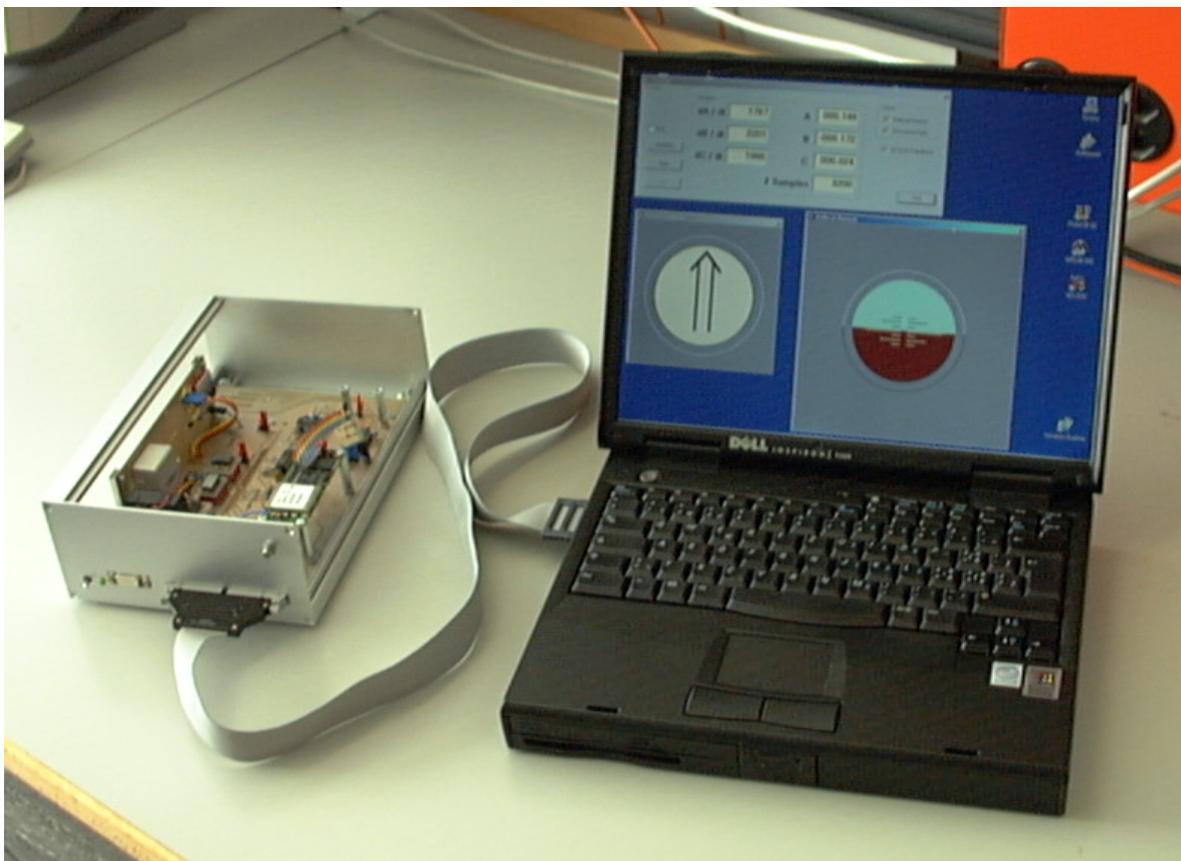


Low Cost Inertial Navigation System



Autoren: Peter Lüthi, Thomas Moser

Betreuer: Markus Uster, Teddy Loeliger

Professor: Prof. Dr. H. Jäckel

Semesterarbeit SS2000

Inhaltsverzeichnis

1 Zusammenfassung	6
2 Einleitung	7
2.1 Was ist Trägheitsnavigation ?	7
2.2 Evaluation unseres Vorgängerprojektes	9
2.2.1 Übersicht	9
2.2.2 Charakterisierung des Vorgängerprojektes	9
2.2.3 Entwicklungsumgebung	9
2.2.4 G-Messer	9
2.2.5 Schlussfolgerung	10
3 Konzept	11
3.1 Übersicht	11
3.2 Hardware	11
3.3 Software	12
4 Realisierung I: Test Board	13
4.1 Übersicht	13
4.2 Aufbau des Test Boards	13
4.2.1 Sensoren	13
4.2.2 Filter	14
4.3 Software	17
4.3.1 National Instruments DAQ-Card 1200	17
4.3.2 Sampling Parameter	17
4.3.3 Das Programm 'MajorScan'	18
4.4 Messungen der Sensoren	18
4.4.1 Evaluation der Sensoren	18
4.4.2 Sensormodelle	18
4.4.3 Temperaturmessungen	18
4.4.4 Beschleunigungsmessungen	19
4.4.5 Algorithmen	19
4.4.6 Filter	19
5 Realisierung II: Sensor Board	20
5.1 Übersicht	20
5.2 Die Hardware des Sensor Boards	20
5.2.1 Spannungsversorgung	20
5.2.2 GPS & MAX232	20
5.2.3 Temperaturkompensation	21
5.2.4 Filter & Verstärker	21
5.3 Software zum Sensor Board	22
5.3.1 Demoprogramm 'Cockpit'	22
5.3.2 Navigationsprogramm 'Espresso'	23
5.3.3 Aufzeichnen der Daten	24

6	Realisierung III: Algorithmen	25
6.1	Repräsentation im Raum	25
6.1.1	Translatorische Koordinaten	25
6.1.2	Rotatorische Koordinaten	25
6.2	Transformation der Messwerte in unser Koordinatensystem	27
6.3	Regelung / Rückführung	27
6.4	Initialisation	28
6.5	Temperaturkompensation	28
7	Resultate	30
7.1	Ausmessen des fertigen Systems	30
7.1.1	Elektrische Verifikation	30
7.1.2	Schwingende Filter	30
7.1.3	Spannungsversorgung	31
7.1.4	Lagemessungen	31
7.1.5	Statische Lagemessung	31
7.1.6	Dynamische Lagemessung	32
7.1.7	Statische Positionsmessung	32
7.2	Erkenntnisse	32
7.2.1	Spannungsversorgung	32
7.2.2	Temperaturkompensation	33
7.2.3	Aufrichtmechanismus	33
7.2.4	Initialisation	33
7.2.5	Richtungsrepräsentation	34
7.2.6	Rechenleistung	34
7.3	Quellen der Ungenauigkeiten	35
7.3.1	Beschränkte Auflösung	35
7.3.2	Bias- und Sensitivitätsdrift / Nichtlinearitäten	35
7.3.3	Alinierungsfehler	36
7.3.4	Interpolations- und Quantisierungsfehler	36
7.3.5	Rechenfehler aufgrund zu grosser Winkeländerungen	37
7.3.6	Fehlerakkumulation	37
8	Fazit	38
9	Ausblick	39
A	Softwarestruktur 'Espresso'	41
B	Schema der Hardware	45
C	Datenblätter	47

Abbildungsverzeichnis

1	Die 6 Freiheitsgrade des Raumes	7
2	Das neue Konzept mit Laptop, PCMCIA-Messkarte und Sensor Board	11
3	Beschleunigungssensor ADXL210	14
4	Gyroscope ENC03	14
5	Gyroscope ENV05	15
6	Das Test Board mit steckbaren Komponenten	16
7	Das Sensor Board mit dem Strömungskanal und dem Lüfter	21
8	Das Sensor Board mit den einzelnen Sensoren	23
9	Repräsentation der Lage im Raum mittels Eulerwinkel	26
10	Das fertige System mit Laptop, Sensor Board und PCMCIA-Messkarte	30
11	Softwarestruktur des Programms 'Espresso'	43

Tabellenverzeichnis

1	Filter & Verstärker des Test Boards	15
2	Filter & Verstärker des Sensor Boards	22
3	Spannungs- und Strommessungen am Sensor Board	31
4	Drift der unregelmäßigen Lagemessung	31
5	Drift der Positionsmessung (Lage geregelt)	32

1 Zusammenfassung

Die Trägheitsnavigation basiert auf Techniken, die sich nach dem Zweiten Weltkrieg entwickelt haben. Die ersten Systeme waren vollkommen mechanisch, gross und demzufolge technisch aufwändig und teuer. Danach kamen 'solid state' Lösungen auf, wie sie heute in der Zivilfliegerei zur Primärnavigation verwendet werden. Diese sind immer noch ziemlich teuer und gross. Seit kurzem existieren vollständig integrierte Beschleunigungs- und Drehgeschwindigkeitssensoren, die klein und vor allem sehr günstig sind. Es stellt sich hier nun die Frage, ob man mit diesen neuartigen Sensoren einen Schritt weiter gehen kann und durch Miniaturisierung eines Trägheitsnavigationssystems mit Hilfe solcher Sensoren den Anwendungsbereich der Trägheitsnavigation erweitern kann. Anwendungen sind im Bereich von Mobile Roboting, Wearable Computing, Automobile und Consumer Electronics genügend vorhanden.

Unser Projekt hatte zur Aufgabe, abzuklären, wie genau eine solche Plattform mit den zur Zeit erhältlichen Sensoren sein kann und wo die Quellen der Ungenauigkeiten liegen. Wir haben dazu einen Prototypen aufgebaut und die dazu notwendigen Algorithmen entwickelt. Durch die anschliessenden Messungen konnten wir aufzeigen, dass im Moment die erreichte Genauigkeit des Systems zum allergrössten Teil durch die beschränkte Auflösung der Beschleunigungs- und Drehgeschwindigkeitssensoren bedingt ist.

2 Einleitung

2.1 Was ist Trägheitsnavigation ?

Trägheitsnavigationsplattformen werden auch mit dem Begriff *INS* bezeichnet, wobei *INS* für *Inertial Navigation System* steht. Es ist hier von Trägheit die Rede, weil mittels einer Masse, bzw. deren Trägheit, die aktuelle Beschleunigung gemessen werden kann. Wenn die bisherige Beschleunigung bekannt ist, so kann durch Integration die Geschwindigkeit berechnet werden, mit der Geschwindigkeit wiederum der zurückgelegte Weg. Wenn man davon ausgeht, dass die anfängliche Position bekannt war, so kann auch die aktuelle Position durch den zurückgelegten Weg errechnet werden. Allerdings ist es wichtig, nicht nur die Beschleunigung selbst, sondern auch deren Richtung zu kennen. Dazu verwendet man sogenannte Gyroscopes, also Kreisel.

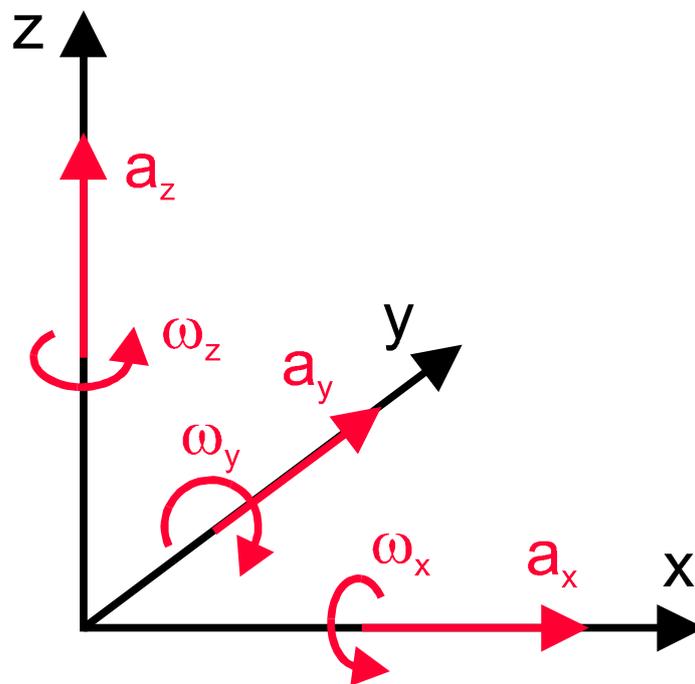


Abbildung 1: Die 6 Freiheitsgrade des Raumes

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

Die Trägheitsnavigation entwickelte sich nach dem zweiten Weltkrieg, z.B. bedienen sich bis heute alle Raumfahrtprogramme dieser Navigationsart, es ist also ein länger bekanntes Prinzip [1]. Anhand der Formeln erkennt man, dass sehr genau gemessen und gerechnet werden muss, um plausible Resultate zu erhalten: Ein Fehler in der Beschleunigung wird doppelt integriert und führt somit sehr schnell zu unbrauchbaren Werten! Um den Rechenaufwand zu minimieren, setzte man in den Anfangszeiten die Beschleunigungsmesser auf eine karthesisch gelagerte, kreiselstabilisierte Plattform. So zeigen alle Sensoren immer in die jeweils gleiche

Richtung, gleichgültig wie sich das Gefährt bewegt. Diese Lösung bedingt aber eine komplizierte, hochpräzise und somit fehleranfällige Mechanik, die entsprechend teuer ausfällt. Nicht zu vernachlässigen ist auch der hierfür benötigte Platz und der grosse Energiebedarf.

Mit dem Aufkommen von leistungsfähigeren Rechnern und nicht-mechanischen Kreiseln (z.B. Lasergyros) wurde dann Mitte der 80er Jahren der Bau sogenannter *Strap Down* INS Plattformen möglich. Hierbei werden Beschleunigungs- und Rotationssensoren benötigt. Diese Sensoren werden nun nicht mehr wie bis anhin in der Lage freibeweglich (karthesisch) aufgehängt, sondern mit der Struktur des Gefährts fest verbunden. Die Kreisel haben nicht mehr die Aufgabe, die Beschleunigungssensoren immer in dieselbe Richtung zeigen zu lassen, sondern sie müssen die Winkeländerungen messen. Dadurch kann bestimmt werden, in welche Richtung die Beschleunigungssensoren orientiert sind. Die gemessenen Beschleunigungen und Rotationen werden entsprechend umgerechnet, d.h. in das Referenzkoordinatensystem transformiert und erst dann integriert. Hiermit lässt sich dann auf die aktuelle Geschwindigkeit und Position schliessen [2]. Aber auch solche INS-Plattformen sind noch sehr teuer und werden daher nur in speziellen Umgebungen eingesetzt, so z.B. in der kommerziellen Luftfahrt, in Waffensystemen und der Raumfahrt.

In den letzten paar Jahren sind nun neue, sehr preisgünstige Sensoren auf den Markt gekommen. Es handelt sich hier um Sensoren, die auf der sog. Micro Electro-Mechanical System (MEMS) - Technologie basieren. Dabei werden feine Siliziumstrukturen eingesetzt, die mechanische Belastung oder Bewegungen in elektrische Signale umwandeln können. Mit dieser Technologie lassen sich somit sehr kleine Sensoren mit Standard-Chip-Gehäusen, wie sie in der Elektronik verwendet werden, realisieren. Die Entwicklung dieser Sensoren wurde v.a. von der Automobilindustrie (Airbagsysteme) und der Automations- und Unterhaltungselektronikindustrie (Bildstabilisierung bei Videokameras) vorangetrieben. Dass diese Sensoren (zur Zeit noch) nicht die gleiche Genauigkeit wie herkömmliche Sensoren aufweisen können ist klar, es wäre aber interessant zu wissen, wie genau diese Sensoren in Verbindung mit einer Strap-Down INS Plattform sind. Die Vorteile eines solchen Systems sind die niedrigen Kosten und die geringen Abmessungen, so dass gänzlich neue Anwendungsbereiche erschlossen werden können. Denkbar wären solche Systeme als Ergänzung zu einem GPS-Empfänger, um kurzfristige Satelliten-Empfangsunterbrüche zu überbrücken oder für mobile Roboter in geschlossenen Räumen.

2.2 Evaluation unseres Vorgängerprojektes

2.2.1 Übersicht

Als erste Aufgabe hatten wir unser Vorgängerprojekt in Betrieb zu nehmen, den G-Messer aus der Semesterarbeit von Marcel Lattmann [3]. Daraufhin sollten erste Messungen mit diesem Gerät durchgeführt werden, um herauszufinden, wo Verbesserungen anzubringen sind und was für unser Projekt übernommen werden kann. Ausserdem hatten wir neue Sensortypen (ADXL210, ENC03 und ENV05) zur Verfügung, deren Integrationsmöglichkeit in das bestehende Projekt überprüft werden sollte.

2.2.2 Charakterisierung des Vorgängerprojektes

Es handelt sich hier um die Semesterarbeit von Marcel Lattmann im Sommersemester 1999 [3]. Das Gerät besteht aus einem handlichen Aluminiumgehäuse mit graphischem LCD und vier Bedienungsknöpfen. Ein grosses, internes Akku-Pack sorgt für eine Betriebsdauer von mehreren Stunden. Das Prozessorboard mit einem Motorola 68332 Microcontroller, welches von A. Stiller im Wintersemester 95/96 als Teil der Diplomarbeit erstellt worden war [4], übernimmt die Messung und Datenaufbereitung. Das G-Messgerät besitzt drei Beschleunigungsmesser mit einem Bereich von +/- 50 g jedoch keine Gyroscoptes zur Rotationsmessung. Zur Evaluation installierten wir zuerst die Software-Entwicklungsumgebung und nahmen den G-Messer in Betrieb. Danach wurde die Software des G-Messers genauer untersucht und schlussendlich das Gerät vermessen.

2.2.3 Entwicklungsumgebung

Es stand uns eine Entwicklungsumgebung von HICROSS zur Verfügung. Die Installation selbst war schon problembehaftet und dazu kaum dokumentiert. Dateien waren auf dem Server in falschen Verzeichnissen, nicht vorhanden oder mit falschen Attributen versehen. Schliesslich haben wir alle Dateien in ein lokales Verzeichnis kopiert, neu sortiert und die Konfigurationsfiles angepasst. Auch die Makefiles mussten überarbeitet werden. Nach zwei kleinen Änderungen im Quellcode kompilierte die Software und auch der Debugger / Downloader konnte gestartet werden.

2.2.4 G-Messer

Bevor der Download des Binary-Files funktionierte, mussten noch die Initialisationsvektoren für den Prozessor angepasst werden. Nun konnte endlich das Programm ins RAM geschrieben und gestartet werden. Leider hat der LCD-Treiber noch Fehler, manchmal musste der G-Messer mehrmals gebootet werden, bis das LCD richtig initialisiert worden war und nicht nur Streifen anzeigte. Bevor die Beschleunigungen gemessen werden können, muss das Gerät via Menu geeicht werden. Allerdings stimmen die angezeigten Werte nicht genau. Die Messung kann nun via RS-232 auf den PC zu geladen und in einer mitgelieferten Visual Basic Applikation visualisiert werden. Das Gerät standalone zu betreiben erwies sich als unmöglich, da der Flashspeicher nicht beschrieben werden konnte. Nach umfangreichen Versuchen gelang es uns zwar, Daten ins Flash zu schreiben, allerdings nur von 0 bis 800 Hex. Über 800h stand immer nur FF, obwohl der Programmierer vorgab, auch diesen Bereich zu beschreiben und keine Fehlermeldung ausgab. Nachdem wir mehrere Berichte von abgeschlossenen Semester- und Diplomarbeiten durchgesehen und Gespräche mit den Studenten geführt hatten, welche sich

zuvor mit demselben Prozessorboard beschäftigt hatten, fanden wir heraus, dass es bis jetzt niemandem gelungen ist, das Flash zu beschreiben. Zusätzlich gaben alle Gesprächspartner der Entwicklungsumgebung sehr schlechte Noten.

2.2.5 Schlussfolgerung

Nach wir das Vorgängerprojekt 2.5 Wochen analysiert hatten, gelangten wir zur Erkenntnis, dass dieses Projekt in der uns zur Verfügung stehenden Zeit mit der vorgegebenen Hardware nicht realisierbar war. Einerseits war der Aufwand für eine Realisierung des Trägheitsnavigationssystems auf dem MC68332 Prozessor Board viel zu hoch und das gesamte System zu unflexibel, andererseits wurde uns klar, dass vom bestehenden G-Messer nur wenig übernommen werden konnte. Bei unserem Projekt ging es ja vor allem darum, herauszufinden wie genau ein Trägheitsnavigationssystem basierend auf Low-Cost Sensoren ist, wo die Quellen für die Ungenauigkeiten liegen und wieviel Rechenleistung benötigt wird. Wir wollten deshalb, mit sicher ausreichenden Ressourcen die Genauigkeit des Systems auf einfache und schnelle Weise analysieren. Nur so kann während unser kurzen Projektdauer geklärt werden, welche Genauigkeit ein solches Gerät überhaupt erreichen kann und mit welchen Trade-Offs eine vernünftige Genauigkeit realisierbar ist. Deshalb entschlossen wir uns zu einen neuen Aufbau.

3 Konzept

3.1 Übersicht

Um die verschiedenen Anforderungen wie Portabilität, Rechenleistung, Speicherkapazität und schnelle Implementation optimal zu vereinigen, erstellen wir folgendes Konzept:

- Als Rechner einen Laptop PIII 500 MHz mit Windows98
- Eine PCMCIA I/O-Messkarte zum Auslesen der analogen Sensorwerte
- Ein separates Sensor Board mit der Analog-Elektronik und einem GPS-Empfänger

3.2 Hardware

Als Laptop wird ein Dell Inspiron 7500 mit 500 MHz Prozessor verwendet. Er verfügt über genügend Rechenleistung und Speicherplatz und ist zusätzlich in der Lage, die Versorgung des kompletten Sensor Boards über die Messkarte zu übernehmen.

Als I/O-Messkarte dient eine DAQ-Card 1200 von National Instruments. Sie hat acht analoge Input- und zwei analoge Outputkanäle mit einer Auflösung von 12 Bit und 24 digitale Ein- oder Ausgänge. Zudem sind noch drei Eingänge zur Pulsbreitenmessung mit 16 Bit Auflösung vorhanden. Es können bis zu 100'000 Messpunkte pro Sekunde aufgenommen werden. Die für ein reibungsloses Zusammenspiel der Messkarte mit dem Betriebssystem unabdingbare Zwischenspeicherung der Messdaten übernimmt ein 2 kB grosses RAM auf der Karte selbst. So können kurzfristige Unterbrechungen durch das Betriebssystem ohne Messdatenverlust toleriert werden.

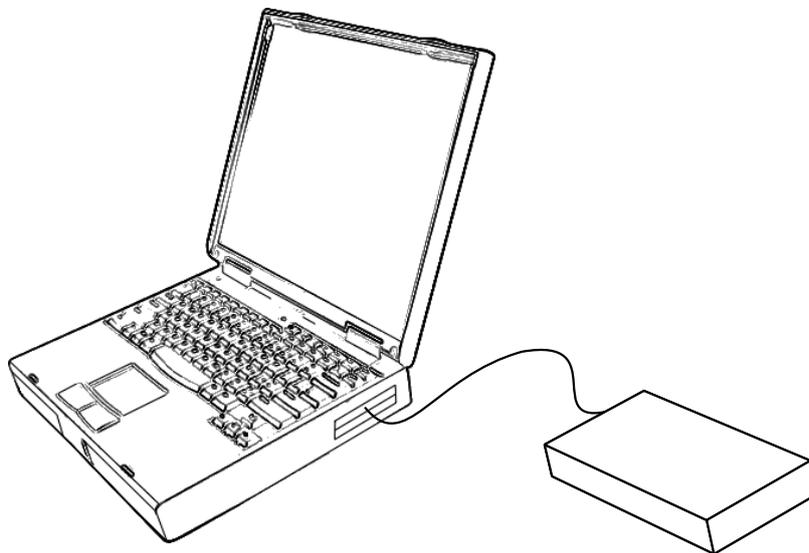


Abbildung 2: Das neue Konzept mit Laptop, PCMCIA-Messkarte und Sensor Board

Auf dem Sensor Board soll neben den Sensoren und der Hardware für die analoge Signalaufbereitung auch noch ein μ -Blox GPS Empfänger untergebracht werden. Vor der

Analog/Digital-Wandlung müssen die Signale zuerst tiefpassgefiltert werden, um Aliasing und Rauschen zu unterdrücken. Um die 12 Bit Auflösung der DAQ-Card 1200 voll auszunützen werden die Signale zudem noch verstärkt.

3.3 Software

Die Software soll in Visual Basic geschrieben werden, da die Firma National Instruments mit ihrer Messkarte einerseits sehr gute Programmierschnittstellen für diese Sprache mitliefert und sich andererseits eine Visualisierung der Resultate schnell bewerkstelligen lässt. Das Programmieren selbst in der Visual Basic Entwicklungsumgebung ist komfortabel und benötigt vergleichsweise wenig Zeit. Das neue Konzept stellt allerdings aufgrund der Ausmasse des Laptops ein limitiert portables System dar, doch für unsere Aufgabe ist dieser Faktor von untergeordneter Bedeutung. Dafür kommt der entscheidende Vorteil einer sehr hohen Flexibilität hinzu. Die Sampling-Raten von bis zu 100 kHz sind mehr als ausreichend und wir können unsere Applikation einiges schneller implementieren und testen, beliebig viele Daten speichern und wenn nötig diese in MATLAB exportieren.

4 Realisierung I: Test Board

4.1 Übersicht

Um auf dem endgültigen Board von Anfang an korrekte Analogelektronik aufzubauen, haben wir uns entschieden, zuerst ein Test Board anzufertigen. Dieses sollte für jeden Sensortypen einen Steckplatz mit dazugehörigen Filtern und Verstärkern aufweisen. Wichtig war hierbei, dass alle Komponenten, wie Widerstände, Kondensatoren, etc. steckbar waren und bei Bedarf schnell angepasst werden konnten. Die Messungen sollten mit der DAQ-Card 1200 durchgeführt, auf dem Laptop aufgezeichnet und anschliessend ausgewertet werden. Aus den Erkenntnissen dieses Aufbaus sollte danach die endgültige Version der Trägheitsnavigationsplattform entwickelt werden.

4.2 Aufbau des Test Boards

Das Board musste so aufgebaut werden, dass wir die folgenden Kenngrößen messen konnten:

- Output ENC03
- Output ENV05
- Output ADXL210 (analog & PWM)
- Output ADXL250 (analog)
- Referenzspannung vom AD780 (2.5V)
- Temperatur mittels AD780

Um alle Sensoren effizient austauschen zu können, haben wir für jeden Sensortyp kleine Platinen angefertigt, die als Footprint ein DIP14-Wide besitzen. Ein DIP14-Wide Footprint besteht aus einem DIP28 Footprint mit 15 mm Pinabstand, gekürzt auf 14 Pins.

4.2.1 Sensoren

Bei den Rotationssensoren ENC03 und ENV05 handelt es sich um 'piezoelectric vibrating gyroscopes'. Sie messen eine Drehgeschwindigkeit über die Korioliskraft, die auf drei vibrierende Piezoplättchen wirkt [2]. Um gegenseitige Störungen zwischen zwei benachbarten ENC03 zu vermeiden, gibt es den Typ A und B mit leicht unterschiedlichen Frequenzen. Ein ENC03 kostet etwa 50 SFr., ein ENV05 200 SFr.

Die Beschleunigungssensoren ADXL250 und ADXL210 von Analog Devices sind auf Siliziumbasis intergriert. Eine elastisch aufgehängte Masse wird in Funktion der Beschleunigung verschoben, diese Verschiebung wird kapazitiv gemessen [2]. Die ADXL250 und ADXL210 kosten ca. 30 SFr.

Der Bereich des A/D Wandlers sollte für eine möglichst genaue Messung vollständig ausgeschöpft werden. Wir haben deshalb für alle Sensorausgänge Bereichsanpassungen mit Verstärkerschaltungen realisiert. Die Verstärkerschaltung transformiert das Sensorausgangssignal von z.B. 1.35 +/- 0.2 V optimal auf den Messeingang von 0 - 5 V der DAQ-Card 1200. Der Arbeitspunkt des Ausgangssignals für die PCMCIA-Karte muss hierfür auf 2.5 V liegen und die Signalamplitude ebenfalls einen Wert von annähernd 2.5 V besitzen. Als Referenz für alle Transformationen dient die Referenzspannung des AD780, die auf 2.5 V festgelegt wurde.



Abbildung 3: Beschleunigungssensor ADXL210

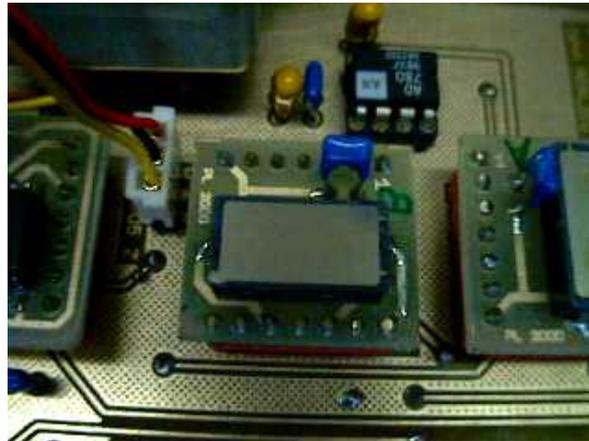


Abbildung 4: Gyroscope ENC03

4.2.2 Filter

Zur Vermeidung von Aliasing beim Messen und um das Rauschen zu minimieren haben wir am Ausgang aller Sensoren aktive Second Order Butterworth Lowpass Filter implementiert. Die Cut-Off Frequenz der einzelnen Filter und die Verstärkung der nachfolgenden Bereichsanpassung des Sensorsignals können der Tabelle 1 entnommen werden.

Beim ENC03 haben wir, wie in den Application Notes von Murata vorgeschlagen, eine AC-Kopplung zur Temperaturkompensation realisiert.



Abbildung 5: Gyroscope ENV05

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

Tabelle 1: Filter & Verstärker des Test Boards

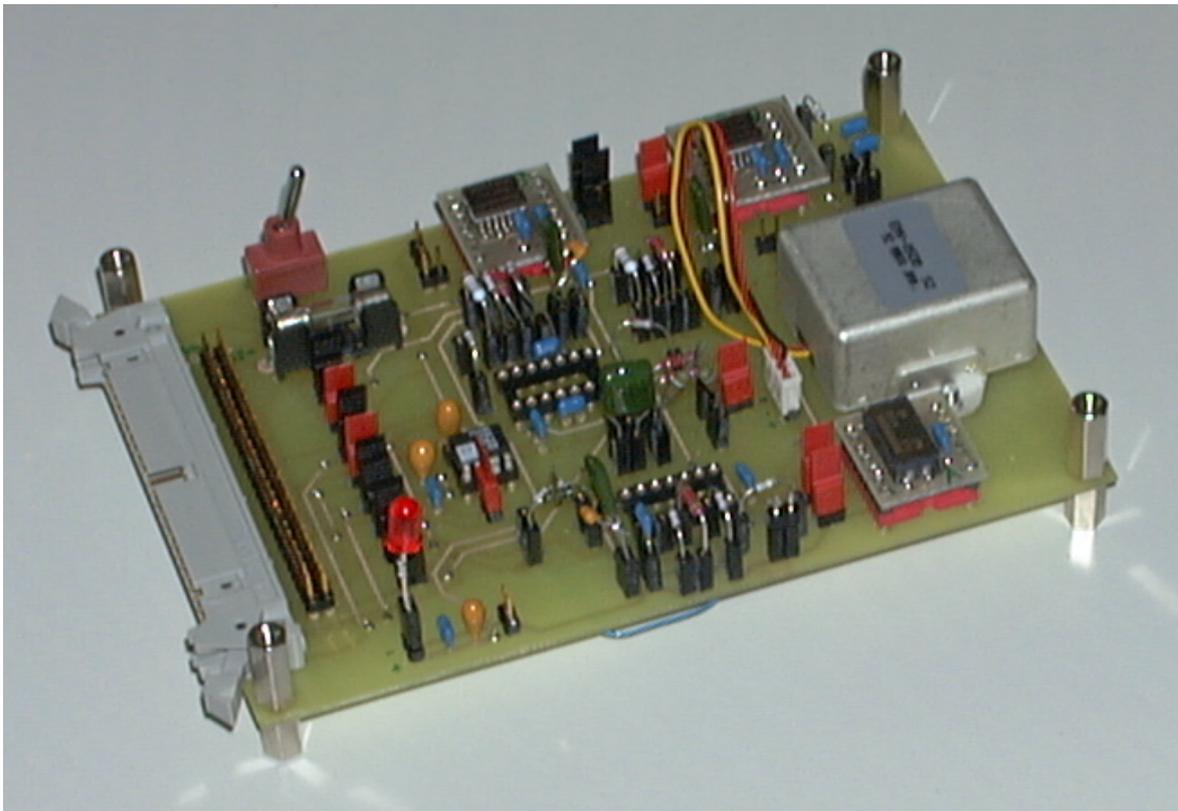


Abbildung 6: Das Test Board mit steckbaren Komponenten

4.3 Software

4.3.1 National Instruments DAQ-Card 1200

Die Inbetriebnahme der DAQ-Card 1200 von National Instruments gestaltete sich einfach. Die Einbindung in Visual Basic geschieht entweder über ActiveX-Controls oder über NI-DAQ-Funktionen. Da die vorhandenen ActiveX-Controls nur Demoversionen (Kosten der Vollversion: \$500) sind und nach 5 Minuten deaktiviert werden, wurden die NI-DAQ-Funktionen für Visual Basic benutzt. Es existieren sehr gute Beispielprogramme, die die Verwendung dieser Funktionen illustrieren und die Online-Help ist aufschlussreich. Allerdings bezieht sich diese nicht nur auf die DAQ-Card 1200, sondern auf alle NI-Produkte. Die DAQ-Card 1200 gehört zur Lab-Serie, und so muss man z.B. `Lab_ISCAN_Start()` verwenden, und nicht etwa `SCAN.Start()`.

Es wird kontinuierlich gemessen, da die genaue Zeitdauer im vornherein noch nicht bekannt ist. Dazu wird der Double-Buffered-Mode verwendet (siehe `DAQ_DB_Config()` in der Online Help). Die DAQ-Card 1200 füllt zunächst die eine Hälfte des Kartenbuffers mit Daten. Dieser Buffer kann ausgelesen werden, sobald er voll ist. Währenddessen wird die zweite Hälfte des Buffers beschrieben.

4.3.2 Sampling Parameter

Etwas heikel sind die Timing-Einstellungen beim Aufsetzen des Samplings. Dazu empfiehlt sich, die Kriterien bei `DAQ_DB_Config()` und `Lab_ISCAN_Start()` genau zu studieren. Hier ein Beispiel aus unserem Code für eine Samplingfrequenz von 150 Hz:

```

' Timebase, SampleInterval und ScanInterval berechnen
' iStatus% = DAQ_Rate(S_Freq, iUnits%, S_sampTimebase, S_sampInterval)
' Bedingungen:
'   - Zeit pro Messung >= 10us (bei Gain < 10) S_sampInterval/realTimebase
'   - realTimebase  sampTimebase
'       100 Hz      5
'       1k Hz      4
'       10 kHz     3
'       100kHz    2
'       1MHz      1
'   - scanInterval < 65535, > 2 und > NrCh / realTimebase + 5 us
'   - SampleRate (Hz) = realTimebase / scanInterval

' Fuer 150 Hz:
  S_sampTimebase = 1
  S_sampInterval = 10
  realTimebase = 1000000
  S_scanInterval = 6667

```

Die `S_sampTimebase` des karteneigenen Timers ist also 1 us (1 MHz), die Zeit zwischen zwei Messwerten zweier benachbarter Kanäle ist $10 * S_sampTimebase$, also 10 us (= minimale Zeit). Die Zeit zwischen zwei kompletten Scans durch alle Kanäle beträgt $6667 * S_sampTimebase = 1/150$, entspricht also 150 Hz.

4.3.3 Das Programm 'MajorScan'

Das Programm MajorScan lässt das Aufzeichnen beliebig vieler Kanäle mit folgenden Frequenzen zu: 1, 10, 50, 150, 300, 500, 2000 Hz. Die gemessenen Werte in Ticks werden in einem Array abgespeichert, welches nach Beenden der Messung im Binärformat (kleiner, schneller) oder in Asciiformat (lesbar) zusammen mit den Samplingparametern (Frequenz, Gain etc.) abgespeichert werden kann. Es liegen auch Importroutinen für MATLAB vor, um die Daten dort weiterzuverarbeiten.

4.4 Messungen der Sensoren

Aus unseren Messungen wollten wir folgende Aussagen gewinnen:

- Vergleichsdaten zur Evaluation der geeignetsten Sensoren (ADXL210 vs. ADXL250 / ENC03 vs. ENV05)
- Daten zur Erstellung der Sensormodelle (Kennlinien)
- Daten zur Korrekturalgorithmen
- Verifikation der analogen Filter und Verstärker (Cut-Off Frequenz & optimale Signalanpassung)

4.4.1 Evaluation der Sensoren

Um die Sensoren vergleichen zu können, ist für jeden Sensortyp je ein Steckplatz auf dem Test Board vorhanden. Somit konnten die verschiedenen Sensoroutputs innerhalb einer Messung direkt miteinander verglichen werden.

4.4.2 Sensormodelle

Hier kann man beliebig viele Parameter berücksichtigen, wie z.B. Temperaturdrift, VDD-Drift, Vibrationen, Beschleunigungs- und Winkelgeschwindigkeits-Kennlinien, Hysteresen, Rauschen, Alinierungsfehler der Dies innerhalb des Packages etc. Für die meisten dieser Parameter benötigt man allerdings aufwändige Einrichtungen wie Drehteller, Beschleunigungsschlitten usw. Wir beschränkten uns deshalb gezwungenermaßen auf die für uns möglichen Messungen: einfache Beschleunigungsmessungen mit Hilfe der Erdbeschleunigung und Temperatur-Biasdrift-Messungen mit einem Klimaschrank.

4.4.3 Temperaturmessungen

Die Temperaturmessungen wurden in einem Klimaschrank durchgeführt, der gekühlt und geheizt werden konnte. Um Kondenswasserbildung auf der Elektronik während der Messung zu vermeiden, sah das Temperaturprofil folgendermaßen aus: 60, 50, 40, 30, 20, 10 Grad Celsius, jeweils während 20 Minuten. Die Nullwerte (unbeschleunigt, nicht bewegt) wurden im Sekundentakt mit der Applikation MajorScan gemessen und aufgezeichnet. Danach sowohl als *.bin(binary) als auch als *.dat(ascii) abgespeichert. In MATLAB wurden dann die gewonnenen Daten ausgewertet. (Siehe Anhang, resp. CD) Es wurden alle Sensoren mehrfach vermessen, um auch eine Aussage über die Reproduzierbarkeit machen zu können.

4.4.4 Beschleunigungsmessungen

Als Referenzbeschleunigung wurden $+1g$, $0g$ und $-1g$ der Gravitation benützt. Damit sind die Daten komplett und ein einfaches, lineares Sensormodell in folgender Form lässt sich ableiten

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

wobei α die Beschleunigung ist, $Sens$ die gemessene Sensitivität und b die temperaturabhängige Biasspannung als Funktion der Temperatur.

4.4.5 Algorithmen

Hier gab es nicht viel auszumessen. Es ging einzig darum, ob es eine Phasenverschiebung zwischen dem Eintreffen der Signale der einzelnen Sensoren gibt, d.h. ob die einzelnen Sensoren unterschiedliche Antwortzeiten aufweisen. Um dies herauszufinden hängten wir das Test Board an einem Faden auf, und verpassten ihm dann mit einem Hammer einen Schock in Form eines Schlags. Dabei wurden die Werte mit 2000 Hz aufgezeichnet und in MATLAB die gewonnenen Daten ausgewertet. Wir konnten keine Phasenverschiebung der Impuls-Antwort feststellen. Zumindest ist die Verschiebung so klein, dass sie in unserem System keine oder zumindest eine untergeordnete Rolle spielt. Wir beschlossen deshalb diesen Einfluss zu vernachlässigen.

4.4.6 Filter

Bei den Filtern ging es darum, die Zweckmässigkeit und die berechneten Frequenzgänge zu überprüfen. Für die Frequenzgänge der Second Order Butterworth Tiefpassfilter legten wir mit einem Signalgenerator sinusförmige Signale an und visualisierten die resultierende Ausgangsamplitude mit einem Oszilloskop (manueller Sweep). Die Übereinstimmung mit den berechneten und simulierten Werten war verblüffend gut. Wir entschieden uns, die Filter für das nächste Board so zu übernehmen.

Der im Datenblatt von Murata vorgeschlagene analoge Hochpass für den ENC03 (zur Temperaturkompensation) mit der sehr tiefen Grenzfrequenz von 0.3 Hz hatte allerdings verzerrte Amplituden- und Phasenverläufe zur Folge. Die Antwort auf eine (möglichst) gleichmässige Drehung um 90 Grad entsprach allem anderen als dem erwarteten Rechtecksignal. Nach genauerer Analyse des Signals mit MATLAB mussten wir feststellen, dass sich diese Filterschaltung unmöglich für unsere Zwecke eignete. Die Temperatur-Bias-Kompensierung musste demzufolge auf eine andere Art und Weise realisiert werden. Wir hatten deshalb anschliessend eine DC-Kopplung des Sensors mit einer Filter- und Verstärkerstufe realisiert, die das Signal optimal an den A/D-Wandler-Eingang anpasste. Hier war nun keine Verzerrung des Signals mehr zu erkennen.

Hingegen war die gemessene Temperaturdrift des Gyros ENC03 kleiner als im Datenblatt angegeben. Somit ist eine Temperaturkompensation erst nach dem A/D Wandler in digitaler Form ebenfalls möglich.

5 Realisierung II: Sensor Board

5.1 Übersicht

Mit den Erkenntnissen, die wir aus dem Aufbau des ersten Boards gewonnen hatten, entwickelten wir das zweite Board, unser sog. Sensor Board. Dieses Board sollte Beschleunigungs- und Rotationssensoren aufweisen, die alle Freiheitsgrade abdecken und somit eine Bewegungs- und Lagebestimmung im Raum ermöglichen. Daraus lässt sich dann die Position im Raum errechnen. Wir verwenden drei ADXL210 Beschleunigungssensoren in X,Y und Z Richtung, zwei Gyros ENC03 für die Messung der Rotation um die X- und Y-Achse und einen Gyro ENV05 zum Messen der Drehung um die Z-Achse. Um Vergleichsmessungen des Trägheitsnavigationssystem mit einer Referenzquelle durchführen zu können, wurde ein GPS Empfänger eingebaut, welches seine Daten über den eingebauten Pegelwandler MAX232 auf die serielle Schnittstelle ausgibt.

5.2 Die Hardware des Sensor Boards

5.2.1 Spannungsversorgung

Das Gerät kann direkt von der PCMCIA-Messkarte mit Strom versorgt werden. Die Karte bietet hierfür einen 5 V Ausgang, der mit maximal 500 mA belastet werden kann. Andererseits haben wir auf der Platine unseres Gerätes einen Anschlussstecker (CON7_1) für eine externe Speisung von 5 V vorgesehen, damit das System auch in Verbindung mit einem DSP-Board mit eingebautem A/D Wandler betrieben werden kann. Mit diesem Anschluss besteht die Möglichkeit, genügend Strom für zusätzliche Komponenten des Systems zu liefern ohne dass die Versorgungsspannung einbricht. Es ist darauf zu achten, dass eine stabile Spannungsquelle mit genau 5 V angeschlossen wird, da die Elektronik und speziell die Sensoren auf diese Spannung ausgelegt worden sind. Abweichungen davon werden sich vor allem in verminderter Sensitivität der Sensorausgabesignale äussern. Zwischen diesen beiden Betriebsarten kann mittels der Jumper J7_1 und J8_1 umgeschaltet werden. Um bei Kurzschlüssen die Messkarte vor allfälligen Überbelastungen zu schützen ist eine Sicherung von 500 mA eingebaut worden.

5.2.2 GPS & MAX232

Das GPS und der Lüfter können wahlweise abgeschaltet werden. Die Abschaltung des GPS und des zugehörigen MAX232 Pegelwandlers geschieht über den Jumper J1_1. Um allfälligen Komplikationen mit Brummschleifen oder kleinen Potenzialunterschieden aufgrund einer zweifach vorhandenen Ground-Leitung vorzubeugen, haben wir zusätzlich einen Jumper Gnd_J1_16 eingebaut. Dieser Jumper hat die Aufgabe die Masseverbindung vom Trägheitsnavigationssystem über die RS232-Verbindung zum Laptop zu unterbrechen, wenn das Massepotenzial schon über das Verbindungskabel der PCMCIA-Messkarte durch die Spannungsversorgung definiert worden ist.

Der Datenaustausch des GPS mit dem Laptop erfolgt über den seriellen Port A des GPS. Dieser Port dient auch für allfällige Änderungen der Konfigurationseinstellungen des GPS. Es kann die μ -Blox Demosoftware verwendet werden, um Konfigurationseinstellungen zu ändern oder die Funktion des GPS zu verifizieren. Das GPS verfügt ausserdem über eine zweiten seriellen Port B für ein DGPS System. Dieser Port wurde ebenfalls über den MAX232 geführt und ist in Form von Stiftanschlüssen auf der Hauptplatine vorhanden. Die RS232 Eingangsleitung

(RS232 B OUT), die dem GPS Daten liefert, muss bei Nichtgebrauch gemäss Datenblatt auf Masse gelegt werden. Es ist dafür der Jumper J1.16 vorgesehen.

5.2.3 Temperaturkompensation

Um die Temperatur aller Sensoren mit einem einzigen Temperatursensor messen zu können, haben wir die Sensoren in eine Art Strömungskanal eingebettet, an dessen Eingang ein kleiner Lüfter für einen kontinuierlichen Luftstrom sorgt.

Wir wissen, dass unsere Plattform sehr empfindlich auf Temperaturschwankungen reagiert, mit dem Strömungskanal wird diese Eigenschaft sogar noch verstärkt und das Gerät auf den ersten Blick offensichtlich verschlechtert. Doch mit dieser Massnahme besitzen alle Sensoren dieselbe Temperatur, die Temperaturmessung des AD780 macht so erst einen Sinn und die Kompensation wird überhaupt ermöglicht. Damit aber die Sensoren durch starke Temperaturschwankungen nicht erheblich gestört werden, muss das komplette System thermisch träge sein. Dies wiederum kann mit guter äusserer Isolation des Gerätes oder mittels Platzieren an einem Ort mit gemässigtem Temperaturverlauf geschehen.

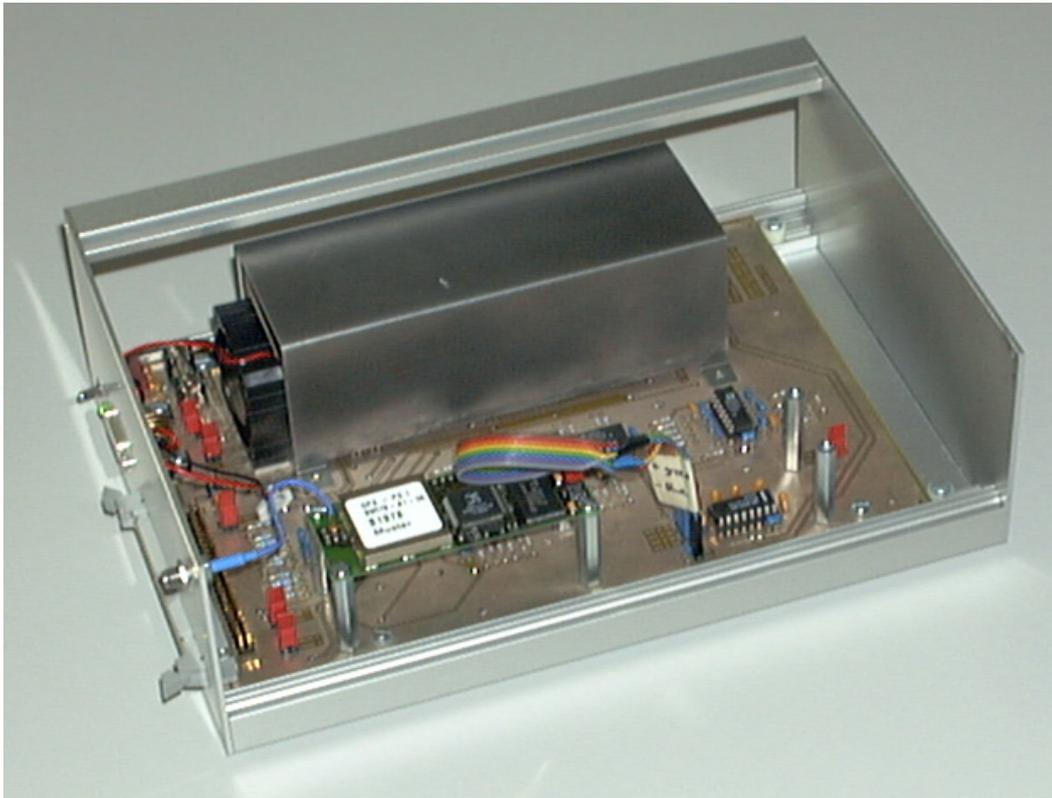


Abbildung 7: Das Sensor Board mit dem Strömungskanal und dem Lüfter

5.2.4 Filter & Verstärker

Wir haben die Filter Cut-Off Frequenzen und die Verstärkungen der einzelnen Sensorausgangsbeschaltungen weitgehend von der ersten Platine übernommen. Der Temperatursen-

sor AD780 erhielt am Temperatureausgang zwecks besserer Auflösung eine Verstärkerschaltung mit $A = 7.8$. Die ursprüngliche AC-Kopplung des ENC03 wurde entfernt, da sie das Signal verzerrte. Sie wurde durch eine reine DC-Kopplung mit nachfolgender Pegelanpassung ersetzt.

Da wir aus Tests mit dem ersten Board herausgefunden hatten, dass die Gyros ENC03 recht ungenau sind, haben wir parallel zu den Gyros ENC03 den Einbau von präziseren ENV05 vorgesehen. So können die Gyros ENC03 einfach mit solchen vom Typ ENV05 ersetzt werden. Die hierfür benötigten Filter und Verstärker sind schon auf dem Board vorhanden, es müssen lediglich die entsprechenden Jumper (J2_1 / J3_1 für die X-Achse, J4_1 / J5_1 für die Y-Achse) gesetzt und die Eingänge der nicht mehr benötigten ENC03 Filter auf GND geschaltet werden, damit diese nicht zu schwingen beginnen. Die Terminierung der Filtereingänge geschieht beim ENC03 direkt auf dem Sockel (hierfür muss vom GND-Pin eine Drahtbrücke nach OUT X bzw. OUT Y eingefügt werden). Bei den ENV05 bietet sich hierfür gerade der Stecker an, der mit einem Jumper das Eingangssignal der Filterschaltung auf Masse legen kann.

Sensor	Filter F_c (-3dB) [Hz]	Verstärkung $ A $
ADXL210	50	2.2
ENC03	50	10
ENV05	25	1.2
AD780	–	7.8

Tabelle 2: Filter & Verstärker des Sensor Boards

Ansonsten wurden die Schaltungen beibehalten. Bei den Filtertypen handelt es sich nach wie vor um Butterworth-Filter 2. Ordnung. Die Tabelle 2 bietet einen Überblick über die Filterkoeffizienten und die einzelnen Verstärkungen.

5.3 Software zum Sensor Board

Es existieren zwei Programme zu diesem Board. Zum einen das 'Cockpit', welches die Lage im Raum mittels künstlichem Horizont und Kreiselkompass grafisch darstellt, angelehnt an ein Flugzeugcockpit. Dieses Programm wurde v.a. zum Debuggen und zu Demonstrationszwecken gebraucht. Zum anderen haben wir das eigentliche Navigationsprogramm 'Espresso' geschrieben. Hier sind Funktionen oder zumindest Funktionskontainer vorhanden, die alle wichtigen Aufgaben einer Navigation übernehmen, wie z.B. Initialisation, alle Koordinatentransformationen, Rückführungen, Kompensationen, etc. In diesem Kapitel sollen diese beiden Programme und deren Struktur genauer erläutert werden.

5.3.1 Demoprogramm 'Cockpit'

Es werden hier im wesentlichen die Drehgeschwindigkeiten gemessen, in Komponenten von den drei Eulerwinkeln zerlegt und integriert. Für genauere Erklärungen zur Transformation siehe Kapitel 6.

Weil nur die Gyros verwendet werden, welche lediglich eine Nullung der aktuellen Offsetspannung erfordern, konnte die Initialisation sehr einfach gehalten werden: Während der Zeitdauer, in der die Checkbox 'Null' aktiviert ist, werden die Mittelwerte der gemessenen Daten gebildet und wir erhalten unsere Offsetwerte. Je länger wir den Mittelwert bilden, desto genauer sollte er werden.

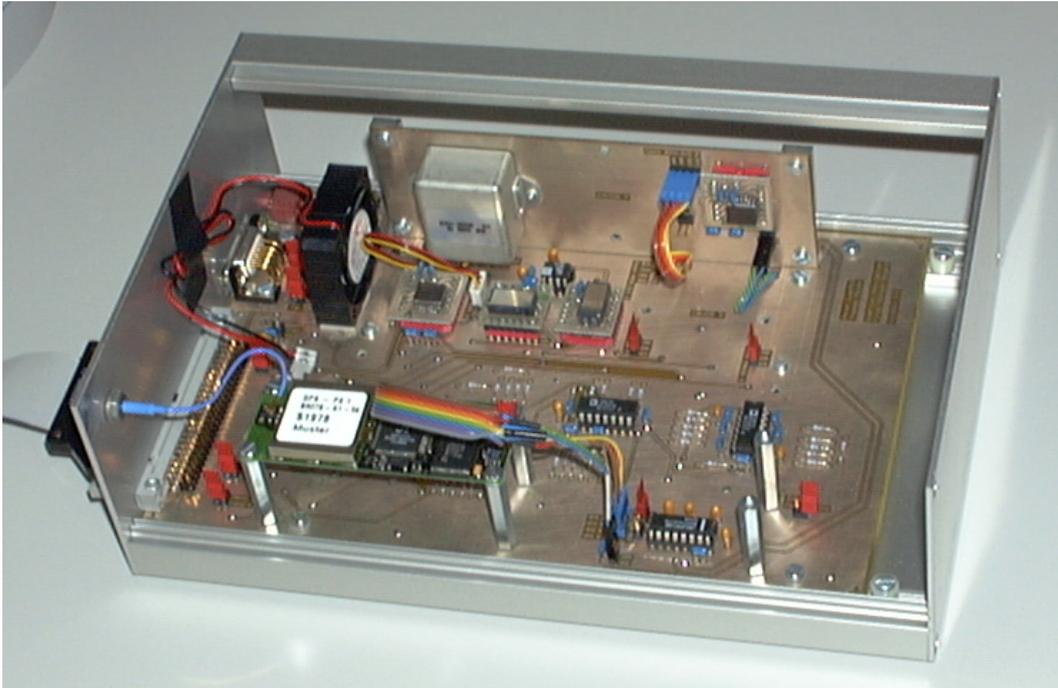


Abbildung 8: Das Sensor Board mit den einzelnen Sensoren

Mit dem Button 'Initialize' werden die Werte der Eulerwinkel und der Rückführung auf ihre Ursprungswerte zurückgesetzt.

Mit dem Button 'Feedback' kann man die Rückführung ein- bzw. ausschalten. α und β (Bank und Pitch) richten sich nach dem mit den Beschleunigungssensoren bestimmten Lot aus. Die Funktionen `f_reg_omega()` und `f_reg_phi()` bestimmen die Drehgeschwindigkeits- resp. die Winkelpositionsregelung. Bei unserer Demo wird der Fehler zwischen Winkelposition und gemessenem g -Vektor mit konstanter Geschwindigkeit (`c_reg_phi`) auf null reduziert. Die dazu entsprechende Drehgeschwindigkeit muss dann rückwirkend angepasst werden (`c_reg_omega`). Die Richtung des virtuellen Kompasses kann mangels Referenz nicht geregelt werden. Für weitere Erklärungen siehe Kapitel 6.

5.3.2 Navigationsprogramm 'Espresso'

'Espresso' bietet wesentlich mehr Funktionalität als 'MajorScan'. Es baut auf einer übersichtlichen Struktur auf, die einfach erweitert werden kann. Alle funktionellen Gruppen wurden in eigenen Funktionen zusammengefasst, so dass man Änderungen in Teilbereichen machen und sie sofort ausprobieren kann. Zur Erläuterung der Softwarestruktur befindet sich im Anhang ein Datenflussdiagramm.

Die von der DAQ-Card 1200 gemessenen Werte kommen als Record mit den Feldern (`ax, ay, az, wx, wy, wz, T`) zur Funktion `f_handle_Buf()`. Diese sorgt nun dafür, dass alle nachfolgenden Funktionen in der richtigen Reihenfolge aufgerufen werden und die Daten in die globale Positionsstruktur `AktPos` geschrieben werden. Die rohen Messdaten in Ticks werden von den entsprechenden Temperaturkompensationsfunktionen der einzelnen Sensoren

anhand der aktuellen Temperatur kompensiert. Dann muss der NullOffset subtrahiert werden und die Ticks in $m/(s^2)$ umgerechnet werden. Die Beschleunigungen übernimmt nun `f_koord_transform_trans()` und die Drehgeschwindigkeiten `f_koord_transform_rot()`. Die einzelnen Komponenten der Beschleunigungen benötigen wir vollständig im Ursprungskoordina-tensystem, während die Rotationsgeschwindigkeiten in Komponenten der Eulerwinkel abge-bildet werden müssen. Deshalb sind in der Software zwei unterschiedliche Transformations-funktionen vorhanden. Jetzt folgen nacheinander eine Regelung(`f_reg_accel()/f_reg_omega()`), eine Integration(`f_int_accel()/f_int_omega()`) und eine Regelung (`f_reg_speed()/f_reg_pos()`). Bei den Beschleunigungen folgt nochmals eine Integration (`f_int_speed()`) und eine Rege-lung(`f_reg_pos()`), um von der Geschwindigkeit auf die zurückgelegte Strecke schliessen zu können. Nun haben wir die Position neu ausgerechnet und sind fertig mit den Berechnungen. Für jeden gemessenen und aufgezeichneten Record wird dieselbe Prozedur wiederholt.

Die Transformationen und Integrationen sind eigentlich 'statisch', d.h. einmal implemen-tiert, müssen sie kaum mehr geändert werden. Hingegen sind die anderen Funktionen frei zugänglich für Veränderungen und Erweiterungen. So könnte man z.B. mit `f_reg_pos()` eine Positionsrückführung realisieren, allerdings hat diese Regelung auch Einfluss auf die aktuelle Geschwindigkeit und Beschleunigung, welche dann auch nachgeführt werden müssten.

5.3.3 Aufzeichnen der Daten

Die rohen Daten in Ticks werden komplett in ein Array gespeichert und sind, wie in 'MajorS-can' auf Disk speicherbar. So kann man, z.B. in MATLAB, die Navigation nachvollziehen, resp. die Algorithmen 'off-line' optimieren.

6 Realisierung III: Algorithmen

In diesem Kapitel werden die verwendeten Algorithmen erklärt und allenfalls Literaturverweise gemacht. Folgende Probleme stellten sich uns in diesen Bereichen:

- Repräsentation der Position und Orientierung im Raum (Winkelposition)
- Transformation der Beschleunigungen und Winkelgeschwindigkeiten in oben definiertes Koordinatensystem
- Korrekte Initialisation vor Beginn der Messung
- Regelungen / Rückführungen
- Temperatur-Kompensation der Sensoren

6.1 Repräsentation im Raum

Es geht hier im wesentlichen um die Wahl eines geeigneten Koordinatensystems, wobei alle sechs Freiheitsgrade korrekt repräsentiert werden müssen. Im allgemeinen sind dies drei translatorische Koordinaten und drei rotatorische Koordinaten. Allerdings beschreiben z.B. die drei Winkel um die Achsen eines karthesischen Koordinatensystems die *Lage* des Objektes im Raum *nicht vollständig*. Es gibt aber viele Arten von Koordinatensystemen, die diese Beschreibung vollständig liefern, jeweils mit ihren Vor- und Nachteilen [5].

6.1.1 Translatorische Koordinaten

Für die Positionskoordinaten werden in der Navigation meist Längen- und Breitengrade verwendet. Dies macht für uns keinen Sinn, da wir uns nur auf sehr beschränktem Raum bewegen, wo die Erdkrümmung keine Rolle spielt. Deshalb entschieden wir uns für ein einfaches karthesisches Koordinatensystem als Referenz. Zur Zeit null schaut die x-Achse rechts zur Seite, die y-Achse nach vorn und die z-Achse nach oben, also ein Rechtssystem.

6.1.2 Rotatorische Koordinaten

Die Orientierung im Raum wird üblicherweise durch eine der drei folgenden Arten repräsentiert: *'Direction Cosines'*, *Eulerwinkel* oder *'Quaternions'*.

Die *'Direction Cosines'* - Methode besteht aus einer 3x3 Matrix mit den Cosinuswerten der Richtungen zu den Achsen. Man kann damit einfach Transformationen durchführen, allerdings ist die Darstellung redundant und keineswegs intuitiv.

Schon besser steht es da bei der Darstellung mittels *Eulerwinkeln*. Mit drei geschickt gewählten Winkeln im Raum ist die Orientierung gegeben. Dies ist sehr intuitiv und effizient, hat aber den Nachteil, dass an zwei Punkten eine Art Singularität existiert (Nord- und Südpol); dort ist die Darstellung nicht mehr eindeutig. Auch sind zunehmend Rechenfehler zu erwarten, wenn man sich diesen Punkten nähert.

Die dritte Methode mit den *'Quaternions'* beschreibt eine Orientierung durch einen 3D-Vektor und eine Drehung um denselben. Somit ergibt sich ein Vektor mit vier Elementen. Hier existiert zwar keine Singularität, aber die Darstellung ist ebenfalls nicht intuitiv.

Wir wählten schlussendlich eine Repräsentation mit Eulerwinkeln, da diese bei allen Methoden für eine visuelle Darstellung sowieso errechnet werden müssen und der Rechenaufwand

hier somit einiges kleiner ausfällt. Die Winkel wurden folgendermassen definiert: α als Drehwinkel um die Längsachse (y'' -Achse), β um die Querachse (x' -Achse) und γ um die Hochachse (z -Achse). Dies ergibt eine recht einfache Transformation, da zwei Sensoren 'beinahe' direkt um ihre Achsen drehen. Die z -Achse ist diejenige unseres kartesischen Ursprungskoordinatensystems. Die x' -Achse ist gegenüber der Ursprungs- x -Achse um den Winkel γ verdreht und die y'' -Achse steht auf der x' -Achse senkrecht, in Richtung der Geräteachse. Man könnte die drei Winkel auch intuitiv als Bank (α), Pitch (β) und Direction (γ) bezeichnen. Und dies sind genau die Werte, welche im 'Cockpit' mit dem künstlichen Horizont und dem Kompass grafisch angezeigt werden.

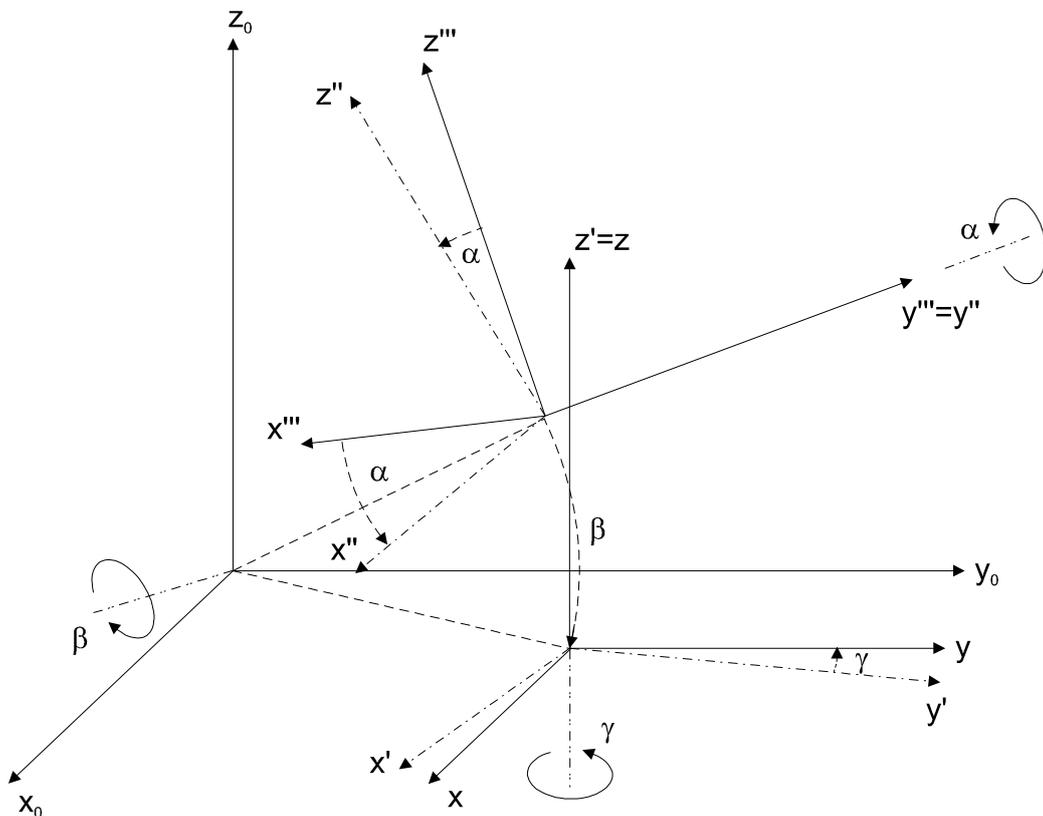


Abbildung 9: Repräsentation der Lage im Raum mittels Eulerwinkel

6.2 Transformation der Messwerte in unser Koordinatensystem

Die Beschleunigungen in Richtung der Gerätekoordinatenachsen (x''',y''',z''') werden um die Winkel α , β und γ (in dieser Reihenfolge) gedreht. Die Transformationsmatrix sieht folgendermassen aus:

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

Die Drehgeschwindigkeiten um die Gerätekoordinatenachsen werden in 'Eulerwinkelgeschwindigkeiten' umgerechnet:

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

Es gibt noch folgendes zu beachten: Die *Reihenfolge* der Rotationen spielt bei grösseren Winkeln eine wesentliche Rolle ! Als Beispiel: Es macht einen Unterschied, ob man in der Nähe des Nordpol zuerst nach Süden geht und dann nach Westen, oder zuerst nach Westen und danach nach Süden. Dieser Fehler kann jedoch bei kleinen Winkeln vernachlässigt werden. Das Intervall zwischen zwei Messungen muss also im Verhältnis zur maximalen Drehgeschwindigkeit klein sein.

6.3 Regelung / Rückführung

Unter das Thema Rückführung fallen z.B. Positionsupdate mit GPS, Ausrichtung der Lage nach dem g-Vektor, einem Kompass oder Tiltsensoren, statistische Methoden, etc [6]. Mit vielen Rückführungen schränkt man jedoch den Einsatzbereich ein. So nimmt man z.B. bei einer Ausrichtung nach dem g-Vektor an, dass man sich in einem Schwerfeld befindet und generell unbeschleunigt ist. Dies trifft auch meistens zu, allerdings auf einer Erdumlaufbahn (Schwereelosigkeit) oder einem Karussell (Zentripetalbeschleunigung) nicht. Kompliziert wird eine Rückführung dadurch, dass man nicht einfach einen durch Integration bestimmten Wert verändern kann, ohne die Veränderungen an den differentiellen Wert zurückzugeben. So kann

man nicht die Position mit dem GPS updaten, ohne die Geschwindigkeit ebenfalls zu korrigieren. Im Falle eines GPS hat man ja auch gerade noch eine Geschwindigkeit, was aber bei einer Rückführung nach dem g-Vektor? Die Drehgeschwindigkeit driftet langsam davon, die Lage wird immer wieder nachgeregelt, doch die Zunahme der Lagedrift wird ebenfalls immer grösser, bis der Lageregler schliesslich nicht mehr nach kommt. Um dies zu vermeiden, muss die Drift der Drehgeschwindigkeit ebenfalls nach null zurückgeregelt werden. Genau dies wird bei unserer Lage-Rückführung in 'Cockpit' und 'Espresso' durchgeführt.

Ein weiteres Problem besteht, wenn ich z.B. die Drehgeschwindigkeiten nach gewissen Gesichtspunkten alleine regeln will, etwa mit Hilfe eines Tiefpasses. Da danach integriert wird, besteht die Gefahr, dass die daraus resultierende Lage zu schwingen beginnt.

6.4 Initialisation

Bei der Initialisation sollen verschiedene Werte bestimmt werden, wie z.B. Geschwindigkeit, Position und Lage. Zudem soll noch eine Nullung der Sensoren erfolgen. Während die Nullung sich auf Veränderungen auf dem Sensor Board beziehen, sind die anderen Werte *extern* zu bestimmen. Extern heisst hier, sie entweder mit der notwendigen Genauigkeit zu messen, oder implizit durch Vorschriften bei der Initialisation dem System zuzuführen [6].

Für unseren Versuchsaufbau sind die globalen Positionskoordinaten nicht relevant, deshalb werden sie bei der Initialisation auf null gesetzt ($x = 0$, $y = 0$, $z = 0$). Man könnte die Anfangsposition aber auch von einem GPS-Empfänger erhalten. Die anfängliche Ausrichtung des Sensor Boards können wir nicht messen, da weder Tiltensoren noch eine Kompass vorhanden sind. Deshalb schreiben wir vor, dass das Board *eben ausgerichtet* und ruhig stehen muss und definieren die Gerätelängsachse (also die y-Achse) als Norden. (siehe 'Espresso': Sub Start_Initialize())

6.5 Temperaturkompensation

In diesem Abschnitt werden Algorithmen zur Kompensation des Biasdriftes in Abhängigkeit der Temperatur vorgeschlagen. Die dazu verwendeten Messdaten sind als *.bin oder *.dat Files oder als Matlab *.fig Graphik auf der beiliegenden CD vorhanden. Der Biasdrift gegenüber der Temperatur wurde mit einem Polynom möglichst kleinen Grades angenähert.

Der ADXL210 ist sehr temperaturstabil, über den Bereich von 60 Grad Celsius variiert er nur um etwa 5 Ticks, dies entspricht ungefähr 6 mV oder 0.12 %. Die Resultate sind gut reproduzierbar und die individuellen Sensoren zeigten alle dasselbe Verhalten. Die Temperaturabhängigkeit ist darüberhinaus einfach linearisierbar. Als T_{ref} bezeichnen wir die Temperatur, welche beim Initialisieren der Gerätes gemessen wurde. Hier unsere verwendete Kompensationsfunktion des ADXL210:

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

Der ENC03 ist nicht sehr temperaturstabil. Einerseits variiert er statisch (also nach dem Einschwingen der Temperatur) um etwa 150 Ticks oder 180 mV bzw. 3.6 % über einen Bereich von 60 Grad Celsius. Andererseits besitzt er noch eine dynamische Komponente, d.h. der Bias ist auch abhängig vom *Differential der Temperatur*. Dabei ändert sich die Grösse des Einflusses abhängig davon, wie schnell sich die Temperatur verändert. Diese Komponente ist je nach Änderungsgeschwindigkeit um einiges grösser als der statische Anteil. Wir haben

keine gute Kompensationsfunktion zur Lösung dieses Problems gefunden, zumal der Klimaschrank keine stabile Temperatur halten konnte, sondern mit seinem Zweipunktregler eine Temperaturkurve erzeugte, die etwa um drei Grad auf- und abpendelte.

Der ENV05 hat ein viel besseres Temperaturverhalten als der ENC03. Er driftet nur etwa um 30 Ticks, 36 mV oder 0.7 % über einen Bereich von 60 Grad Celsius. Dieser qualitative Unterschied äussert sich natürlich im Preis. Er reagiert aber ebenfalls auf das Differential der Temperatur, allerdings nur wenig, sicher auch in Folge der viel grösseren Masse. Genau diese Masse muss aber auch bei der Temperaturkompensation berücksichtigt werden. Eine Temperaturänderung wird im Sensor selbst und somit auch der Einfluss auf das Ausgangssignal nur langsam wirksam. Die besten Ergebnisse erreicht man, wenn man die Temperaturen etwa 400 Sekunden lang beibehält und den Mittelwert bildet. Danach lässt sich folgende Kompensationsfunktion anwenden:

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

Die Reproduzierbarkeit der Sensordaten ist ziemlich gut. Wir hatten aber nur einen einzigen ENV05 zur Verfügung, und so können wir keine Aussage machen, ob alle ENV05 durch eine solche Funktion kompensiert werden können.

Zum Schluss noch eine Bemerkung: Im allgemeinen verändert sich die Temperatur nicht allzu schnell. Unsere Navigationsplattform ohne Rückführung ist aber nur recht kurze Zeit stabil. Man muss die Wichtigkeit der Temperaturkompensationsfunktionen in unserem System daher relativieren, da während der kurzen Zeit, in der plausible Daten zur Verfügung stehen, die Temperatur der Sensoren in guter Näherung als konstant angenommen werden kann.

7 Resultate

7.1 Ausmessen des fertigen Systems

Das Vermessen des Systems war eine der Hauptaufgaben, die wir zu erfüllen hatten. Es sollte festgestellt werden, wie genau es ist, wo die Quellen der Ungenauigkeiten zu finden sind und mit welchen Massnahmen eine höhere Genauigkeit zu erreichen wäre.



Abbildung 10: Das fertige System mit Laptop, Sensor Board und PCMCIA-Messkarte

7.1.1 Elektrische Verifikation

Die elektrische Verifikation bestätigte die eigentliche Fehlerfreiheit des Sensor Boardes. Zwei Probleme mussten allerdings erkannt und behoben werden: Die Filter schwingten und die Versorgungsspannung sank bei zunehmendem Stromverbrauch zu stark.

7.1.2 Schwingende Filter

Der Grund für das Schwingen der Filter lag bei den neuen Operationsverstärkern vom Typ LM6134. Sie haben ein Gain-Bandwidth von 10 MHz und eine Slew-Rate von $12 \text{ V}/\mu\text{s}$. Nach Ersetzen durch die Op-Amps vom Typ OP491 vom Test Board funktionierten die Filter wieder optimal, da diese eine Gain-Bandwidth von lediglich 3 MHz und eine Slew-Rate von $0.5 \text{ V}/\mu\text{s}$

aufweisen. Das Schwingen trat nur bei kaskadierten Operationsverstärkern auf, also z.B. bei der Schaltung des ENC03, wo dem Filter noch eine Verstärkerstufe folgt.

7.1.3 Spannungsversorgung

Die Versorgungsspannung ist frei von hochfrequenten Störungen, hingegen wurden unter verschiedenen Belastungen folgende Werte der Versorgungsspannung gemessen:

Konfiguration	Spannung [V]	Stromverbrauch [mA]
Sensoren & Filter & LED	5.00	27.8
" & Fan	4.87	109
" & GPS & MAX232 (ohne Fan)	4.75	193
" & GPS & MAX232 & Fan	4.64	270

Tabelle 3: Spannungs- und Strommessungen am Sensor Board

Die DAQ-Card 1200 liefert laut Spezifikation 500 mA. Die Versorgungsspannung sinkt aber schon bei kleineren Belastungen. Ein kürzeres Kabel zwischen der I/O - Karte und dem Sensor Board könnte das Problem vielleicht etwas entschärfen. Als Folge haben wir für unsere Messungen den Fan und das GPS abgeschaltet, um alle unnötigen Stromkonsumenten zu eliminieren.

7.1.4 Lagemessungen

Die Lage unseres Geräts wurde statisch und dynamisch vermessen, d.h. im Stillstand und bei Bewegungen im Raum. Die dynamischen Messungen sind nicht sehr aussagekräftig, da die Bewegungen aufgrund fehlender Messeinrichtungen von Hand ausgeführt werden mussten. Sie sind deshalb nicht reproduzierbar, dennoch lassen sie auf verschiedene Fehlerquellen schliessen.

7.1.5 Statische Lagemessung

Hierzu wurde das Programm 'Cockpit' verwendet. Das Sensor Board stand dabei auf dem Fussboden, um Erschütterungen zu vermeiden. Unmittelbar nach dem Einschalten sind die Gyro-Daten mit erhöhten Fehlern behaftet, und so begannen wir die Messungen immer erst nach einer Einschwingphase von mindestens 20 Minuten. Es wurden nun wiederholte Messungen bei 150, 300 und 500 Hz gemacht. Die Resultate sind in der Tabelle 4 zusammengefasst.

Sensor	150 Hz	300 Hz	500 Hz
Horizontallage, 30"	2.5 deg	2.0 deg	2.0 deg
Richtung, 30"	0.7 deg	0.5 deg	0.5 deg
Horizontallage, 60"	5.0 deg	5.0 deg	5.0 deg
Richtung, 60"	1.5 deg	1.1 deg	1.1 deg

Tabelle 4: Drift der unregelten Lagemessung

Die angegebenen Werte dürfen nicht zu genau interpretiert werden, da die Streuung der wiederholten Messungen recht gross ist. Man erkennt aber, dass durch höhere Sampling-Raten keine signifikante Verbesserung erreicht werden kann. (Das System befindet sich ja auch im Stillstand.) Die ausgewiesene Richtung ist genauer als die Horizontallage. Dies liegt daran, dass die Richtung im Stillstand durch den präziseren Gyro ENV05 gemessen wird. Bei der dynamischen Messung vermischen sich dann allerdings die unterschiedlichen Genauigkeiten durch die Transformationen.

7.1.6 Dynamische Lagemessung

Das Sensor Board wurde in langsam kreisenden Bewegungen nach allen Seiten bewegt und wiederum mit den drei verschiedenen Sampling-Frequenzen vermessen. Wichtig ist dabei, dass die Drehgeschwindigkeitslimiten der Gyros nicht überschritten werden. Das 'schwächste Glied' hierbei stellt der ENV05 mit maximal 80 Grad pro Sekunde dar.

Bei Bewegungen mit weniger als ca. 30 Grad Ausschlag liegen die Abweichungen in derselben Grössenordnung wie bei Ruhelage des Systems, unabhängig von der Abtastfrequenz.

Macht man grössere Ausschläge, nehmen die Fehler abhängig von der Sampling-Frequenz zu. Bei 70 Grad sind sie mit 150 Hz etwa dreimal, mit 500 Hz noch etwa doppelt so gross.

7.1.7 Statische Positionsmessung

Hierzu wurde das Programm 'Espresso' verwendet. Das Sensor Board stand dabei wiederum auf dem Boden und wurde nicht bewegt. Die Positionsmessung hängt stark von einer korrekten Lagemessung ab, da man nur mit Hilfe der Lage zwischen Erdbeschleunigung und Fortbewegungsbeschleunigung unterscheiden kann. So führt ein Fehler von 1 Grad in der Horizontallage schon nach einer Minute zu einem Positionsfehler von 314 m, bzw. einer Geschwindigkeit von 10.5 m/s. Der im Kapitel 6 besprochene g-Vektor Aufrichtmechanismus sorgte deshalb während unserer Messungen für eine möglichst horizontale Plattform. Die gemessenen Positionsfehler können der Tabelle 5 entnommen werden.

Sensor	150 Hz	300 Hz	500 Hz
Abweichung, 30"	8 m	8 m	8 m
Abweichung, 60"	40 m	40 m	40 m

Tabelle 5: Drift der Positionsmessung (Lage geregelt)

Wie bei der Lagemessung ist auch hier die Streuung erheblich, die Werte sind deshalb nur als Grössenordnung zu verstehen. Eine dynamische Positionsmessung haben wir zwar gemacht, es sind aber keine sinnvollen Werte herausgekommen.

7.2 Erkenntnisse

7.2.1 Spannungsversorgung

Der Strombedarf des Systems inklusive GPS und Lüfter beträgt 270 mA und befindet sich somit innerhalb der Spezifikationen der PCMCIA-Messkarte. Hiermit können jedoch nur die Sensoren und Filter ausreichend mit Strom versorgt werden, obwohl ursprünglich vorgesehen

war, alle Komponenten über die Messkarte zu versorgen. Der Spannungsabfall von 0.35 V über dem Messkabel zur PCMCIA-Karte war jedoch zu hoch. Wir hatten deshalb bei Messungen in dieser Betriebsart das GPS und den Lüfter nicht angeschlossen, um die Messdaten nicht negativ zu beeinflussen.

7.2.2 Temperaturkompensation

Grundsätzlich ist es möglich eine gute Temperaturkompensation der Sensoren zu erreichen, da die Temperaturabhängigkeiten reproduzierbar sind. Allerdings besteht beim ENC03 die Schwierigkeit, dass der Einfluss der Temperatur auf das Signal noch einen differentiellen Anteil aufweist, d.h. falls sich die Temperatur schneller ändert, wird die Verzerrung dementsprechend grösser. (Es sei hier vermerkt, dass sich dieser Sensor wie ein 'drehgeschwindigkeitsabhängiger Temperatursensor' verhält.) Wir haben keine geeignete Funktion gefunden, welche sowohl schnelle als auch langsame Änderungen korrigieren kann.

Es blieb uns also nichts anderes übrig, als passende Umgebungsbedingungen vorzusetzen. Unter passenden Umgebungsbedingungen verstehen wir ein gemässigttes Klima ohne schnelle Temperaturänderungen im Verhältnis zur Messdauer. Zum gegenwärtigen Zeitpunkt handelt es sich um ein empfindliches Messgerät, bei dem solche Auflagen respektiert werden müssen. Jedoch gilt es zu beachten, dass die Zeitspanne, während der das System ohne Update präzise Resultate liefern sollte, eine sehr wichtige Rolle spielt. Mit der aktuellen Trägheitsnavigationsplattform hat es keinen Sinn, eine aufwändige Temperaturkompensation durchzuführen, da die Position nach weniger als einer Minute Betrieb ohne Update unbrauchbar ist.

7.2.3 Aufrichtmechanismus

Aufgrund der verhältnismässig grossen Drift der Gyros ENC03 ist eine länger andauernde Messung nur sinnvoll, wenn diese Drift kompensiert werden kann. Mit einem Aufrichtmechanismus, wie er in Kapitel 6 beschrieben wird, kann dieser Unzulänglichkeit erfolgreich begegnet werden.

7.2.4 Initialisation

Die Initialisation stellt ein Problem für sich dar. Beim Einschalten der Navigationsplattform müssen verschiedene Werte bestimmt werden, bei deren Berechnung man aber auf Schwierigkeiten stösst. Diese Probleme wollen wir nachfolgend genauer erläutern.

Die Position und die Lage im Raum werden folgendermassen berechnet:

$$\mathbf{s}(\mathbf{t}) = \iint a(t)dt^2 + \int v_0dt + s_0$$

und

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

Wobei $\mathbf{s}(t)$ für den Distanzvektor vom Ursprung und $\varphi(t)$ für den Lagevektor (α, β, γ) stehen. Bei der Initialisation müssen wir nun v_0 , s_0 und φ_0 bestimmen, also die Anfangsgeschwindigkeit, die Anfangsposition und die ursprüngliche Lage im Raum. Zusätzlich möchten wir noch den Nulloffset (z.B. durch Alterung, unvollkommene Temperaturkompensation etc.)

der sechs Sensoren kennen, da eine Abweichung dort direkt zu einem Positions- resp. Lagefehler aufintegriert wird. Dies sind also $5 * 3 = 15$ Initialwerte, die zu bestimmen sind.

Mit einem GPS kann man die Position und die Geschwindigkeit bei der Initialisation messen und so diese Werte bestimmen. Doch über die Lage im Raum macht das GPS keine Aussage und den Nulloffset der Sensoren kann auch nicht gemessen werden. Dafür benötigen wir mehr Informationen, sprich zusätzliche Sensoren.

Man kann sich jedoch einige Sensoren sparen, wenn man die Initialisierungsbedingungen entsprechend vorgibt. Es wäre denkbar, dass Stillstand und ebene Lage für die Initialisation des Geräts vorausgesetzt werden und somit v_0 und φ_0 implizit null sind. Auch die Nullwerte der Beschleunigungs- und Drehgeschwindigkeitssensoren sind nun direkt messbar, bis auf die Beschleunigung in z-Richtung, die beträgt nämlich $+1$ g. Alle Parameter sind auf diese Weise bestimmt. Allerdings sind diese speziellen Voraussetzungen nicht immer praktikierbar. Ich kann mein Auto nicht immer genau eben hinstellen, um mein Navigationssystem zu initialisieren. Um den allgemeinen Fall einer schiefen Initillage (φ_0) mit zu berücksichtigen, muss sie ebenfalls gemessen werden können. Hierzu darf ich auf keinen Fall meine Beschleunigungssensoren benutzen, weil ich diese sonst nicht mehr korrekt initialisieren kann: Man kann die eine Ungenauigkeit einer Komponente nicht mit Hilfe von Daten korrigieren, die selbst von dieser Komponente stammen.

Die Lage und der Nulloffset der sechs Sensoren zusammen ergeben neun Werte, wir brauchen also auch neun unabhängige Messgrößen, um diese zu bestimmen. Es sind mehr Informationen gefragt, dazu sind z.B. zwei Tiltsensoren und ein Kompass notwendig.

7.2.5 Richtungsrepräsentation

Drei oft verwendete Arten einer Repräsentation der Richtung im Raum werden im Kapitel 6 beschrieben. Die von uns gewählte Darstellung mit den Eulerwinkeln hat sicher in Bezug auf Rechenleistung und Anschaulichkeit grosse Vorteile. Wenn aber alle möglichen Orientierungen bis hin zur senkrechten Lage eingenommen werden sollen, treten Probleme auf, da sich dort Singularitäten befinden und sich die Rechenfehler in deren Nähe erheblich vergrössern. Für ein Fahrzeug oder Sportflugzeug stellt dies kein Problem dar. Hingegen, wenn die Plattform auf einem Roboterarm oder in einem Akrobatikflugzeug eingesetzt werden soll, wo senkrechte Lagen eingenommen werden können, ist eine Lagedarstellung mit 'Quaternions' oder 'Direction Cosines' ohne Singularitäten vorzuziehen. In einem weiteren Schritt könnten dann daraus immer noch die Eulerwinkel berechnet werden, um eine für den Menschen anschauliche Darstellung zu erhalten.

7.2.6 Rechenleistung

Im Programm 'Espresso' werden pro kompletten Sample Record etwa 80 Multiplikationen, 12 Sinus() oder Cosinus() und 2 ArcTan() Funktionen gebraucht, dazu ein vielfaches an Additionen. Bei einer Abtastfrequenz von 300 Hz ergibt dies ungefähr 20'000 64-bit Multiplikationen und 4200 trigonometrische Funktionen pro Sekunde. In diesen Zahlen beinhaltet sind die vorgestellten Temperaturkompensationen, die Transformationen und der Lage-Aufrichtmechanismus. Je nach Aufwand bei der Aufbereitung der Sensordaten und den verwendeten Rückführmechanismen braucht man natürlich erheblich mehr Rechenleistung.

7.3 Quellen der Ungenauigkeiten

Folgende Fehlerquellen sind unseres Erachtens relevant:

- Beschränkte Auflösung der Sensoren
- Bias- und Sensitivitätsdrift
- Nichtlinearitäten der Sensoren
- Alinierungsfehler (Die vs. Package, Package vs. Platine)
- Interpolations- und Quantisierungsfehler
- Rechenfehler wegen zu grossen Winkelveränderungen
- Gegenseitige Fehlerakkumulation

7.3.1 Beschränkte Auflösung

Eine der Hauptursachen für die Ungenauigkeiten des Gesamtsystems liegt in der beschränkten Auflösung der Sensoren. Der ENV05 kann Drehgeschwindigkeiten bis minimal 0.1 Grad pro Sekunde messen. Wenn man die halbe Auflösung als Fehler annimmt, also 0.05 Grad pro Sekunde, so ergibt dies allein eine Abweichung von 3 Grad pro Minute. Das Datenblatt des ENC03 liefert keine entsprechenden Angaben, der Sensor ist aber mit Sicherheit ungenauer. Den Datenblättern des ADXL210 kann folgende Formel zur Berechnung des 'Noise' entnommen werden. Wir haben die Cut-Off Frequenz unserer Filter bei 50 Hz angesetzt:

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

7.3.2 Bias- und Sensitivitätsdrift / Nichtlinearitäten

Im Stillstand resultiert der Fehler v.a. aus der Bias-Drift. Nichtlinearitäten und Sensitivitätsschwankungen spielen hier eine untergeordnete Rolle, da eventuelle Fehler durch die Initialisation (NullOffset) kompensiert werden. Die Drift wird zum allergrössten Teil von der Temperatur hervorgerufen. Zur Illustration kann man einen Finger während der Messung auf einen ENC03 setzen und die wegdriftende Lage sogleich beobachten.

Bei einer Bewegung kommen nun noch Nichtlinearitäten und Sensitivitätsveränderungen der Sensoren ins Spiel. Der ENC03 hat gemäss Datenblatt eine Abweichung der Linearität von 5 % Full Scale (FS), der ENV05 von immerhin 0.5 % FS. Die Sensitivität schwankt bei beiden um -20 / +10 % Full Scale und da die Umrechnungen der gemessenen Winkelgeschwindigkeiten in die Eulerwinkelgeschwindigkeiten mit den durch den Linearitätsfehler behafteten Eulerwinkeln selbst geschieht, entstehen neue Fehler.

Die Genauigkeit der Beschleunigungssensoren ist in dieser Hinsicht etwas besser, da sich deren Linearitätsfehler nur auf 0.2 % Full Scale belaufen. Durch die doppelte Integration wird dieser Vorteil jedoch mehr als ausgeglichen.

7.3.3 Alinierungsfehler

Die zwei Sensoren auf dem ADXL210 haben laut Datenblatt einen maximalen Alinierungsfehler zwischen Die und Package von ± 1 Grad. Zusätzliche Alinierungsfehler entstehen durch die Ungenauigkeit bei der mechanischen Anordnung der Sensoren auf dem Sensor Board. Die Koordinatentransformationen gehen aber davon aus, dass die Sensoren senkrecht aufeinander stehen. Es ist offensichtlich, dass hier ein Fehler entsteht.

Die drei oben genannten Fehlerquellen, nämlich die Bias-Drift, die Nichtlinearitäten und die Alinierungsfehler, können durch gute Sensormodelle kompensiert werden. Hierzu benötigt man allerdings recht aufwändige Messeinrichtungen. Zur Bestimmung der Nichtlinearität der Drehgeschwindigkeitssensoren ist ein Drehteller mit genauer Referenzumdrehungsgeschwindigkeit unerlässlich. Für die Vermessung der Beschleunigungssensoren kann man die Erdbeschleunigung verwenden, natürlich nur im Bereich von -1 bis $+1$ g. Darüber hinaus hilft nur ein Beschleunigungsschlitten. Schlussendlich sollte die Plattform als Gesamtsystem vermessen werden. Hier kann man nun die Alinierungsfehler sämtlicher Sensoren und des mechanischen Aufbaus bestimmen.

Wünschenswert wären also parametrisierbare Sensormodelle, deren Parameter bei der 'Factory Calibration' bestimmt und auf der Plattform gespeichert würden. Eine INS Plattform ist ein empfindliches Messgerät, weshalb eine solch aufwändige Vermessung jedes einzelnen Gerätes vor Inbetriebnahme gerechtfertigt ist.

7.3.4 Interpolations- und Quantisierungsfehler

Durch die Abtastung des analogen Signals entstehen Interpolations- und Quantisierungsfehler. In unserer Implementation haben wir bei der Integration lediglich die Stützwerte aufsummiert. Um die Größenordnung der Interpolationsfehler abzuschätzen, wurde die Plattform mit verschiedenen Abtastfrequenzen vermessen. Da die Signale gefiltert werden (Grenzfrequenz 25 Hz resp. 50 Hz), wird der Interpolationsfehler mit zunehmender Abtastfrequenz kleiner. Wie in Kapitel 7 beschrieben, ist die Genauigkeit unserer Plattform aber nur minim davon abhängig, d.h. es existieren Fehler, die um Größenordnungen grösser sind, als die Interpolationsfehler.

Zu den Quantisierungsfehlern ist zu bemerken, dass der Gyro ENV05 eine Auflösung von ungefähr 0.1 Grad pro Sekunde bei maximal ± 80 Grad pro Sekunde besitzt, somit würden 10.5 bit reichen, bzw. unser A/D ist mit 12 bit sicher genau genug. Die Auflösung des Sensors selbst ist somit schlechter als die A/D-Wandlung.

Der ADXL210 hat eine Auflösung von etwa 4.5 mg auf einen Bereich von ± 10 g, d.h. die Granularität unserer Abtastung ist in etwa gleich gross, wie die des Sensorssignals selbst. Hier spielt die Quantisierung also eine Rolle, und man könnte durch Mittelwertbildung und Dithering des Analogsignals vielleicht eine Verbesserung erreichen. Unsere Versuche mit eingeschaltetem Dithering der DAQ-Card 1200 und einer Mittelwertbildung über acht Messpunkte brachten aber keine erkennbare Verbesserung. Eine andere Variante wäre die Einschränkung des Bereiches auf z.B. ± 5 g, dann würde die Quantisierung noch halb so gross wie die Auflösung des Sensors ausfallen.

Interpolations- und Quantisierungsfehler existieren, sind aber in unserem Aufbau von untergeordneter Bedeutung wie die Messungen zeigten.

7.3.5 Rechenfehler aufgrund zu grosser Winkeländerungen

Im Kapitel 6 wird erklärt, dass zu grosse Winkeländerungen Rechenfehler zur Folge haben. Es spielt eine Rolle, ob zuerst eine Rotation um die x-Achse und dann eine um die y-Achse erfolgt, oder umgekehrt. Bei sehr kleinen Winkeln allerdings ist dieser Fehler vernachlässigbar. Man muss also darauf achten, dass die Abtastintervalle so klein gehalten werden, dass von einer 'gleichzeitigen' Drehung um die Achsen ausgegangen werden kann. Diese Fehler zeigen sich natürlich nur in bewegten Systemen bei grossen Winkelgeschwindigkeiten. Sie sind bei den statischen Messungen nicht relevant. In unserem Fall auch bei den dynamischen nicht, da bei der maximalen Winkelgeschwindigkeit von 80 Grad pro Sekunde (ENV05) und einer Abtastfrequenz von 300 Hz die Winkeländerungen kleiner als 0.3 Grad sind.

7.3.6 Fehlerakkumulation

Die Fehlerakkumulation ist ein Problem bei der Trägheitsnavigation, aber keine eigentliche Fehlerquelle. Jeder Parameter kann Auswirkungen auf alle anderen haben. Ein Fehler in der Lage um die eine Achse verfälscht die Transformationen und somit die anderen Lagewinkel. Dies wiederum führt zu Transformationsfehlern der Beschleunigungen und bewirkt zusätzlich, dass eine Komponente des g-Vektors irrtümlicherweise als Bewegungsbeschleunigung zu einer Geschwindigkeit resp. Distanz aufintegriert wird. Hierbei handelt es sich also um eine gegenseitige Abhängigkeit und Akkumulation der einzelnen Fehler.

8 Fazit

Eine Trägheitsnavigationsplattform mit Low-Cost Komponenten ist grundsätzlich möglich. Die von uns erreichte Genauigkeit der *Lagemessung* ist schon ziemlich gut. Um über längere Zeit eine brauchbare Lagemessung zu erhalten, sollte sie in Verbindung mit einem Aufrichtungsmechanismus eingesetzt werden.

Die *Positionsmessung* hingegen ist zum gegenwärtigen Zeitpunkt nicht möglich. Durch die beschränkte Genauigkeit der Sensoren verbunden mit der zweifachen Integration wird die Positionsangabe innert kürzester Zeit unbrauchbar. Eine Verbesserung der Auflösung um mindestens einen Faktor 10 lassen eine Positionsbestimmung realistisch erscheinen.

Durch die Korrektur von systematischen Fehlern, wie Temperaturdrift, Nichtlinearitäten, Sensitivitätsschwankungen, etc. kann zusätzlich viel zur Präzision der Plattform beigetragen werden.

Für eine Initialisierung der Plattform in jeder beliebigen Lage sind 3 weitere Sensoren (2 Tiltsensoren & 1 Kompass) nötig, da ansonsten die ursprüngliche Lage nicht vollständig bestimmt werden kann. Das Fehlen dieser Komponenten wird zur Zeit durch die Vorgabe der exakt horizontalen Ausrichtung bei der Initialisierung umgangen.

9 Ausblick

Es müssen verschiedene Probleme angegangen werden, um eine brauchbare Trägheitsnavigationsplattform zu erhalten:

Da als wesentliche Fehlerquelle die Auflösung der Sensoren ermittelt wurde, sind genauere Sensoren eine wichtige Voraussetzung für bessere Resultate. Der heutige Trend hin zu besseren und billigeren Sensoren wird dies ermöglichen.

Aber auch digitale Methoden zur Signalaufbereitung wie Mittelwertbildung oder das Kalmanfilter tragen zu Verbesserungen bei. Gerade das Kalmanfilter kann stark korrelierte Messgrößen, wie wir sie in der Problematik der Trägheitsnavigation vorfinden, gut korrigieren.

Bei grösserer Genauigkeit des Systems spielen dann Alinierungsfehler zwischen Die und Sensorpackage sowie zwischen der Platine und den Sensoren zunehmend eine Rolle. Diese gilt es zu messen und dann zu korrigieren.

Es müssen weitere sinnvolle Rückführungsmechanismen gefunden werden, ohne die eine Trägheitsnavigation auf Dauer nicht realistisch ist. Unser Lageregler nach der Schwerkraft hat sich hier zwar bewährt, es sind aber dennoch weitere Anstrengungen notwendig, insbesondere für eine Positionsrückführung.

Ein weiterer Schritt ist die Miniaturisierung. Das System muss kleiner und damit tragbar werden, um nicht nur in Autos, sondern auch in Bereichen wie 'Wearable Computing' oder in der 'Consumer Electronic' Verwendung zu finden. Die Vision von einem 'INS on a Chip' ist realistisch und wird in nicht allzuferner Zukunft sicherlich verwirklicht werden.

Literatur

- [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*, ETH Zurich, IFE, 1999.
- [4] A. Stiller, *Digitale Filterhardware*, ETH Zurich, IFE, 1996.
- [5] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, Peter Peregrinus Ltd., 1997.
- [6] E. Nebot S. Sukkariéh, H. Durrant-Whyte, "Inertial navigation aided with gps information," in *M2VIP '97*. 1997, IEEE Press.

A Softwarestruktur 'Espresso'

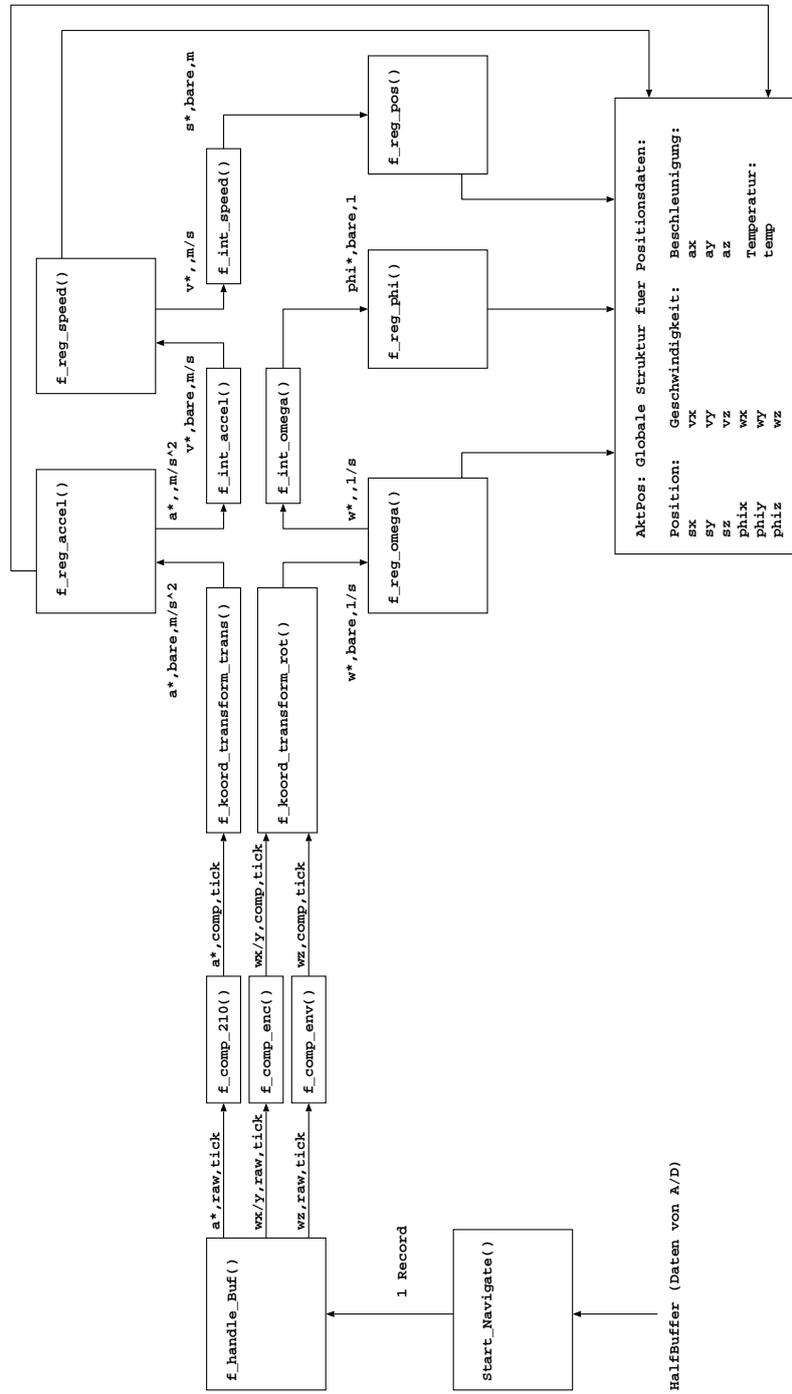


Abbildung 11: Softwarestruktur des Programms 'Espresso'

B Schema der Hardware

C Datenblätter