Auflösung... [Improving the resolution...]" on German page 38, line 7 of the actual text below the heading of section 8 (English page 54, line 7 of the actual text in section 8) is missing one or more words.

Low Cost Inertial Navigation System



Authors: Peter Lüthi, Thomas Moser

Advisors: Markus Uster, Teddy Loeliger

Professor: Prof. Dr. H. Jäckel

Term Paper SS2000

Table of Contents

1	Abstract	6
2	Introduction	7
2.1	What is Inertial Navigation?.....	7
2.2	Evaluation of Our Preceding Project.....	9
2.2.1	Overview.....	9
2.2.2	Characterization of the Preceding Project.....	10
2.2.3	Programming Environment.....	10
2.2.4	G-Meter.....	11
2.2.5	Conclusion.....	11
3	Design	13
3.1	Overview.....	13
3.2	Hardware.....	13
3.3	Software.....	14
4	Implementation I: Test Board	16
4.1	Overview.....	16
4.2	Design of the Test Board.....	16
4.2.1	Sensors.....	17
4.2.2	Filters.....	18
4.3	Software.....	20
4.3.1	National Instruments DAQ-Card 1200.....	20
4.3.2	Sampling Parameter.....	20
4.3.3	The ‘MajorScan’ Program.....	21
4.4	Measurements of the Sensors.....	21
4.4.1	Evaluation of the Sensors.....	22
4.4.2	Sensor Model.....	22
4.4.3	Temperature Measurements.....	22
4.4.4	Acceleration Measurements.....	23
4.4.5	Algorithms.....	23
4.4.6	Filter.....	24
5	Implementation II: Sensor Board	25
5.1	Overview.....	25
5.2	The Hardware of the Sensor Board.....	25
5.2.1	Power Supply.....	25
5.2.2	GPS & MAX232.....	26
5.2.3	Temperature Compensation.....	27
5.2.4	Filter & Amplifier.....	28
5.3	Software for the Sensor Board.....	29
5.3.1	‘Cockpit’ Demo Program.....	30

5.3.2	'Espresso' Navigation Program.....	31
5.3.3	Recording the Data.....	32
6	Implementation III: Algorithms	33
6.1	Representation in Space.....	33
6.1.1.	Translatory Coordinates.....	33
6.1.2	Rotating Coordinates.....	34
6.2	Transformation of Measured Values in Our Coordinate System....	35
6.3	Regulation Process/Feedback.....	36
6.4	Initialization.....	37
6.5	Temperature Compensation.....	38
7	Results	41
7.1	Gauging the Finished System.....	41
7.1.1	Electrical Verification.....	41
7.1.2	Oscillating Filter.....	42
7.1.3	Power Supply.....	42
7.1.4	Positional Measurements.....	43
7.1.5.	Static Positional Measurement.....	43
7.1.6	Dynamic Positional Measurement.....	44
7.1.7	Static Positional Measurement.....	44
7.2	Findings.....	45
7.2.1	Power Supply.....	45
7.2.2	Temperature Compensation.....	45
7.2.3	Righting Mechanism.....	46
7.2.4	Initialization.....	46
7.2.5	Heading Representation.....	48
7.2.6	Processing Power.....	49
7.3	Sources of Inaccuracies.....	49
7.3.1	Limited Resolution.....	49
7.3.2	Bias and Sensitivity Drift/Nonlinearities.....	50
7.3.3	Alignment Errors.....	51
7.3.4	Interpolation and Quantization Errors.....	51
7.3.5	Processing Errors Based on Excessive Angular Changes...	52
7.3.6	Accumulation of Errors.....	53
8	Results	54
9	Outlook	55
A	'Espresso' Software Structure	57
B	Diagram of Hardware	59

List of Figures

1	The 6 Degrees of Freedom of the Space.....	7
2	The New Design with Laptop, PCMCIA Measuring Card and Sensor Board...	14
3	Acceleration Sensor ADXL210.....	18
4	ENC03 Gyroscope.....	18
5	ENV05 Gyroscope.....	19
6	The Test Board with Components that Can be Plugged In	19
7	The Sensor Board with the Flow Channel and the Fan.....	27
8	The Sensor Board with the Individual Sensors.....	30
9	Representation of Position in Space by Means of Euler Angle.....	35
10	The Finished System with Laptop, Sensor Board and PCMCIA Measuring Card.....	41
11	Software Structure of the 'Espresso' Program.....	58

Table Directory

1	Filter and Amplifier of the Test Board.....	19
2	Filter and Amplifier of the Sensor Board.....	28
3	Voltage and Current Measurements on the Sensor Board.....	42
4	Drift of Unregulated Positional Measurement.....	43
5	Drift of Positional Measurement (Regulated Position).....	45

1 Abstract

The inertial navigation is based on techniques that have been developed after World War II. The first systems were completely mechanical, large, and consequently technologically intensive and expensive. Then, 'solid-state' approaches occurred, as they are now used in civilian flying for primary navigation. The latter are always still fairly expensive and large. Completely integrated acceleration and rotational speed sensors, which are small and primarily very advantageous, have existed for a short time. The question here is now whether one can go a step further with these new sensors and can expand the area of use of the inertial navigation by miniaturization of an inertial navigation system with the aid of such sensors. There are adequate applications in the areas of mobile robotics, wearable computing, automobile and consumer electronics.

Our project had the object of clarifying how accurate such a platform with the currently available sensors can be and where the sources of inaccuracies lie. For this purpose, we have designed a prototype and developed the algorithms that are necessary in this respect. By the subsequent measurements, we could demonstrate that currently, the accuracy of the system that is achieved is for the most part owed to the limited resolution of the acceleration and rotational speed sensors.

2 Introduction

2.1 What is Inertial Navigation?

Inertial navigation platforms are also referred to with the term *INS*, whereby *INS* stands for *Inertial Navigation System*. Here, inertia is the item in question, since the actual acceleration can be measured by means of a mass or the inertia thereof. If the previous acceleration is known, the speed can be calculated by integration with the speed, in turn, of the path covered. When it is assumed that the original position was known, the current position can also be calculated by the path covered. It is important, however, not only to know the acceleration itself, but also the direction thereof. To this end, so-called gyroscopes, i.e., gyros, are used.

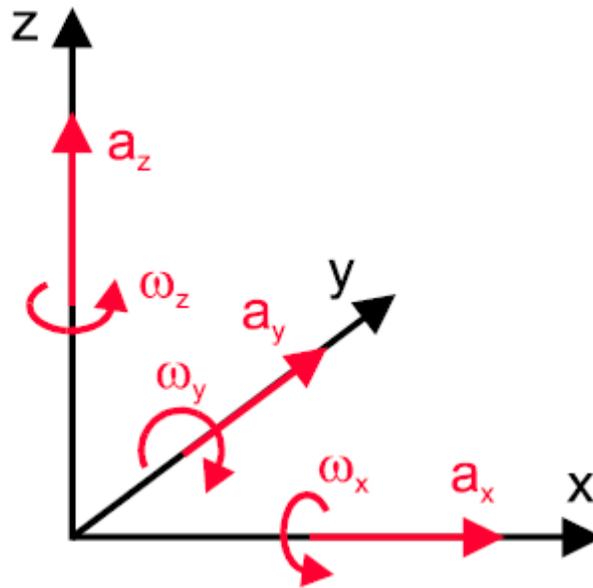


Figure 1: The 6 Degrees of Freedom in Space

$$s(t) = \iint a(t) dt^2 \quad \varphi(t) = \int \omega(t) dt$$

Inertial navigation was developed after World War II, e.g., up until now, all space travel programs have used this type of navigation; it is thus a principle [1] that has been known for a longer time. Based on the formulas, it is recognized that very precise measurements and calculations have to be made to obtain plausible results: an error in acceleration is double-integrated and thus very quickly results in corrupt data! To minimize the processing time, the acceleration meter is placed at the starting time upon a Cartesian-bearing, gyro-stabilized platform. Thus, all sensors always point in the same direction in each case regardless of how the vehicle moves. This solution, however, calls for a complicated, highly precise, and thus error-susceptible mechanics, which accordingly turns out to be expensive. Also, the space required for this purpose and the large amount of energy required cannot be disregarded.

With the emergence of more powerful computers and non-mechanical gyros (e.g., laser gyros), the design of so-called *Strap Down* INS platforms was then possible in the mid-1980's. In this connection, acceleration and rotational sensors are required. These sensors are now no longer suspended freely movable in position (Cartesian suspension) but rather are rigidly connected to the structure of the vehicle. The gyros no longer have the object of allowing the acceleration sensors to always point in the same direction, but rather they have to measure the angular changes. As a result, it can be determined in what direction the acceleration sensors are oriented. The measured accelerations and rotations are correspondingly converted, i.e., transformed into the reference coordinate system and only then integrated. Thus, the current speed and position can then be derived [2]. However, such INS platforms are also still very expensive and are therefore

used only in special environments, thus, e.g., in commercial aviation, in weapon systems, and in space travel.

In the last couple of years, new, very reasonably priced sensors have come on the market. These are sensors that are based on the so-called Micro Electro-Mechanical System (MEMS) technology. In this case, fine silicon structures that can convert mechanical loads or movements into electrical signals are used. With this technology, very small sensors with standard chip housings, as they are used in electronics, can be produced. The development of these sensors was promoted primarily by the automobile industry (airbag systems) and the automation and entertainment electronics industry (image stabilization in video cameras). The fact that these sensors (at this time) cannot exhibit the same accuracy as conventional sensors is evident, but it would be advantageous to know how accurate these sensors are in connection with a Strap-Down INS platform. The advantages of such a system are the low costs and the small dimensions, so that completely new applications can be developed. Such systems were conceivable as a way to supplement a GPS receiver to span short-term satellite reception gaps or for mobile robots in enclosed spaces.

2.2 Evaluation of Our Preceding Project

2.2.1 Overview

As a first object, we had put into operation our preceding project, the G-meter from the term paper of Marcel Lattmann [3]. Initial measurements should then be made with this device to find out where improvements are to be provided and what can be included for our project. In addition, we had new sensor types (ADXL210, ENC03 and

ENV0) available, whose possible integration into the existing project should be examined.

2.2.2 Characterization of the Preceding Project

This is the term paper of Marcel Lattmann in the summer semester of 1999 [3]. The device consists of a manageable aluminum housing with graphic LCD and four control buttons. A large, internal battery pack provides for an operating time of several hours. The processor board with a Motorola 68332 microcontroller, which had been developed by A. Stiller in the winter semester of 1995/96 as part of his thesis [4], takes over the functions of measurement and data processing. The G-measuring device has three acceleration meters with a range of +/- 50 g, but no gyroscope for rotational measurement. For evaluation, we first installed the software development environment and put the G-meter into operation. Then, the software of the G-meter was examined more precisely, and in the end the device was calibrated.

2.2.3 Programming Environment

A HICROSS programming environment was available to us. The installation itself was already causing problems and difficult to document in this respect. Data were provided on the server in false indices, not present, or with false attributes. Finally, we copied all data into a local index, newly sorted and matched to the configuration file. Also, the Makefiles had to be revised. After two small changes compiled in source code, the software and also the debugger/downloader could be started.

2.2.4 G-Meter

Before the binary file was downloaded, the initialization vectors for the processor still had to be matched. Now, the program could finally be written in RAM and started. Unfortunately, the LCD driver still has defects; the G-meter had to be rebooted several times until the LCD had been correctly initialized and did not show only stripes. Before acceleration can be measured, the device has to be calibrated via menu. However, the indicated values are not exactly consistent. The measurement can now be loaded on the PC via RS-232 and can be visualized in a Visual Basic application that is supplied. It has proven impossible to operate the stand-alone device since nothing could be written to the flash memory. After extensive tests, we were able to write data to the flash, but only from 0 to 800 Hex. Only FF lasted over 800 hours, although the programmer claimed that he had written to this area and did not issue an error message. After we had examined several reports from completed semester work and theses and had talked with the students who had previously dealt with the same processor board, we found out that up until now, no one has been able to write to the flash. In addition, all those involved in the programming environment gave very low scores.

2.2.5 Conclusion

After we had analyzed the preceding project for 2.5 weeks, we made the discovery that this project could not be implemented in the time available to us with the specified hardware. On the one hand, the expense for the production of the inertial navigation system on the MC68332 processor board was much too high and the entire system was too inflexible; on the other hand, it was clear to us that of the existing G-

meters, only a few could be included. The purpose of our project was primarily to find out how accurate an inertial navigation system based on low-cost sensors is, where the sources for the inaccuracies lie, and how much processing power is required. We would therefore like to analyze, with sufficient resources, the accuracy of the system in a simple and quick way. Only thus can it be clarified, during our short project period, what accuracy such a device can actually achieve and with what trade-offs a reasonable accuracy can be produced. Therefore, we are resolved to develop a new design.

3 Design

3.1 Overview

To optimally combine the various requirements such as portability, processing power, memory capacity and quick implementation, we developed the following design:

- As a computer, a PIII 500 MHz laptop with Windows98
- A PCMCIA I/O measuring card for reading the analog sensor values
- A separate sensor board with analog electronics and a GPS receiver

3.2 Hardware

As a laptop, a Dell Inspiron 7500 with a 500 MHz processor is used. It has sufficient processing power and memory space and in addition is able to take over the function of supplying the complete sensor board via the measuring card.

A DAQ-Card 1200 from National Instruments is used as an I/O measuring card. It has eight analog input channels and two analog output channels with a resolution of 12 bits, and 24 digital inputs or outputs. In addition, there are still three inputs to the pulse width measurement with 16 bit resolution. Up to 100,000 measuring points can be recorded per second. The intermediate storage of the measured data that is indispensable for a smooth interaction of the measuring card with the operating system takes over the function of a 2 kB RAM on the card itself. Thus, short-term gaps can be tolerated by the operating system without loss of measured data.

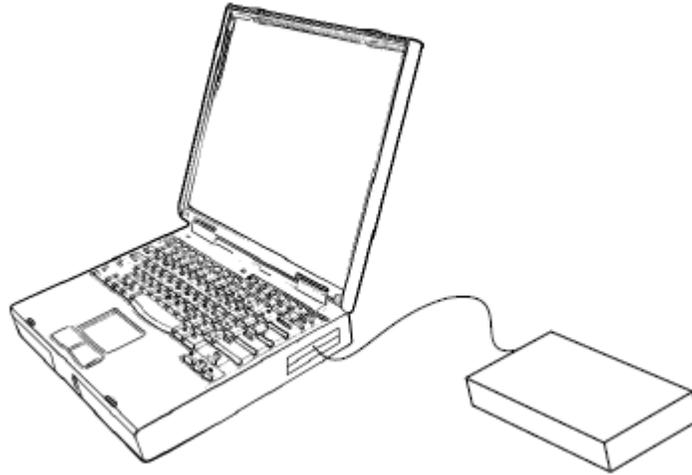


Figure 2: The New Design with Laptop, PCMCIA Measuring Cards and Sensor Board

In addition to the sensors and the hardware for the analog signal processing, a μ -blox GPS receiver is also to be installed on the sensor board. Before the analog/digital conversion, the signals first have to be low-pass-filtered to suppress aliasing and static. To fully use the 12-bit resolution of the DAQ-Card 1200, the signals are also further intensified.

3.3 Software

The software is to be written in Visual Basic, since the National Instruments Company, with its measuring card, includes, on the one hand, very good programming interfaces for this language and, on the other hand, can quickly execute a display of the results. The programming itself in the Visual Basic programming environment is comfortable and requires comparatively little time. The new design, however, based on the proportions of the laptop, represents a limited portable system; for our purposes, however, this factor is of secondary importance. The decisive advantage of a very high

flexibility is added to this. The sampling rates of up to 100 kHz are more than sufficient, and we can implement and test our application somewhat more quickly and store any number of data and, if necessary, export the latter into MATLAB.

4 Implementation I: Test Board

4.1 Overview

To build from scratch on proper analog electronics on the final board, we decided to produce a test board first. For each sensor type, the latter should have a slot with corresponding filters and amplifiers. In this connection, it was important that all components, such as resistors, condensers, etc., could be plugged in and could be quickly matched if necessary. The measurements should be performed with the DAQ-Card 1200, recorded on the laptop, and then evaluated. From the findings of this design, the final version of the inertial navigation platform should then be developed.

4.2 Design of the Test Board

The board had to be designed so that we could measure the following parameters:

- ENC03 output
- ENV05 output
- ADXL210 output (analog & PWM)
- ADXL250 output (analog)
- Reference voltage of AD780 (2.5 V)
- Temperature by means of AD780

To be able to exchange all sensors efficiently, we have produced small boards, which have a DIP14-wide as a footprint, for each sensor type. A DIP14-wide footprint consists of a DIP28 footprint with a 15 mm pin interval, shortened to 14 pins.

4.2.1 Sensors

The rotational sensors ENC03 and ENV05 are 'piezoelectric vibrating gyroscopes.' They measure a rotational speed via the Coriolis force, which acts on three small vibrating piezo plates [2]. To preclude mutual disruptions between two adjacent ENC03, there are types A and B with slightly different frequencies. An ENC03 costs approximately 50 SFr; an ENV05 costs approximately 200 SFr.

The acceleration sensors ADXL250 and ADXL210 of analog devices are integrated based on silicon. An elastically suspended mass is moved based on the acceleration; this shift is measured on a capacitive basis [2]. The ADXL250 and ADXL210 cost about 30 SFr.

The range of the A/D converter should be completely exploited for as accurate a measurement as possible. We have therefore produced range adaptations with amplifier circuits for all sensor outputs. The amplifier circuit transforms the sensor output signal from, e.g., 1.35 +/- 0.2 V optimally on the measuring input of 0-5 V of the DAQ-Card 1200. The working point of the output signal for the PCMCIA card has to be 2.5 V in this connection, and the signal amplitude also has a value of approximately 2.5 V. As a reference for all transformations, the reference voltage of AD780, which was determined at 2.5 V, is used.

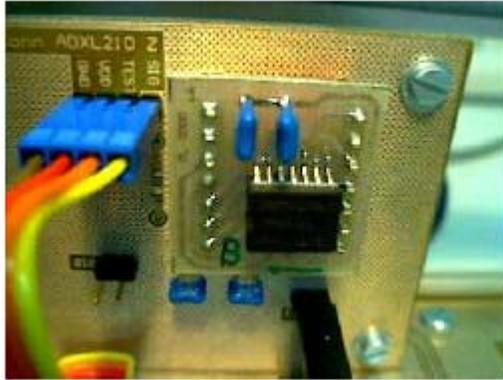


Figure 3: ADXL210 Acceleration Sensor

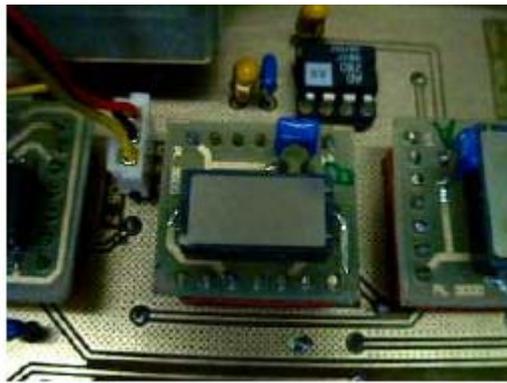


Figure 4: ENC03 Gyroscope

4.2.2 Filter

To avoid aliasing during measuring and to minimize static, we implemented all sensors of the active Second Order Butterworth Lowpass Filter at the output. The cut-off frequency of the individual filters and the amplification of the subsequent range adaptation of the sensor signal can be deduced from Table 1.

With ENC03, we have produced an AC coupling for temperature compensation as proposed in the application notes by Murata.



Figure 5: ENV05 Gyroscope

Sensor	Filter Fc (-3dB) [Hz]	Amplification A
ADXL210	50	2.2
ADXL250	50	5.6
ENC03	50	10
ENV05	25	1.2

Table 1: Filter & Amplifier of the Test Board

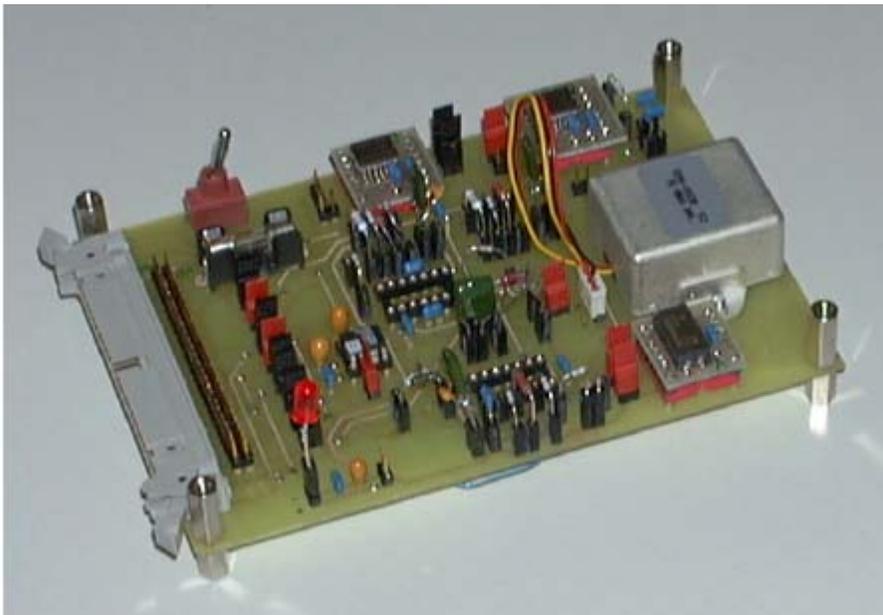


Figure 6: The Test Board with Components that Can be Plugged In

4.3 Software

4.3.1 National Instruments DAQ-Card 1200

The start-up of the DAQ-Card 1200 of National Instruments was configured in a simple manner. The incorporation in Visual Basic takes place either via ActiveX controls or via NI-DAQ functions. Since the existing ActiveX controls are only demo versions (costs of complete version: \$500) and are deactivated after 5 minutes, the NI-DAQ functions were used for Visual Basic. There are very good sample programs that illustrate the use of these functions, and the Online-Help is informative. However, the latter relates not only to the DAQ-Card 1200, but rather to all NI products. The DAQ-Card 1200 is part of the Lab series and thus, e.g., Lab_ISCAN_Start () has to be used and not, for example, SCAN.Start ().

It is measured continuously, since the exact time period is still not known in the beginning. In this respect, the double-buffered mode is used (see DAQ_DB_Config() in Online Help.) The DAQ-Card 1200 first fills one-half of the card buffer with data. This buffer can be read as soon as it is full. In the meantime, the second half of the buffer is described.

4.3.2 Sampling Parameters

The timing adjustments are somewhat awkward when setting the sampling. In this respect, it is recommended to study the criteria in the DAQ_DB_config () and Lab_ISCAN_Start () exactly. Here is an example of our code for a sampling frequency of 150 Hz:

```

‘ Calculate Timebase, SampleInterval and ScanInterval
‘ iStatus% = DAQ_Rate (S_Freq, iUnits%, S_sampTimebase, S_sampInterval)

```

```

‘ Conditions:
‘   - Time per measurement >= 10 us (with gain < 10) S_sampInterval/realTimebase
‘   - realTimebase  sampTimebase
‘       100 Hz      5
‘       1 kHz      4
‘       10 kHz     3
‘       100 kHz    2
‘       1 MHz     1
‘   - scanInterval < 65535, > 2 and > NrCh/realTimebase + 5 us
‘   - Sample Rate (Hz) = realTimebase/scanInterval

‘ For 150 Hz:
‘   S_sampTimebase = 1
‘   S_sampInterval = 10
‘   realTimebase = 1000000
‘   S_scanInterval = 6667

```

The $S_sampTimebase$ of the on-board timer is thus 1 us (1 MHz), and the time between two measured values of two adjacent channels is $10 * S_sampTimebase$, thus 10 us (= minimum time). The time between two complete scans through all channels is $6667 * S_sampTimebase = 1/150$ and thus corresponds to 150 Hz.

4.3.3 The ‘MajorScan’ Program

The MajorScan program allows the recording of any number of channels with the following frequencies: 1, 10, 50, 150, 300, 500, 2000 Hz. The measured values in ticks are stored in an array, which can be stored together with the sampling parameters (frequency, gain, etc.) after the measurement in binary format (smaller, faster) or in Ascii format (readable) is completed. Import routines for MATLAB are also present to further process the data there.

4.4 Measurements of the Sensors

From our measurements, we would like to obtain the following information:

- Comparison data for evaluating the most suitable sensors (ADXL210 vs. ADXL250 / ENC03 vs. ENV05)
- Data for generating the sensor model (characteristics)
- Data for correction algorithms
- Verification of analog filters and amplifiers (cut-off frequency & optimum signal matching)

4.4.1 Evaluation of the Sensors

To be able to compare the sensors, one plug-in position each is present on the test board for each type of sensor. Thus, the various sensor outputs could be directly compared to one another within a measurement.

4.4.2 Sensor Models

Here, any number of parameters can be taken into consideration, such as, e.g., temperature drift, VDD drift, vibrations, acceleration and angular velocity characteristics, hystereses, static, in-package alignment errors of the dies, etc. For most of these parameters, however, expensive systems such as rotary tables, acceleration sleds, etc., are required. We therefore of necessity limited ourselves to the measurements that are possible for us: simple acceleration measurements with the aid of acceleration due to gravity and temperature-bias drift measurements with a climatic exposure test cabinet.

4.4.3 Temperature Measurements

The temperature measurements were performed in a climatic exposure test

cabinet, which could be cooled and heated. To avoid the formation of condensate on the electronics during the measurement, the temperature profile appeared as follows: 60, 50, 40, 30, 20, 10 degrees Celsius, in each case for 20 minutes. The zero values (unaccelerated, not moved) were measured and recorded every second with the MajorScan application. Then, it was stored both as *.bin(binary) and as *.dat (ascii). In MATLAB, the obtained data were then evaluated. (See attachment, or CD). All sensors were measured several times to be able to make a statement on the reproducibility.

4.4.4 Acceleration Measurements

As reference acceleration, +1 g, 0 g and -1 g of gravitation were used. Thus, the data are complete, and a simple, linear sensor model can be derived in the following form

$$\alpha(\mathbf{v}, \mathbf{T}) = Sens * v + b(T)$$

whereby α is the acceleration, *Sens* is the measured sensitivity, and b is the temperature-dependent bias voltage as a function of temperature.

4.4.5 Algorithms

Here, there was not much to measure. The only question was whether it provides a phase shift between the arrivals of the signals from the individual sensors, i.e., whether the individual sensors have different response times. To find this out, we suspended the test board on a wire and then hit it with a hammer, a shock in the form of an impact. In this case, the values were recorded at 2000 Hz, and the data obtained was evaluated in MATLAB. We could not determine any phase shift from the pulse response. At least,

the shift is so small that in our system, it plays no role or at least a subordinate role. We therefore resolved to disregard this influence.

4.4.6 Filter

The only question in the case of the filters was to check the suitability and the calculated frequency responses. For the frequency responses of the Second Order Butterworth low-pass filters, we generated sinusoidal signals with a signal generator and visualized the resulting starting amplitude with an oscilloscope (manual sweep). The printout with the calculated and simulated values was amazingly good. We decided to acquire the filters for the next board.

The analog high-pass filter proposed in the data sheet of Murata for the ENC03 (for temperature compensation) with the very low cut-off frequency of 0.3 Hz led to distorted amplitude and phase plots, however. The response to a uniform (as much as possible) rotation by 90 degrees corresponded to everything but the expected square-wave signal. After more precise analysis of the signal with MATLAB, we had to determine that this filter circuit was impossibly suited for our purposes. The temperature-bias compensation consequently had to be produced in another way. We therefore then produced a DC coupling of the sensor with a filter and amplifier stage that optimally matched the signal to the A/D-converter input. Here, distortion of the signal could now no longer be detected.

However, the measured temperature drift of the ENC03 gyro was smaller than what was indicated on the data sheet. Thus, temperature compensation is also possible only behind the A/D converter in digital form.

5 Implementation II: Sensor Board

5.1 Overview

With the findings that we had obtained from the design of the first board, we developed the second board, our so-called sensor board. This board should have acceleration and rotational sensors that cover all degrees of freedom and thus make possible a determination of movement and position in space. The position in space can then be calculated from this. We use three ADXL210 acceleration sensors in X, Y and Z direction, two ENC03 gyros for the measurement of the rotation around the X- and Y-axes, and an ENV05 gyro for measuring the rotation around the Z-axis. To be able to make comparison measurements of the inertial navigation system with a reference source, a GPS receiver was incorporated, which delivers its data to the serial interface via the incorporated level converter MAX232.

5.2 The Hardware of the Sensor Board

5.2.1 Power Supply

The device can be supplied with current directly by the PCMCIA measuring card. For this purpose, the card offers a 5 V output, which can be charged with a maximum of 500 mA. In contrast, we have provided a connector plug (CON7_1) on the board of our device for an external supply of 5 V so that the system can also be operated in connection with a DSP board with an incorporated A/D converter. With this connection, the possibility exists to supply sufficient current for additional components of the system without bringing in the supply voltage. Care must be taken that a stable voltage source is

connected to exactly 5 V, since the electronics and especially the sensors had been designed for this voltage. Deviations therefrom are primarily noted in reduced sensitivity of sensor data signals. A switch can be made between these two types of operations by means of jumpers J7_1 and J8_1. To protect the measuring card from possible overloads during short-circuiting, a back-up of 500 mA had been incorporated.

5.2.2 GPS & MAX232

The GPS and the fan optionally can be turned off. The GPS and the related MAX232 level converter are turned off via the jumper J1_1. To prevent possible complications with ripple pick-ups or small potential differences due to a ground line that exists in two places, we additionally incorporated a jumper Gnd_J1_16. This jumper has the object of interrupting the bonding of inertial navigation systems via the RS232 connection to the laptop if the mass potential had already been defined by the power supply via the connecting cable of the PCMCIA measuring card.

The data exchange of the GPS with the laptop is carried out via the serial port A of the GPS. This port is also used for optional changes of the configurational settings of the GPS. The μ -blox demo-software can be used to alter configuration settings or to verify the function of the GPS. In addition, the GPS has available a second serial port B for a DGPS system. This port was also guided via the MAX232 and is present on the motherboard in the form of pin connectors. As per the data sheet, the RS232 input line (RS232 B OUT), which supplies the GPS data, has to be grounded when not in use. For this purpose, the jumper JI_16 is provided.

5.2.3 Temperature Compensation

To be able to measure the temperature of all sensors with a single temperature sensor, we have embedded the sensors in a type of flow channel at whose input a small fan supplies a continuous air stream.

We know that our platform reacts in a very sensitive manner to temperature fluctuations; this property is even more amplified with the flow channel, and the device obviously deteriorates at first glance. With this measure, however, all sensors have the same temperature, the temperature measurement of the AD780 only then makes sense, and compensation is actually made possible. Thus, the sensors are not considerably disrupted by strong temperature fluctuations, but the complete system has to be thermally inert. This in turn can take place with good outer insulation of the device or by means of positioning a site with a moderate temperature plot.



Figure 7: The Sensor Board with the Flow Channel and the Fan

5.2.4 Filter & Amplifier

We have included the filter cut-off frequencies and the amplifications of the individual sensor output wirings largely from the first board. At the temperature output, the temperature sensor AD780 received an amplifier circuit with $A = 7.8$ for the purpose of better resolution. The original AC coupling of the ENC03 was removed, since it distorted the signal. It was replaced by a pure DC coupling with subsequent level matching.

Since we had found out from tests with the first board that the ENC03 gyros are quite inaccurate, we have provided the incorporation of more precise ENV05 gyros parallel to the ENC03 gyros. Thus, the ENC03 gyros can be replaced simply with those of type ENV05. The filters and amplifiers that are required for this purpose are already present on the board; they have to set only the corresponding jumpers (J2_1 / J3_1 for the x-axis, J4_1 / J5_1 for the y-axis), and the inputs of the no longer required ENC03 filter are switched to GND, so that the latter do not begin to oscillate. In the case of ENC03, the termination of the filter inputs takes place directly on the understructure (to do so, a wire link has to be inserted from the GND pin after OUT X or OUT Y). In the ENV05, for this purpose, specifically the connector is available, which the input signal of the filter circuit can ground with a jumper.

Sensor	Fc (-3dB) Filter [Hz]	Amplification [A]
ADXL210	50	2.2
ENC03	50	10
ENV05	25	1.2
AD780	-	7.8

Table 2: Filters and Amplifiers of the Sensor Board

Otherwise, the circuits were maintained. The filter type, as before, is a Butterworth filter of the second order. Table 2 offers an overview on the filter coefficients and the individual amplifications.

5.3 Software for the Sensor Board

There are two programs for this board. On the one hand, the ‘cockpit,’ which graphically depicts the location in space by means of artificial horizon and gyro compass, is based on an aircraft cockpit. This program was acquired, i.a., for debugging and for demonstration purposes. On the other hand, we have written the actual ‘Espresso’ navigation program. Here, functions or at least functional containers are present that take on all important objects of navigation, such as e.g., initialization, all coordinate transformations, feedback, compensations, etc. In this chapter, these two programs and their structure are to be explained more precisely.

5.3.1 ‘Cockpit’ Demo Program

Here, essentially the rotational speeds are measured in components of which three Euler angles are separated and integrated. For more specific clarifications on transformation, see Chapter 6.

Since only the gyros are used, which makes necessary only zeroing of the currently active offset voltage, the initialization could be maintained very simply: during the time period in which the checkbox ‘Zero’ is activated, the mean values of the measured data are formed, and we obtain our offset values. The longer we form the mean value, the more precise it should be.



Figure 8: The Sensor Board with Individual Sensors

With the ‘initialize’ button, the values of the Euler angle and the feedback are reset to their original values.

With the ‘feedback’ button, the feedback can be turned on or off. α and β (bank and pitch) are oriented according to the lot determined with the acceleration sensors. The functions `f_reg_omega ()` and `f_reg_phi()` determine the rotational speed or the angular position control. In our demo, the error between angular position and measured g-vector at constant speed (`c_reg_phi`) is reduced to zero. The rotational speed that corresponds to this must then be matched retroactively (`c_reg_omega`). The direction of the virtual compass cannot be adjusted for lack of reference. For further clarifications, see Chapter 6.

5.3.2 'Espresso' Navigation Program

'Espresso' offers significantly more functionality than 'MajorScan.' It relies on a transparent structure that can be simply expanded. All functional groups were consolidated in separate functions, so that changes can be made in partial areas and can be immediately tested. To explain the software structure, a data flow diagram is found attached.

The values that the DAQ-Card 1200 measures function (f_handle_Buf()) as a record with the fields (ax, ay, az, wx, wy, wz, T). This now ensures that all subsequent functions are called up in the correct sequence, and the data are written into the overall positional structure AktPos. The raw measured data in ticks are compensated by the corresponding temperature compensation functions of the individual sensors based on the current temperature. Then, the zero offset has to be subtracted, and the ticks are converted into m/s^2 . Acceleration now takes on f_koord_transform_trans() and the rotational speeds f_koord_transform_rot(). The individual components of acceleration are required to be completely in the original coordinate system, while the rotational speeds have to be plotted in components of the Euler angle. Therefore, two different transformation functions are present in the software. There now follow in succession a regulation (f_reg_accel()/f_reg_omega()), an integration (f_int_accel()/f_int_omega()), and a regulation (f_reg_speed()/f_reg_pos()). In acceleration, an integration (f_int_speed()) and a regulation (f_reg_pos()) again follow, by which the speed on the distance covered can be derived. Now, we have newly calculated the position and are finished with the calculations. For each measured and plotted record, the same procedure is repeated.

The transformations and integrations are actually ‘static,’ i.e., implemented one time; they hardly have to be changed. However, the other functions are freely available to changes and expansions. Thus, e.g., a positional feedback could be carried out with, e.g., `f_reg_pos()`, but this regulation also has an effect on the current speed and acceleration, which then also have to be readjusted.

5.3.3 Recording the Data

The raw data in ticks are stored completely in one array and can be stored on a disk, as in ‘MajorScan.’ Thus, e.g., in MATLAB, the navigation can be reproduced or the ‘off-line’ algorithms can be optimized.

6 Implementation III: Algorithms

In this chapter, the algorithms that are used are explained and, if need be, literature citations are given. We confronted the following problems in these areas:

- Representation of the position and orientation in space (angular position)
- Transformation of acceleration and angular velocities in the above-defined coordinate system
- Correct initialization before the measuring is begun
- Regulation process/feedback
- Temperature compensation of the sensors

6.1 Representation in Space

Here, essentially the selection of a suitable coordinate system is involved, whereby all six degrees of freedom have to be represented correctly. In general, these are three translatory coordinates and three rotating coordinates. However, e.g., the three angles around the axes of a Cartesian coordinate system do *not completely* describe the *location* of the object in space. There are, however, many types of coordinate systems that completely produce this description, in each case with its advantages and disadvantages [5].

6.1.1 Translatory Coordinates

For the positional coordinates, in most cases, degrees of length and width are used

in navigation. This makes no sense to us, since we move only in a very limited space, where the curvature of the earth does not have a role. Therefore, we decided on a simple, Cartesian coordinate system as a reference. At present zero, the x-axis goes right to the side, the y-axis goes forward, and the z-axis goes up, i.e., a right-handed system.

6.1.2 Rotating Coordinates

The orientation in space is usually represented by one of the following three types: '*Direction Cosines*,' *Euler Angle* or '*Quaternions*.'

The '*Direction Cosines*' method consists of a 3 x 3 matrix with the cosine values of the directions to the axes. Transformations can thus be performed in a simple way, but the representation is redundant and in no way intuitive.

It is already better in the representation by means of *Euler angles*. The orientation is given with three skillfully selected angles in space. This is very intuitive and efficient, but it has the drawback that at two points, a type of singularity exists (north and south pole); there, the representation is no longer clear. Also, an increasing number of processing errors can be expected if these points are approached.

The third method with the '*Quaternions*' describes an orientation through a 3D vector and a rotation by the same. Thus, a vector is produced with four elements. Here, no singularity exists, but the representation is also non-intuitive.

We ultimately selected a representation with Euler angles, since the latter have to be calculated with all methods for a visual representation in any event, and the time necessary for computing thus turns out to be somewhat less here. The angles were defined as follows: α as an angle of rotation around the longitudinal axis (y''-axis), β

around the transversal axis (x' -axis), and γ around the vertical axis (z -axis). This produces a quite simple transformation, since two sensors rotate ‘nearly’ directly around their axes. The z -axis is that of our Cartesian original coordinate system. The x' -axis is rotated relative to the original x -axis by the angle γ , and the y'' -axis is perpendicular to the x' -axis, in the direction of the axis of the device. The three angles could also be referred to intuitively as bank (α), pitch (β), and direction (γ). And these are precisely the values that are graphically indicated in the ‘cockpit’ with the artificial horizon and the compass.

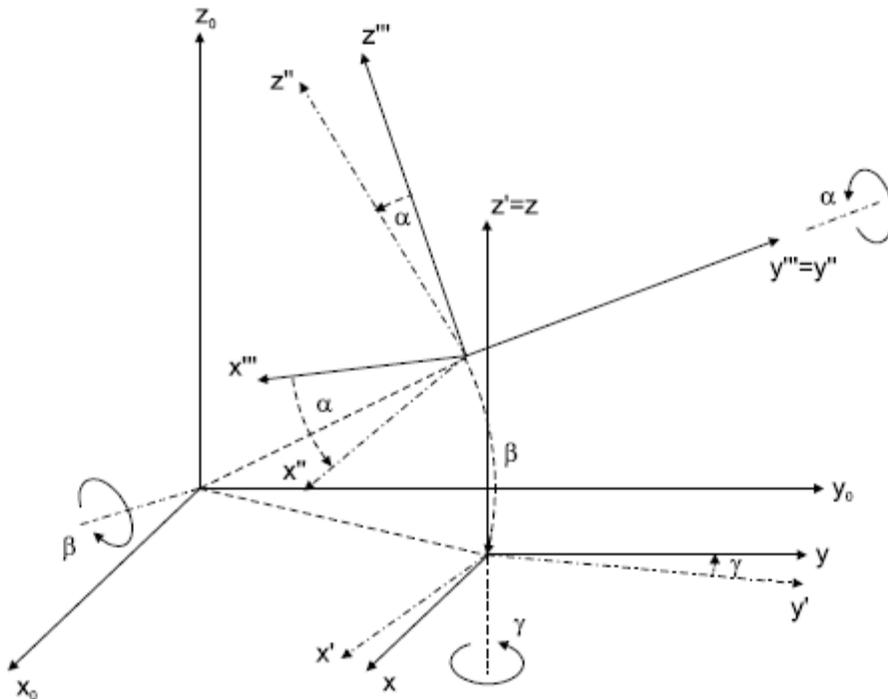


Figure 9: Representation of the Location in Space by Means of Euler Angles

6.2 Transformation of Measured Values in Our Coordinate System

Accelerations in the direction of the device coordinate axes (x'' , y'' , z'') are

rotated by the angles α , β , and γ (in this sequence). The transformation matrix appears as follows:

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} \cos(\gamma) * \cos(\alpha) & -\sin(\gamma) * \cos(\beta) & -\cos(\alpha) * \sin(\beta) * \sin(\gamma) \\ -\sin(\gamma) * \sin(\beta) * \sin(\alpha) & \cos(\gamma) * \cos(\beta) & +\sin(\alpha) * \cos(\gamma) \\ \sin(\beta) * \sin(\alpha) * \cos(\gamma) & \cos(\gamma) * \cos(\beta) & \cos(\gamma) * \sin(\beta) * \cos(\alpha) \\ -\cos(\alpha) * \sin(\gamma) & \cos(\gamma) * \cos(\beta) & +\sin(\gamma) * \sin(\alpha) \\ \cos(\beta) * \sin(\alpha) & -\sin(\beta) & \cos(\beta) * \cos(\alpha) \end{pmatrix} \begin{pmatrix} \Delta x''' \\ \Delta y''' \\ \Delta z''' \end{pmatrix}$$

The rotational speeds around the device coordinate axes are calculated in ‘Euler angle speeds’:

$$\begin{pmatrix} \Delta \alpha \\ \Delta \beta \\ \Delta \gamma \end{pmatrix} = \begin{pmatrix} 1 & \sin(\alpha) * \tan(\beta) & -\cos(\alpha) * \tan(\beta) \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) / \cos(\beta) & \cos(\alpha) / \cos(\beta) \end{pmatrix} \begin{pmatrix} \Delta \alpha''' \\ \Delta \beta''' \\ \Delta \gamma''' \end{pmatrix}$$

There are still the following to be considered: In the case of larger angles, the *sequence* of rotations plays an essential role! As an example: it makes a difference whether an individual near the North Pole first goes toward the south and then toward the west or first goes toward the west and then toward the south. These errors, however, can be disregarded in the case of small angles. The interval between two measurements thus has to be small in comparison to the maximum rotational speed.

6.3 Regulation Process/Feedback

For example, positional updates with GPS, orientation of position according to the g-vector, a compass or tilt sensors, statistical methods, etc., [6] fall into the category of feedback. The range of uses is limited, however, with a lot of feedback. Thus, e.g., in an

orientation according to the g-vector, it is assumed that it is found in a gravitational field and is in general not accelerated. In most cases, however, this also does not apply to an Earth orbit (zero gravity) or a carousel (centripetal acceleration). Feedback is complicated by the fact that an integration-determined value cannot simply be changed without feeding back the changes to the differential value. Thus, the position cannot be updated with the GPS without the speed also being corrected. In the case of a GPS, is there just one speed, but what about the case of feedback based on the g-vector? The rotational speed slowly drifts away, the position is readjusted again and again, and still the increase in positional drift also rises until finally the positional regulator can no longer compensate. To avoid this, the drift of the rotational speed also has to be regulated back to zero. This is performed precisely in our positional feedback in ‘Cockpit’ and ‘Espresso.’

Another problem exists if, e.g., I want to regulate the rotational speeds according to certain factors alone, for example with the aid of a low-pass filter. Since it is then integrated, the danger exists that the position resulting therefrom will begin to oscillate.

6.4 Initialization

In initialization, various values are to be determined, such as, e.g., speed, position and location. Moreover, a zeroing of the sensors is also to be carried out. While the zeroing relates to changes in the sensor board, the other *external* values are to be determined. Here, ‘external’ means either measuring them with the necessary accuracy or feeding them back to the system implicitly by means of instructions upon initialization [6].

For our experimental set-up, the global positional coordinates are not relevant, therefore they are set at zero in the initialization ($x = 0, y = 0, z = 0$). The starting position could also be obtained, however, by a GPS receiver. We cannot measure the initial orientation of the sensor board since neither tilt sensors nor a compass are present. Therefore, we prescribe that the board must be *oriented flat* and has to be settled, and we define the longitudinal axis of the device (i.e., the y-axis) as north. (See ‘Espresso’: Sub Start_Initialize()).

6.5 Temperature Compensation

In this section, algorithms for compensation of the bias drift based on temperature are proposed. The measured data used for this purpose are present as *.bin or* .dat Files or as Matlab *.fig graphics on the enclosed CD. The bias drift relative to temperature was approximated with a polynomial of the smallest possible degree.

The ADXL210 is very temperature-stable: it varies over the range of 60 degrees Celsius by only about 5 ticks; this corresponds approximately to 6 mV or 0.12%. The results are easy to reproduce, and the individual sensors all showed the same behavior. The temperature dependency, moreover, can be linearized in a simple way. We refer to the temperature as T_{ref} , which was measured in the initialization of the device. Here, our compensation function of the ADXL210 that was used:

$$a_{comp}(\mathbf{T}) = a_{mess} - 0.1 * (T - T_{ref})$$

The ENC03 is not very temperature-stable. On the one hand, it varies statically (i.e., based on the temperature swings) by about 150 ticks or 180 mV or 3.6% over a range of 60 degrees Celsius. On the other hand, it also has a dynamic component, i.e.,

the bias is also dependent on the *temperature differential*. In this case, the magnitude of the effect changes based on how fast the temperature changes. This component is somewhat larger than the static portion depending on the speed of change. We have not found any good compensation function for solving this problem; in particular the conditioning cabinet was not able to maintain a stable temperature but rather produced a temperature curve – which could go up or down by approximately three degrees - with its two-point regulator.

The ENV05 has a much better temperature behavior than the ENC03. It drifts by only about 30 ticks, 36 mV or 0.7% over a range of 60 degrees Celsius. This qualitative difference manifests itself naturally in cost. It also reacts, however, to the temperature differential, but only slightly, certainly also because of the much greater mass. Specifically this mass, however, also has to be taken into consideration in the case of temperature compensation. A temperature change is only slowly effective in the sensor itself and thus also in the effect on the output signal. The best results are obtained when the temperatures remain for about 400 seconds and form the mean value. Then, the following compensation function can be applied:

$$\omega_{\text{comp}}(\mathbf{T}) = \omega_{\text{mess}} + 2.608 * 10^{-5} * T^3 - 0.0435 * T^2 + 23.213 * T - 1981.2$$

The reproducibility of the sensor data is fairly good. We had only a single ENV05 available, however, and thus we cannot make any statement on whether every ENV05 can be compensated by such a function.

Finally, one more comment: In general, the temperature does not change excessively quickly. Our navigation platform without feedback is stable for only a quite short time, however. The importance of the temperature compensation functions in our

system therefore has to qualify, since during the short time in which plausible data are available, the temperature of the sensors can be assumed in good approximation to be constant.

7 Results

7.1 Gauging the Finished System

The measuring of the system was one of the main objects that we had to meet. It should be noted how precise it is, where the sources of the inaccuracies are to be found, and with which measures a higher accuracy was to be achieved.



Figure 10: The Finished System with Laptop, Sensor Board and PCMCIA Measuring Card

7.1.1 Electrical Verification

The electrical verification confirmed the actual freedom from defects of the sensor board. Two problems had to be detected and eliminated, however. The filters oscillated, and the supply voltage dropped excessively with increasing power consumption.

7.1.2 Oscillating Filter

The reason for the oscillation of the filter was in the new operational amplifiers of type LM6134. They have a gain-bandwidth of 10 MHz and a slew rate of $12 \text{ V}/\mu\text{s}$. After replacement by the type-OP491 op-amps of the test board, the filters functioned optimally again, since these have a gain-bandwidth of only 3 MHz and a slew rate of $0.5 \text{ V}/\mu\text{s}$. The oscillation occurred only in cascaded operational amplifiers, thus, e.g., in the circuit of the ENC03, where the filter also follows an amplifier stage.

7.1.3 Power Supply

The power supply is free of high-frequency disruptions, however, the following power-supply values were measured under various loads:

Configuration	Voltage [V]	Power Consumption [mA]
Sensors & Filter & LED	5.00	27.8
“ & Fan	4.87	109
“ & GPS & MAX232 (without Fan)	4.75	193
“ % GPS & MAX232 & Fan	4.64	270

Table 3: Voltage and Current Measurements on the Sensor Board

According to the specification, the DAQ-Card 1200 delivers 500 mA. The power supply drops already, however, with smaller loads. A shorter cable between the I/O card and the sensor board could perhaps defuse the problem somewhat. As a result, we have turned off the fan and the GPS for our measurements to eliminate all unnecessary power consumers.

7.1.4 Positional Measurements

The position of our device was measured statically and dynamically, i.e., at a standstill and with movements in space. The dynamic measurements are not very significant, since the movements have to be made by hand because of defective measuring devices. They therefore cannot be reproduced; nevertheless, they indicate various error sources.

7.1.5 Static Positional Measurement

To this end, the ‘Cockpit’ program was used. In this case, the sensor board was on the floor to preclude vibration. Immediately after the board is turned on, the gyro data are prone to an increased number of errors, and thus we always began the measurements after an initial transient oscillation phase of at least 20 minutes. Repeated measurements were made at 150, 300 and 500 Hz. The results are combined in Table 4.

Sensor	150 Hz	300 Hz	500 Hz
Horizontal Position, 30”	2.5 degrees	2.0 degrees	2.0 degrees
Direction, 30”	0.7 degree	0.5 degree	0.5 degree
Horizontal Position, 60”	5.0 degrees	5.0 degrees	5.0 degrees
Direction, 60”	1.5 degrees	1.1 degrees	1.1 degrees

Table 4: Drift of Unregulated Positional Measurement

The indicated values should not be interpreted too precisely since the scattering of the repeated measurements is quite large. It is recognized, however, that no significant improvement can be achieved by higher sampling rates. (The system is also motionless.) The direction that is read out is more accurate than the horizontal position. This depends

on the direction being measured at a standstill by the more precise Gyro ENV05. In the case of the dynamic measurement, however, the different accuracies are then intermingled by the transformations.

7.1.6 Dynamic Positional Measurement

The sensor board was moved in slowly gyrating movements in all directions and in turn measured at three different sampling frequencies. In this case, it is important that the rotational speed limits of the gyros are not exceeded. In this connection, the ENV05 with at most 80 degrees per second represents the 'weakest link.'

In movements with less than an approximately 30-degree deflection, the deviations are on the same order of magnitude as when the system is in the position of rest, regardless of the sampling rate.

If greater deflections are made, the errors increase based on the sampling frequency. At 70 degrees, they are about three times as large with 150 Hz, and they are about twice as large with 500 Hz.

7.1.7 Static Positional Measurement

In this connection, the 'Espresso' program was used. In this case, the sensor board was in turn on the floor and was not moved. The positional measurement depends greatly on a correct positional measurement, since the position between acceleration due to gravity and progressive movement acceleration can be distinguished only with assistance. Thus, an error of 1 degree in the horizontal position already results after one minute in an error of position of 314 m or a speed of 10.5 m/s. The g-vector righting

mechanism discussed in Chapter 6 therefore ensured as horizontal a platform as possible during our measurements. The measured errors of position can be found in Table 5.

Sensor	150 Hz	300 Hz	500 Hz
Deviation, 30''	8 m	8 m	8 m
Deviation, 60''	40 m	40 m	40 m

Table 5: Drift of the Positional Measurement (Regulated Position)

As in the positional measurement, the scattering is also considerable here; the values are therefore defined only as an order of magnitude. We have made a dynamic positional measurement, but no useful values emerged.

7.2 Findings

7.2.1 Power Supply

The power requirement of the system including GPS and fan is 270 mA and thus is found within the specifications of the PCMCIA measuring card. Only the sensors and filters can hereby be sufficiently supplied with current, however, although it was originally provided that all components could be supplied via the measuring card. The voltage drop by 0.35 V via the measurement cable to the PCMCIA card was too high, however. In measurements in this type of operation, we therefore had not connected the GPS and the fan in order not to have a negative influence on the measured data.

7.2.2 Temperature Compensation

In principle, it is possible to achieve a good temperature compensation of the

sensors, since the temperature dependencies can be reproduced. The problem exists with ENC03, however, that the effect of temperature on the signal still has a differential proportion, i.e., if the temperature changes more quickly, the distortion is correspondingly greater. (It can be observed here that this sensor behaves like a ‘rotational-speed-dependent temperature sensor.’) We have not found any suitable function that can correct both quick and slow changes.

Thus, our only remaining option was to assume appropriate environmental conditions. We define appropriate environmental conditions as a moderate climate without quick temperature changes in terms of measurement duration. At the current time, this is a sensitive measuring device, in which such conditions have to be complied with. However, it is essential to take into consideration that the time span, during which the system should supply accurate results without updates, plays a very important role. With the current inertial navigation platform, it has no appreciation for performing an expensive temperature compensation, since the position may be inoperative after less than one minute of operation without updates.

7.2.3 Righting Mechanism

Based on the comparatively large drift of the ENC03 gyros, a longer-lasting measurement is only useful if this drift can be compensated. With a righting mechanism, as it is described in Chapter 6, this deficiency can be counteracted successfully.

7.2.4 Initialization

Initialization represents a problem by itself. When turning on the navigation

platform, various values have to be determined, in whose calculation, however, problems arise. We want to explain these problems more precisely below.

Position and location in space are calculated as follows:

$$\mathbf{s}(t) = \iint a(t)dt^2 + \int v_0 dt + s_0$$

and

$$\varphi(t) = \int \omega(t)dt + \varphi_0$$

whereby $\mathbf{s}(t)$ stands for the distance vector from the origin and $\varphi(t)$ stands for the location vector (α, β, γ) . In the initialization, we now have to determine v_0 , s_0 , and φ_0 , i.e., the initial speed, the initial position, and the original location in space. In addition, we would also like to know the zero offset (e.g., by aging, incomplete temperature compensation, etc.) of the six sensors, since a deviation there is integrated directly in an error of position or location. These are thus $5 * 3 = 15$ initial values that are to be determined.

The position and the speed in the initialization can be measured with a GPS, and thus these values can be determined. The GPS makes no statement on the location in space, however, and the zero offset of the sensor also cannot be measured. We require more information for this, i.e., additional sensors.

Some sensors can be eliminated, however, if the initialization conditions are specified appropriately. It would be conceivable that motionlessness and a flat location for the initialization of the device are assumed, and thus v_0 and φ_0 are implicitly zero. Also, the zero values of the acceleration and rotational speed sensors can now be measured directly up to the acceleration in z-direction, which is namely +1 g. All

parameters are determined in this way. These special requirements are not, however, always achievable. I cannot always position my car exactly horizontal to initialize my navigation system. In the general case, to allow for a sloping initial location (φ_0), it also has to be able to be measured. To this end, I must not use my acceleration sensors under any circumstances since I otherwise can no longer correctly initialize the latter. An inaccuracy of a component cannot be corrected with the aid of data that themselves originate from these components.

The location and the zero offset of six sensors together produce nine values; we thus also use nine independent measurement categories to determine the latter. More information is asked for; to this end, e.g., two tilt sensors and a compass are necessary.

7.2.5 Heading Representation

Three frequently-used types of a representation of the direction in space are described in Chapter 6. The representation with the Euler angles that we selected has major advantages as regards processing power and understandability. If all possible orientations have been taken, up to and including the perpendicular position, however, problems arise since singularities exist there and considerably amplify the processing errors near them. For a motor vehicle or a two-seater airplane, this represents no problem. However, if the platform is to be used on a robot arm or on an acrobatic aircraft, wherein vertical positions can be assumed, a positional representation with ‘quaternions’ or ‘direction cosines’ without singularities is to be preferred. In another step, the Euler angles could always still be calculated therefrom to obtain a representation that is understandable to humans.

7.2.6 Processing Power

In the 'Espresso' program, about 80 multiplications, 12 sine () or cosine() and 2 arc tan() functions are used per complete sample record, a multiple of additions. At a sampling rate of 300 Hz, this produces approximately 20,000 64-bit multiplications and 4,200 trigonometric functions per second. Contained in these numbers are the presented temperature compensations, the transformations, and the positional righting mechanism. Depending on expense in the preparation of sensor data and the feedback mechanisms that are used, of course considerably more processing power is used.

7.3 Sources of Inaccuracies

In our opinion, the following error sources are relevant:

- Limited resolution of sensors
- Bias and sensitivity drift
- Nonlinearities of sensors
- Alignment errors (die vs. package, package vs. board)
- Interpolation and quantization errors
- Processing errors based on excessive angular changes
- Mutual error accumulation

7.3.1 Limited Resolution

One of the main causes of the inaccuracies in the overall system lies in the limited resolution of sensors. The ENV05 can measure rotational speeds up to a minimum of 0.1 degree per second. If half-resolution is assumed as an error, i.e., 0.05 degree per second,

this alone produces a deviation of 3 degrees per minute. The data sheet of ENC03 does not yield any corresponding information, but the sensor is definitely more inaccurate.

The following formula to calculate the ‘noise’ can be found in the data sheets of ADXL210. We have prepared the cut-off frequency of our filter at 50 Hz:

$$\text{Noise(rms)} = (500\mu\text{g}/\sqrt{\text{Hz}}) \times (\sqrt{(50\text{Hz} \times 1.5)}) = 4.3\text{mg}$$

7.3.2 Bias and Sensitivity Drift/Nonlinearities

At a standstill, errors result primarily from the bias-drift. Nonlinearities and sensitivity fluctuations play a subordinate role here, since optional errors can be compensated by the initialization (zero offset). The drift is produced for the largest part by the temperature. For an illustration, a finger can be put on an ENC03 during measurement, and the drifting location can be observed immediately.

In a movement, nonlinearities and sensitivity changes of the sensors now come into play. According to the data sheet, the ENC03 has a deviation of linearity of 5% full scale (FS), the ENV05 of at least 0.5% FS. The sensitivity fluctuates both at -20/+10% full scale, and since the calculation of the measured angular velocities at the Euler angle speeds takes place with the Euler angles themselves, associated with the linearity errors, new errors develop.

The accuracy of the acceleration sensors in this respect is somewhat better, since the linearity errors thereof only amount to 0.2% full scale. By the doubled integration, this advantage is more than balanced, however.

7.3.3 Alignment Errors

According to the data sheet, the two sensors with ADXL210 have a maximum alignment error between die and package of ± 1 degree. Additional alignment errors are produced by the inaccuracy in the mechanical arrangement of sensors on the sensor board. The coordinate transformations start from the fact, however, that the sensors are perpendicular to one another. It is obvious that an error occurs here.

The three above-mentioned error sources, namely the bias-drift, the nonlinearities and the alignment errors, can be compensated by good sensor models. To this end, however, quite expensive measuring devices are required. To determine the nonlinearity of the rotational speed sensors, a rotary table with accurate reference rotational speed is essential. For measuring the acceleration sensors, acceleration due to gravity is used, naturally only in a range of -1 to +1 g. Moreover, only one acceleration sled is assisted. Finally, the platform should be measured as an entire system. Here, the alignment errors of all sensors and the mechanical design can now be determined.

Thus, sensor models that can be parameterized, whose parameters were determined in 'factory calibration' and stored on the platform, were desirable. An INS platform is a sensitive measuring device, and thus such an intricate measurement of any individual device before start-up is justified.

7.3.4 Interpolation and Quantization Errors

Interpolation and quantization errors are produced by the scanning of analog signals. In our implementation, we only summed up the basic values in the integration. To estimate the order of magnitude of the interpolation errors, the platform was measured

with various scanning frequencies. Since the signals are filtered (cut-off frequency 25 Hz or 50 Hz), the interpolation error is smaller with increasing scanning frequency. As described in Chapter 7, the accuracy of our platform, however, is only minimally dependent thereon, i.e., errors exist that are larger in magnitude than the interpolation errors.

It can be noted in the quantization errors that the ENV05 gyro has a resolution of approximately 0.1 degree per second at a maximum of ± 80 degrees per second, thus 10.5 bits would be adequate, or with 12 bits, our A/D is certainly accurate enough. The resolution of the sensor itself is thus poorer than the A/D conversion.

The ADXL210 has a resolution of about 4.5 mg in a range of ± 10 g, i.e., the granularity of our scanning is about as large as the sensor signal itself. Here, the quantization thus plays a role, and an improvement could perhaps be reached by mean value formation and dithering of the analog signal. Our tests with activated dithering of the DAQ-Card 1200 and a mean value formation over eight measuring points do not provide recognizable improvement, however. Another variant would be the limiting of the range to, e.g., ± 5 g, and then the quantization would still turn out to be half as large as the resolution of the sensor.

Interpolation and quantization errors exist, but they are of subordinate importance in our design as the measurements showed.

7.3.5 Processing Errors Based on Excessive Angular Changes

Chapter 6 explains that excessive angular changes result in processing errors. It plays a role in whether first a rotation is carried out around the x-axis and then around the

y-axis, or vice versa. With very small angles, however, this error can be disregarded. It must also be ensured that the scanning intervals are kept small enough that a ‘simultaneous’ rotation around the axes can be assumed. Of course, these errors are indicated only in systems moved at high angular velocities. They are not relevant in static measurements. In our case, it is also not in the dynamic velocity, since at the maximum angular velocity of 80 degrees per second (ENV05) and a scanning frequency of 300 Hz, the angular changes are less than 0.3 degree.

7.3.6 Accumulation of Errors

The accumulation of errors is a problem in inertial navigation, but no actual error source. Each parameter can have effects on all of the others. An error in the position around the one axis distorts the transformations and thus the other position angle. This in turn leads to transformation errors of the accelerations and in addition causes a component of the g-vector to be erroneously integrated into a speed or distance as a motion acceleration. In this connection, this thus involves a mutual dependency and accumulation of the individual errors.

8 Results

An inertial navigation platform with low-cost components is possible in principle. The accuracy of the *positional measurement* that we achieved is already fairly good. To obtain a usable positional measurement over an extended time, it should be used in connection with a righting mechanism.

The *positional measurement*, however, is not possible at the current time. The positional information is unusable within the shortest time because of the limited accuracy of the sensors connected with the two-fold integration. Improving the resolution by a factor of at least 10 can realistically emerge in a determination of position.

By the correction of systematic errors, such as temperature drift, nonlinearities, sensitivity fluctuations, etc., in addition much can be done regarding the precision of the platform.

For an initialization of the platform in any position, 3 additional sensors (2 tilt sensors and 1 compass) are necessary, since otherwise the original position cannot be completely determined. The absence of these components is now circumvented by specifying the exact horizontal orientation upon initialization.

9 Outlook

Various problems have to be tackled to obtain a usable inertial navigation platform:

Since the resolution of the sensors was determined to be a significant source of errors, more precise sensors are an important requirement for better results. The current trend for better and less expensive sensors will make this possible.

Digital methods for signal processing, such as mean value formation or Kalman filters, also contribute to improvements, however. Specifically the Kalman filter can easily correct highly correlated measurement values, such as we find in problems of inertial navigation.

In the case of greater accuracy of the system, alignment errors between die and sensor package as well as between the board and the sensors increasingly play a role. This applies in measuring and then in correcting.

Additional useful feedback mechanisms have to be found, without which inertial navigation is not realistic in the long run. Our positional regulator according to the action of gravity has proven itself here, but further efforts are still necessary, in particular for positional feedback.

Another step is miniaturization. The system has to be smaller and thus portable to be used not only in a car, but also in areas such as ‘wearable computing’ or in ‘consumer electronics.’ The vision of an ‘INS on a chip’ is realistic and will reliably be implemented in the not-too-distant future.

Literature

- [1] C. F. O'Donnell et al., *Inertial Navigation, Analysis and Design*, McGraw-Hill, 1964.
- [2] Anthony Lawrence, *Modern Inertial Technology*, Springer-Verlag, 1993.
- [3] M. Lattmann, *G-Messer für den mobilen Einsatz [G-Meter for Mobile Use]*, ETH Zurich, IFE, 1999.
- [4] A. Stiller, *Digitale Filterhardware [Digital Filter Hardware]*, ETH Zurich, IFE, 1996.
- [5] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, Peter Peregrinus, Ltd., 1997.
- [6] E. Nebot, S. Sukkarieh, H. Durrant-Whyte, "Inertial Navigation Aided with GPS Information," in *M2VIP '97*. 1997, IEEE Press.

A 'Espresso' Software Structure

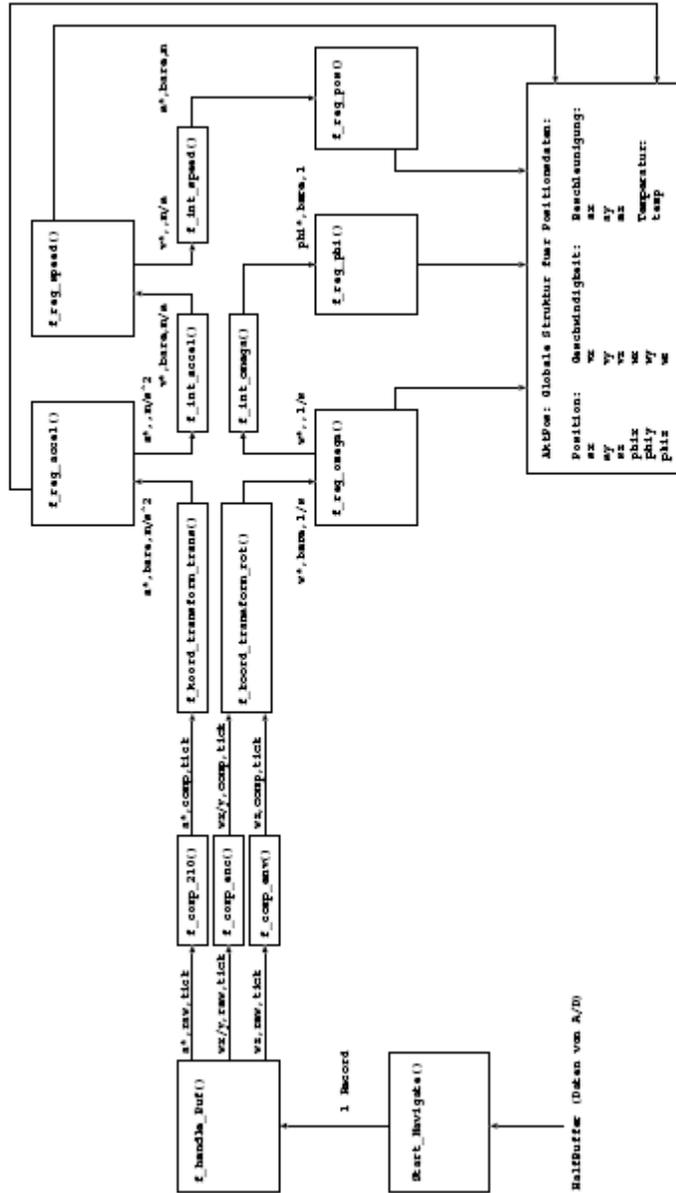


Figure 11: Software Structure of the 'Espresso' Program

[Key to Figure 11:]

Daten von A/D = A/D Data

AktPos: Globale Struktur fuer Positionsdaten: = AktPos: Overall Structure for Positional Data

Geschwindigkeit = Speed

Beschleunigung = Acceleration

Temperatur = Temperature

B Diagram of the Hardware

C Data Sheets