

Schlussbericht

für das Verbundvorhaben

S3ARV – Demonstration robuster autonomer GNC-Funktionen für unbemannte Luft- und Raumfahrzeuge mit reduzierter Baugröße (Small & Safe Space Autonomous Robotic Vehicles)

Teilvorhaben

Navigation und RAMS-Analysen

Ausführende Stelle

Technische Universität Dresden
Fakultät Elektrotechnik und Informationstechnik
Institut für Automatisierungstechnik (IfA),
Prof. Dr. techn. Klaus Janschek (Projektleiter)

Version 1.0

Dresden, 30. Juni 2015

Angaben zu Vorhaben

Verbundvorhaben

S3ARV – Demonstration robuster autonomer GNC-Funktionen für unbemannte Luft- und Raumfahrzeuge mit reduzierter Baugröße (Small & Safe Space Autonomous Robotic Vehicles)

Teilvorhaben

Navigation und RAMS-Analysen

Zeitraum

01.01.2012 – 31.12.2014

Gefördert durch

Bundesministerium für Wirtschaft und Energie

über Deutsches Zentrum für Luft- und Raumfahrt e.V., Projektdirektion Raumfahrtmanagement / Förderadministration

Zuwendung auf Ausgabenbasis

Förderkennzeichen 50RA1208

Kassenkennzeichen 810302553857

Verbundpartner

Universität Stuttgart

Fakultät Luft- und Raumfahrttechnik und Geodäsie

Institut für Flugmechanik und Flugregelung (iFR), Prof. Dr.-Ing. Walter Fichter

Technische Universität Dresden

Fakultät Elektrotechnik und Informationstechnik

Institut für Automatisierungstechnik (IfA), Prof. Dr. techn. Klaus Janschek

Inhaltsverzeichnis

Inhaltsverzeichnis.....	3
Akronyme	5
1 Aufgabenstellung.....	6
2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde	7
2.1 Institut für Flugmechanik & Flugregelung (iFR), Universität Stuttgart.....	8
2.1.1 Verfügbare Technologien	8
2.1.2 Relevante Methoden	8
2.2 Institut für Automatisierungstechnik (IfA), Technische Universität Dresden	9
2.2.1 Verfügbare Technologien	9
2.2.2 Relevante Methoden	9
3 Planung und Ablauf des Vorhabens.....	10
4 Wissenschaftliche und technische Basis des Vorhabens	14
4.1 Wissenschaftliche und technische Basis für das Vorhaben bei Verbundpartnern	14
4.1.1 Institut für Flugmechanik & Flugregelung (iFR).....	14
4.1.2 Institut für Automatisierungstechnik (IfA)	14
4.2 Andere bekannte Konstruktionen, Verfahren und Schutzrechte, die für die Durchführung des Vorhabens benutzt wurden.....	14
5 Zusammenarbeit mit anderen Stellen	15
5.1 Zusammenarbeit der Verbundpartner.....	15
5.2 Zusammenarbeit mit Dritten.....	15
6 Verwendung der Zuwendung und erzielttes Ergebnis	16
6.1 Erarbeitung eines detaillierten Systemkonzeptes.....	16
6.2 Systemtestbench mit Flugplattform	16
6.3 Systemtestbench mit Lander-Demonstrator	20
6.4 Perzeption	21
6.5 Navigation	23
6.5.1 Ansatz zur autonomen bildbasierten Navigation	23
6.5.2 Basis-SLAM Algorithmus.....	24
6.5.3 Laborbasierte Testbenches für die Navigationsalgorithmen	26
6.5.4 Test und Verifikation	27
6.5.5 Echtzeitbildverarbeitung	28
6.6 RAMS-Analysen	29
6.6.1 Optimale Kapselung von ausfallkritischen Softwareeinheiten.....	29
6.6.2 Erkennung und Behandlung von kritischen Fehlern.....	32
6.7 Wissenschaftliche IfA-Highlights im Überblick.....	33
6.8 Bearbeitungsstand nach Arbeitspaketen: Zusammenfassung.....	34
7 Wichtigste Positionen des zahlenmäßigen Nachweises	34
7.1 0812 „Beschäftigte E12-E15 / BAT IIa bis I“ und 0822 „Beschäftigungsentgelte“	34
7.2 0831 Gegenstände bis zu 410 / 400 EUR	34
7.3 0843 Sonstige allgemeine Verwaltungsaufgaben	34

7.4	0846 Dienstreisen.....	34
7.5	0850 Gegenstände und andere Investitionen von mehr als 410 / 400 EUR im Einzelfall.....	35
8	Notwendigkeit und Angemessenheit der geleisteten Arbeit.....	35
9	Voraussichtliches Nutzen und Verwertbarkeit des Ergebnisses im Sinne des fortgeschriebenen Verwertungsplans	36
10	Während der Durchführung des Vorhabens dem Zuwendungsempfänger bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen	37
11	Erfolgte oder geplante Veröffentlichungen des Ergebnisses	37
12	Quellen	38
13	Anhang	38
13.1	Modifikation von wichtigen Systembegriffen aus dem Antrag	38
13.2	Zuordnung spezieller Bezeichnungen in UML-Modellen.....	40

Akronyme

CPU	central processing unit (engl.)
EKF	Extended Kalman Filter (engl.)
FPGA	Field Programmable Gate Array (engl.)
GNC	Guidance, Navigation, Control (engl.)
GPU	graphics processing unit (engl.)
HiL-TB	Hardware-in-the-Loop-Testbench (engl.)
IMU	inertial measurement unit (engl.)
LiDAR	Light detection and ranging (engl.)
PiL-TB	Processor-in-the-Loop-Testbench (engl.)
RAMS	Reliability, Availability, Maintainability, Safety (engl.)
RANSAC	random sample consensus (engl.)
SFD	software-implemented hardware fault detectors (engl.)
SiL-TB	Software-in-the-Loop-Testbench (engl.)
SIMD	single instruction, multiple data (engl.)
SLAM	Simultaneous, Localisation and Mapping (engl.)
UAV	unmanned aerial vehicle (engl.)
UML	Unified Modeling Language (engl.)

1 Aufgabenstellung

In diesem Kapitel wird die Aufgabenstellung für das gesamte Verbundvorhaben (weiter Vorhaben) dargelegt.

Zukünftige Missionen im Bereich der Raumfahrtrobotik werden sich mittel- und langfristig stark auf Autonomiefunktionen stützen müssen, um die gewünschte operationelle Flexibilität gewährleisten zu können. Zu den geforderten Autonomiefunktionen gehören insbesondere die Navigation (inkl. die Erfassung der Umgebung), die Bewegungsplanung (Guidance) unter Berücksichtigung von operationellen Grenzen und Hardware-Grenzen, sowie fehlertolerantes Verhalten. Bei realistischen Anwendungen muss als eine weitere wesentliche Randbedingung eine unbestimmte Informationslage berücksichtigt werden. Unter die Randbedingung der Machbarkeit ist es zur Erfüllung der o. g. Autonomieforderungen sinnvoll, „kleine“ Systeme im Sinne von Leistungsaufnahme, Masse und Baugröße anzustreben. Das bedeutet jedoch auch, dass an Bord nur limitierte Sensorinformation und Rechnerleistung zur Verfügung steht.

Aufgrund dieser Überlegungen kann ein Entwicklungsbedarf im Bereich von folgenden Autonomiefunktionen

- Perzeption (Akquisition der Umgebung mit Sensoren),
- Navigation (Lokalisierung und Mapping),
- Control (Fahrzeugstabilisierung, Schnittstelle zu Guidance),
- Guidance (Bahnplanung, Hindernisvermeidung und Einhaltung dynamischer Beschränkungen),
- RAMS (Reliability, Availability, Maintainability, Safety)

identifiziert werden, der folgende Zielsetzungen verfolgen sollte:

- Small: Trotz der hohen Anforderungen an die Rechenleistung, die üblicherweise bei autonomen Systemen erforderlich ist, soll die Baugröße, Masse und Leistungsaufnahme des Systems begrenzt bleiben und damit Kosten limitiert beziehungsweise reduziert werden. Dies resultiert in begrenzter Rechenleistung, weshalb neuartige Ansätze im Zusammenspiel zwischen Hardware und Software Anwendung finden müssen.
- Safe. Hinter dem Begriff „safe“ verbergen sich mehrere Teilaspekte: Erstens sollen nur Algorithmen zum Einsatz kommen, die in endlicher Rechenzeit ein Ergebnis liefern, auch wenn dieses ggf. suboptimal ist. Zweitens sollen sichere Lösungen der Navigation und Guidance berechnet werden. Bei der Bewegungsplanung wird insbesondere gefordert, dass eindeutig definierte Randbedingungen/Begrenzungen garantiert, d. h. sicher eingehalten werden (Kollisionsfreiheit, Flugzustände, etc.). Die Aufgaben Guidance und Navigation können im Rahmen vom Simultaneous Planning, Localisation and Mapping (SPLAM) verkoppelt sein. Drittens benötigt ein sicheres System Mechanismen zur Fehlererkennung, Lokalisierung und ggf. Diagnose. Bei kleinen Systemen muss hierbei auf funktionale und analytische Redundanz anstelle reiner Hardware-Redundanz zurückgegriffen werden.
- Space. Das zu entwickelnde System soll das Potential besitzen, in raumfahrttaugliche Hardware umgesetzt zu werden.
- Autonom. Das zu entwickelnde System soll in einer unbekanntenen Umgebung unabhängig von einer externen Infrastruktur funktionieren.

Das Vorhaben verfolgt somit das Ziel, zukünftig erforderliche Autonomiefunktionen im Rahmen einer ganzheitlichen Systembetrachtung unter den o.g. Randbedingungen eines Raumfahrteinsatzes zu entwickeln und mithilfe eines Funktionsmodells zu demonstrieren.

Der Schwerpunkt der Referenzmission und damit des Flugvehikels liegt bei diesem Vorhaben bei der präzisen, sicheren und autonomen Landung auf Planeten. Es soll ein Flugvehikel mit 6

Freiheitsgraden betrachtet werden. Insbesondere in Hinsicht auf „Small“ und „Autonom“ sollten kompakte optische und inertielle Sensoren favorisiert werden.

Es werden zwei Szenarien verfolgt. Das Landeszenario mithilfe eines unbemannten Fluggerätes vom Typ Octocopter repräsentiert eine terrestrische Anwendung, z. B. im Bereich Katastrophenhilfe. Ein Großteil der Ergebnisse ist im Sinne der Funktionalität jedoch auch auf Raumfahrtsszenarien übertragbar. Das extraterrestrische Szenario wird mithilfe eines roboterbasierten Testbenches mit einem Demonstrator des Landehelikopters simuliert.

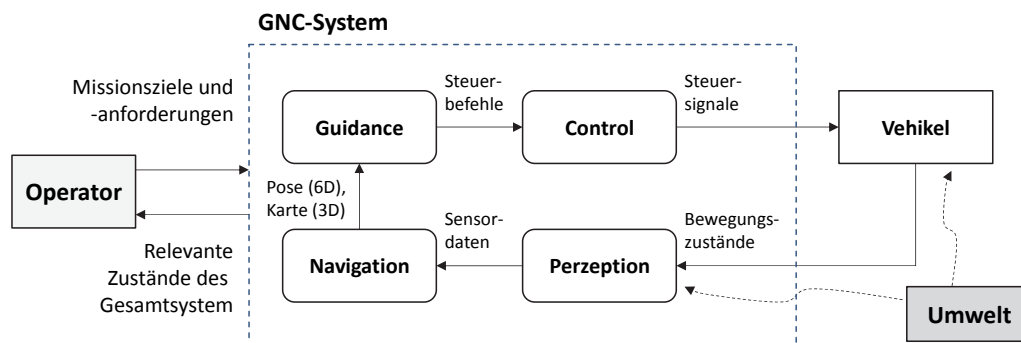


Abbildung 1.1 Erforderliche Autonomiefunktionen im Systemkontext.

Die vom Institut für Flugmechanik und Flugregelung (iFR) schwerpunktmäßig bearbeiten Gruppen von Aufgaben sind:

- Guidance-Funktion (Bahnplanung unter Einhaltung von dynamischen Beschränkungen inkl. Hindernisvermeidung),
- Control-Funktion (Basisregler zur Fahrzeugstabilisierung und Schnittstelle für übergeordnete Guidance),
- Untersuchungen zur Analytischen Redundanz, Detektion von Aktuatorausfällen und Rekonfiguration,
- Perzeption mit inertialer Sensorik.

Das Institut für Automatisierungstechnik (IfA) hat sich im Projekt auf folgende Aufgabenbereiche konzentriert:

- Navigation (inkl. Mapping, auf Grundlage der Daten aus der Perzeption),
- RAMS-Analysen,
- Perzeption mit bildgebender Sensorik.

Gemeinsame Aspekte des Verbundvorhabens werden vor allem in Kapiteln 1-5, 8, 10 und 13 des vorliegenden Berichtes dargelegt. Kapitel 1-5, 6.1, 6.2, das letzte Unterkapitel von Kapitel 6, Kapitel 8, 10 und 13 sind in den Schlussberichten der Verbundpartner identisch. Auf die Aspekte des Teilvorhabens des jeweiligen Verbundpartners wird in den Schlussberichten ab Kapitel 6 eingegangen.

2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Zu den wichtigen Voraussetzungen, unter denen das Vorhaben durchgeführt wurde, gehört vor allem die gute wissenschaftliche und technologische Basis für Forschung und Entwicklung auf projektbezogenen Themengebieten der Verbundpartner:

- Institut für Flugmechanik und Flugregelung (iFR), Universität Stuttgart: Guidance, Control von autonomen Fluggeräten,

- Institut für Automatisierungstechnik (IfA), Technische Universität Dresden: optische Navigation und RAMS-Analysen.

Ein Überblick wird am Ende des Kapitels gegeben.

Weitere Voraussetzungen stellten eine klare Definition und eine optimale Abgrenzung der Verantwortlichkeitsbereiche der Verbundpartner dar, die eine effektive Gestaltung der Zusammenarbeit im Projekt ermöglichten.

2.1 Institut für Flugmechanik & Flugregelung (iFR), Universität Stuttgart

Das Institut für Flugmechanik und Flugregelung (iFR) der Universität Stuttgart wurde 1991 gegründet und wird seit April 2007 von Prof. Walter Fichter geleitet, der davor 17 Jahre lang in der Raumfahrtindustrie tätig war.

Die Forschungsaktivitäten am iFR gliedern sich in die folgenden drei Bereiche: Autonome Systeme mit den Schwerpunkten: kleine autonome Fluggeräte (UAVs) und autonome Unterwasserfahrzeuge (AUVs); Präzisionsregelungssysteme für Satelliten mit den Anwendungsbereichen „Drag-free Control“ und „Precision Pointing“ und zeitperiodische Systeme für aktive Rotoren.

Das iFR ist in allen Bereichen in Industrieprojekte involviert, Beispiele aus der Raumfahrt sind LISA Pathfinder (Astrium Friedrichshafen), DEOS (Astrium Bremen), SRSIM (Astos Solutions). Ähnliche Verbindungen existieren auch im Bereich der Luftfahrt, hauptsächlich zu Eurocopter (jetzt Airbus Helicopters) und MBDA.

2.1.1 Verfügbare Technologien

Am iFR werden seit einigen Jahren Forschungsarbeiten im Bereich autonomer unbemannter Fluggeräte durchgeführt. Vor 2007 waren diese Arbeiten hauptsächlich in Aktivitäten zur Lenkung, Regelung und Navigation von Luftschiffen eingebettet. Seit 2007 sind diese Arbeiten auf eine breitere Basis gestellt worden. Hieraus ergaben sich einige Entwicklungen, die als Vorarbeiten für den vorliegenden Antrag relevant sind.

FPGA-basierter Bordrechner. FPGA-Bausteine erlauben eine extrem schnelle Ausführung von Programmen, bei kleiner Baugröße und geringem Leistungsbedarf. Die Integration von FPGA und herkömmlichen Mikroprozessor in einen Bordrechner (eine Platine) erlaubt es, spezielle rechenintensive und ggf. parallelisierbare Algorithmenteile auf den FPGA-Teil auszugliedern.

Bodenstation. Zur Bedienung eines unbemannten Fluggerätes ist natürlich eine Bodenstation notwendig, mit deren Hilfe man Telekommandos senden und entsprechende Telemetriedaten zur Überwachung und Steuerung der Flugmission empfangen kann.

Echtzeit-Testbench. Eine Testbench, die noch vor dem eigentlichen Flugversuch benutzt wird, um die in Bordsoftware implementierten Algorithmen in einer Echtzeitumgebung möglichst realitätsnah zu verifizieren und zu testen. Das iFR benutzt dazu eine eigens entwickelte Lösung (siehe auch Kapitel 4.1.1)

2.1.2 Relevante Methoden

Nichtlineare modellprädiktive Regelung zur Bewegungsplanung. Am iFR werden Bewegungsplanungsverfahren auf der Basis der nichtlinearen modellprädiktiven Regelung entwickelt. Diese Verfahren werden speziell zur Implementierung auf relativ kleinen (Masse, Größe, Leistung) Bordrechnern zugeschnitten, z. B. einem FPGA-basierten Bordrechner.

Analytische Redundanz. Sensorüberwachung mithilfe analytischer Redundanz und ggf. paralleler Filterung wurde am iFR sowohl im Rahmen der Beteiligung an DEOS Phase B entwickelt, als auch innerhalb der Aktivitäten des FPGA-basierten Bordsystems. Dies war ein sehr guter Ausgangspunkt für die geplanten Arbeiten.

Entwurf von komplexeren Basisregelkreisen. In Spezialfällen, z. B. bei Bodeneffektstörungen können besondere regelungstechnische Maßnahmen bei Basisregelkreisen notwendig sein. Das iFR besitzt hier einschlägige Erfahrung.

2.2 Institut für Automatisierungstechnik (IfA), Technische Universität Dresden

Das Institut für Automatisierungstechnik (IfA) an der Fakultät Elektrotechnik und Informationstechnik der Technischen Universität Dresden beinhaltet die beiden Professuren Automatisierungstechnik (Prof. K. Janschek) und Prozessleittechnik (Prof. L. Urbas) und wird von Prof. Janschek als Institutsdirektor geleitet. Vor seiner Berufung im Jahre 1995 war Prof. Janschek 13 Jahre in der Industrie tätig, davon 10 Jahre in der Raumfahrt (Guidance, Navigation and Control).

Die Forschungsaktivitäten der Professur Automatisierungstechnik (Prof. Janschek) gliedern sich in folgende Bereiche: Systementwurf, Robotik und Mechatronik, Industrielle Automatisierung.

2.2.1 Verfügbare Technologien

Optischer-Fluss-Prozessoren (optical flow processors). Am IfA werden Echtzeit-Implementierungen der Berechnung des optischen Flusses mit Labormodellen und Demonstratoren untersucht: PC-basierte Implementierung mit mehreren Prozessorkernen und SIMD-Einheiten, Implementierung in einem Grafikprozessor (GPU); Implementierung in einem optischen Prozessor. Durch ein spezielles optoelektronisches Design kann auf Basis eines optischen Korrelators ein extrem leistungsfähiger und gleichzeitig kompakter Optoelectronic Optical Flow Processor realisiert werden, wie er für autonome visuelle Navigationssysteme benötigt wird. Die entwickelten Modelle sind aufgrund von Masse/Volumen/Leistungsverbrauch nicht auf sehr kleinen Flugplattformen einsatzfähig, bieten jedoch hinreichende Echtzeitrechenleistungen für Feldtests bei einer Implementierung im bodenbasierten Systemsegment.

Spacecraft Rendezvous & Planetary Landing Testbench. Am IfA steht ein Labor-Testbench je nach Konfiguration für Spacecraft Rendezvous bzw. Planetare Landung mit drei translatorischen und drei rotatorischen Freiheitsgraden zur Verfügung.

2.2.2 Relevante Methoden

Optical Flow Navigation. Unter Nutzung der Technologie der optischen Korrelatoren arbeitet das IfA seit mehreren Jahren an Verfahren zur visuellen (bildgestützten) Navigation mit Monokamerasystemen unter Verwendung des optischen Flusses.

Planetary Landing (ESA). Seit 2003 arbeitet das IfA an Konzepten und Verfahren zur visuellen Navigation für planetare Landung auf der Basis der o. a. Optical Flow Navigation mit einer Monokamera. Als Haupt- und Subcontractor wurde in Kooperation mit EADS-Astrium Toulouse im Rahmen von mehreren ESA-Projekten eine viel beachtete und über ausführliche Simulationen verifizierte Lösung entwickelt.

Rendezvous-SLAM. Am IfA wurden im Rahmen eines TUD/EADS-Astrium Graduiertenkollegs Verfahren zum autonomen Servicing von unbekanntem und nicht-kooperativen Raumfahrzeugen untersucht. Für die Aufgabe der Relativnavigation von Chaser (Service-Satellit) zu Target (nicht-kooperatives Raumfahrzeug, z. B. havariertes Satellit) wurde ein neuer Filteransatz – Rendezvous-SLAM – entwickelt.

Modellbasierte Performance- und RAMS-Analyse. Modernes Systems Engineering stützt sich auf eine möglichst frühzeitige und umfassende Verhaltensbeschreibung komplexer Systeme mittels semi-formalen Systemmodellen. Am IfA werden spezielle Ansätze entwickelt, um ausgehend von UML-basierten High-Level-Systemmodellen bereits in frühen Projektphasen gesicherte Verhaltensaussagen hinsichtlich Performance (z.B. Laufzeiten) sowie RAMS - Reliability, Availability, Maintainability, Safety (=Dependability) zu machen.

3 Planung und Ablauf des Vorhabens

Einen Ausgangspunkt für das Vorhaben stellten die Vorarbeiten der Verbundpartner, die in Kapitel 2 und speziell in Kapitel 4.1 zusammenfassend dargestellt sind. In der Ausgangsform existierten diese Einzelfunktionen unabhängig voneinander. Diese sollten im Sinne eines Gesamtsystems integriert betrachtet werden. Beide Partner sollten also gleichzeitig und abgestimmt an einer Thematik arbeiten.

Das Vorhaben wurde bei der Planung in vier Gruppen von Arbeitspaketen unterteilt (Abbildung 3.1):

- In der Gruppe **AP1: „Systementwurf“** wurden Definition von Systemanforderungen und Randbedingungen im Kontext der Projektziele, Erarbeitung eines Systemkonzeptes auf Hardware- und Softwareebene, Erarbeitung eines Testplans sowie eine abschließende zusammenfassende Auswertung der gesammelten Daten geplant.
- Zu den Zielen der Gruppe **AP2: „Algorithmenentwurf & Entwicklung“** gehörten Entwurf und Entwicklung jeweiliger GNC-Algorithmen inkl. Implementierung, Verifikation und Test in geeigneten laborbasierten Simulations-Testbenches (SIM-TB), Software-in-the-Loop-Testbenches (SiL-TB) und Hardware-in-the-Loop-Testbenches (HiL-TB), wobei bei den letzteren bordbasierte Rechereinheiten zum Einsatz kommen sollten. In dieser Gruppe sollten auch die Aspekte der RAMS und der Analytischen Redundanz bearbeitet werden. Die Entwicklung der benötigten Testbenches wurde in den Arbeitspaketen vorgesehen. Einige Komponenten sollten dabei in den Arbeitspaketen der Gruppe AP3 bereitgestellt werden.
- Im Plan der Gruppe **AP3: „Implementierung & Integration“** waren Implementierung und Integration von wichtigen Systemkomponenten: Bordsensorik, bordbasierte und bodenbasierte Rechereinheiten, Bordrechner-Software als Rahmensoftware für die Implementierung der GNC-Algorithmen, Bodenstation (Ground Control & Monitoring) vorgesehen. Der Bordrechner mit einer konfigurierten Betriebssystemsoftware war dabei generell für (spätere) Implementierung der GNC-Algorithmen vorgesehen. Die Navigationselektronik sollte vor allem aus einer Kamera, Inertialsensorik, einem bordbasierten Bildverarbeitungsrechner und einem bodenbasierten und über ein RF-Link verbundenen Optischer-Fluss-Prozessor bestehen. Für die zwei letzteren Rechereinheiten sollte eine Bildverarbeitungssoftware zur Extraktion des optischen Flussfeldes entwickelt werden.
- Gruppe **AP4: „Demonstration“**. Im Mittelpunkt der Gruppe standen Vorbereitung und Durchführung von Test und Verifikation der entwickelten Algorithmen unter realistischen Bedingungen terrestrischer Flugmissionen und unter simulierten realitätsnahen Bedingungen einer planetaren Landung. Dafür sollten die Testplattformen (Flugplattform, Testbench) (AP4.2, AP4.1) unter Verwendung der aus AP3 bereitgestellten Bordrechner, Navigationselektronik und der Bordrechner-Software entwickelt werden. In nachfolgenden iterativen Schritten sollten Integrationen mit definierten Implementierungen der Algorithmen sowie einigen bodenbasierten Komponenten durchgeführt werden (in AP4.3). Die definierten Versionen der Algorithmen sollten dann im Systemkontext getestet und verifiziert werden (AP4.3). Es wurden auch laufende Flugdemonstrationen der erreichten Einsatzfähigkeit des entwickelten Gesamtsystems aus Flughardware und -software unter realen terrestrischen Umgebungsbedingungen geplant (AP4.4).

Auf Grundlage der vorgestellten Struktur der Arbeitspakete wurde die Zeitplanung des Projektes entworfen und bei Durchführung des Vorhabens verfolgt. Das Balkendiagramm in Abbildung 3.2 zeigt den Ablauf des Projektes, sowie die Arbeitsteilung zwischen den Verbundpartnern.

An dieser Stelle soll vermerkt werden, dass etliche Begriffe und Bezeichnungen einiger Systemkomponenten aus dem Projektantrag während der Entwicklung zum Zwecke der Optimierung der Systemarchitektur modifiziert wurden. Diese Sachverhalte werden in Kapitel 13.1 erläutert. Im nachfolgenden Text werden die aktuellen Namen verwendet. Auf die ursprünglichen Antragsnamen wird nach Bedarf laufend hingewiesen.

Die Arbeit wurde in folgende Projektphasen eingeteilt (siehe Abbildung 3.2).

Zunächst wurden in der Projektphase-1 die Systemanforderungen und der Systementwurf in einer Grundversion festgelegt (in AP1). Hierzu gehören auch relevante Randbedingungen und Konzepte zur Test und Verifikation. Die Arbeiten in dieser ersten Projektphase wurden in einem integrierten Team der beiden Partner durchgeführt.

Die Projektphase-2 diente vor allem dazu, frühzeitig eine Testplattform mit Grundfunktionen auf Hardware- und Software-Ebene zu schaffen und kritische technische Unzulänglichkeiten rechtzeitig zu erkennen und ggf. zu korrigieren (in AP2, AP3, AP4). Jeder der beiden Partner arbeitete in dieser Phase zu einem deutlichen Teil auf seinem Spezialgebiet. Die integrierte Grundfunktionalität (V0) konnte in einem Flugversuch beim Forschungsgespräch 2 planmäßig und erfolgreich demonstriert werden.

Im weiteren Verlauf des Projektes (Projektphase-3) wurde an der Erweiterung der Funktionalität gearbeitet. Die Schwerpunkte waren erforderliche Erweiterung der Funktionalitäten der Flugplattform und der bodenbasierten Rechereinheiten in Richtung der angestrebten Systemintegrität (in AP3) sowie Aufbau und Tests von weiteren geplanten Flugplattformen (AP4.2) und des Testbenches (AP4.1); Modellierung der optischen Perzeption (in AP2.1); Entwicklung von Algorithmen für die Autonomiefunktionen (Guidance, Navigation, Control in AP2) und Berücksichtigung der RAMS-Aspekte (AP2.5) und der analytischen Redundanz (AP2.6); Systemintegration, Test und Verifikation der entwickelten Funktionen und Algorithmen (AP4.3); Flugdemonstrationen (AP4.4). Die Arbeiten der Verbundpartner wurden in der Phase entsprechend ihren Spezialgebieten und unter Berücksichtigung der erforderlichen Kooperation (vor allem Systemintegration und -Systemtest) durchgeführt (Abbildung 3.2).

Die Arbeiten wurden im Wesentlichen nach der vorgestellten Planung durchgeführt. Nennenswert sind in diesem Kontext folgende Anpassungen: beim Arbeitspaket AP2.4 „SPLAM“ wurde die Zielstellung geändert und im 4. Fortschrittsgespräch abgestimmt; im Arbeitspaket AP3.4 „Bodenstation“ wurde ein Teil der ursprünglich für iFR geplanten „Bodenstation“ bei der Optimierung und Umsetzung der Systemarchitektur von IfA übernommen (zur Überwachung der Navigationsalgorithmen und RAMS-bezogenen Funktionen); im Arbeitspaket AP4.2 „Flugplattform“ wurden die zwei Flugplattformen auch bei IfA gebaut (konform mit den geplanten Ausgaben und der Anlage zum Antrag „Gegenstände und andere Investitionen (Geräteliste“).

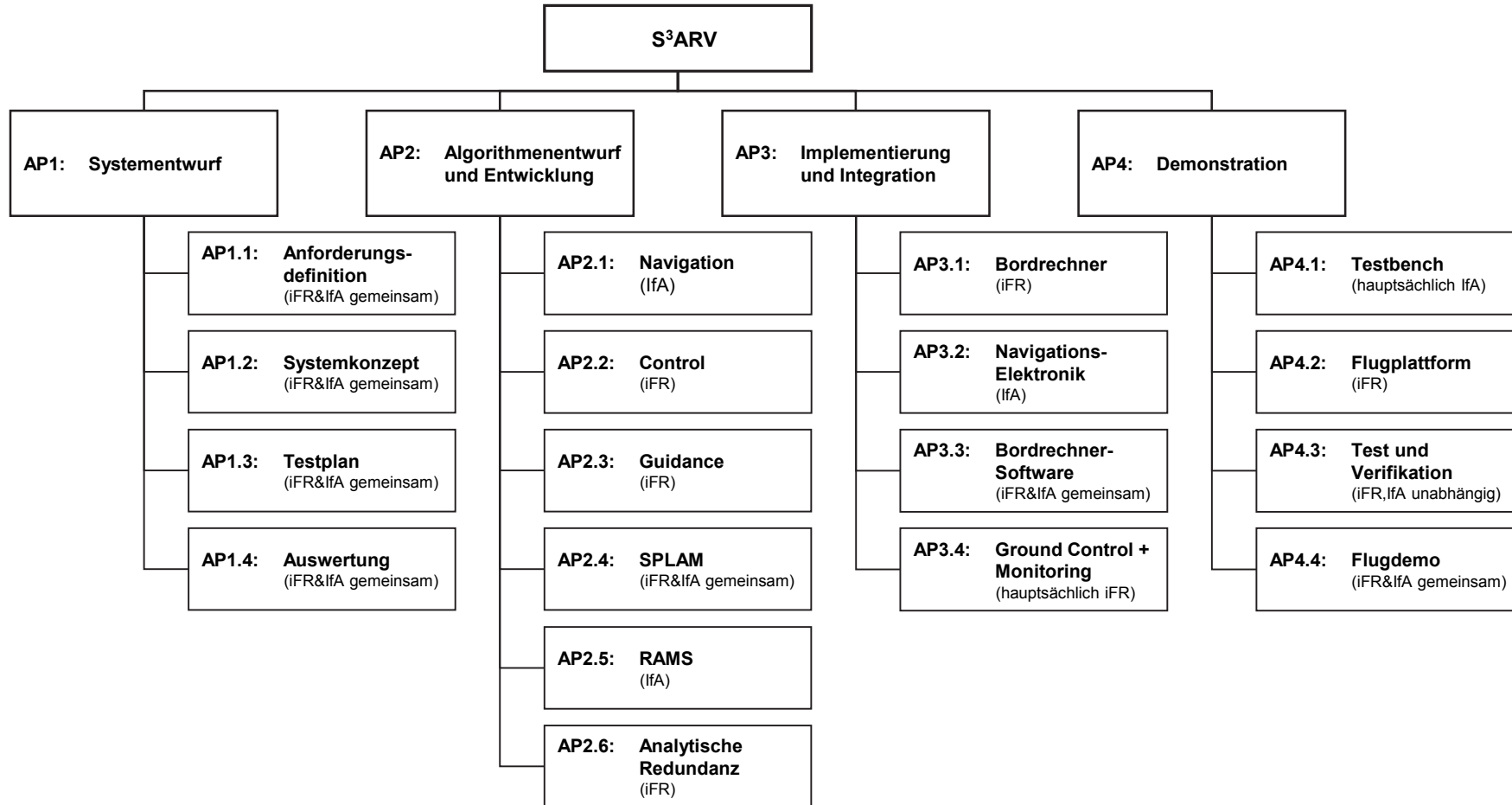


Abbildung 3.1 Arbeitsstrukturplan des Vorhabens, sowie die geplante Arbeitsteilung zwischen IfA und iFR.

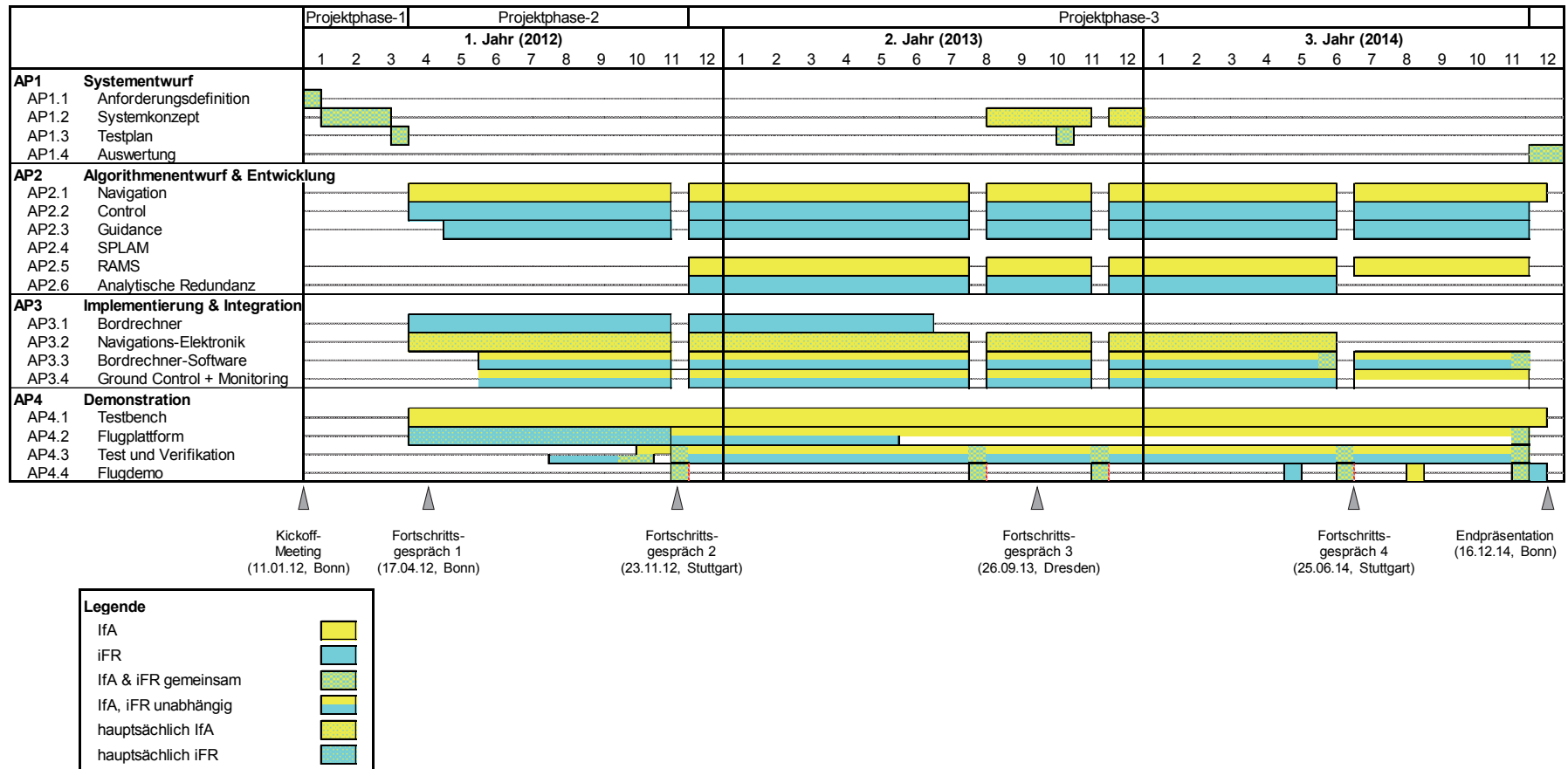


Abbildung 3.2 Balkendiagramm: Ablauf des Vorhabens, sowie die Arbeitsteilung zwischen IfA und iFR.

4 Wissenschaftliche und technische Basis des Vorhabens

In diesem Kapitel wird wissenschaftlicher und technischer Stand erläutert, an den die Arbeiten in diesem Vorhaben angeknüpft wurden. Schwerpunktmäßig werden bekannte Konstruktionen, Verfahren und Schutzrechte dargestellt, die für die Durchführung des Vorhabens benutzt wurden.

4.1 Wissenschaftliche und technische Basis für das Vorhaben bei Verbundpartnern

4.1.1 Institut für Flugmechanik & Flugregelung (iFR)

FPGA-basierter Bordrechner. Ein am iFR entwickelter Bordcomputer für kleine UAV (Generation eins siehe Weimer 2010), der in der zweiten Generation neben einem herkömmlichen Mikroprozessor auch einen FPGA-Baustein als Co-Prozessor für rechenaufwändige Algorithmen enthält.

Bodenstation. Eine Software, die am Institut für Flugmechanik und Flugregelung entwickelt wurde, um bei Flugversuchen Telemetriedaten von UAVs zu visualisieren und Telekommandos zur Missionssteuerung und Aufzeichnung von Logdateien zum UAV zu senden (siehe auch Abschnitt 2.1.1).

Echtzeit-Testbench. Eine am iFR entworfene Testbench bestehend aus Echtzeitsimulationsumgebung (MATLAB/Simulink), Interfaceplatinen zur Anbindung des FPGA-basierten Bordrechners an die Simulationsumgebung sowie Visualisierungsmöglichkeiten, um noch vor dem Flugversuch Funktionalitäten der implementierten Algorithmen zu verifizieren.

Nichtlineare modellprädiktive Regelungsalgorithmen zur Bewegungsplanung. Methode zur Bahn- bzw. Bewegungsplanung basierend auf Zeitbereichssimulationen, das in seiner speziellen Ausprägung zur Implementierung auf kleinen UAVs mit kleiner Hardware (Masse, Größe, Leistungsaufnahme) besonders gut geeignet ist. Das iFR konnte diese Methode für Flächenflugzeuge bereits erfolgreich einsetzen (Joos 2011a und Joos 2011b).

4.1.2 Institut für Automatisierungstechnik (IfA)

Verfahren und Software zur Berechnung des Optischen Flusses. Ein Verfahren zur korrelationsbasierten Berechnung des optischen Flusses (Janschek 2005) und eine Demo-Software mit Echtzeitimplementierung für Intel-Prozessoren unter Verwendung mehrerer Prozessorkerne und SIMD-Einheiten (Schlicht 2011).

Spacecraft Rendezvous & Planetary Landing Testbench. Ein roboterbasiertes Labor-Testbench mit drei translatorischen und drei rotatorischen Freiheitsgraden. Das Testbench wurde am IfA ursprünglich für Simulation von Spacecraft-Rendezvous-Missionen entwickelt (MiPOS - Miniature Proximity Operation Simulator, Schnitzer 2011).

Modellbasierte Performance- und RAMS-Analyse. Ein Forschungsgebiet am Institut für Automatisierungstechnik stellt die modellbasierte Systemanalyse dar, die primär auf Verbesserung der RAMS-Eigenschaften komplexer heterogener Systeme orientiert ist. Ein im Rahmen von Vorarbeiten entwickeltes probabilistisches Modell für Fehlerfortpflanzung (Morozov 2014) wurde im vorliegenden Vorhaben als Grundlage für die Arbeit verwendet.

4.2 Andere bekannte Konstruktionen, Verfahren und Schutzrechte, die für die Durchführung des Vorhabens benutzt wurden

Verfahren und Software zur Extraktion von Keypoints in Bildern. SURF-Verfahren zur Detektion und Beschreibung von fleckähnlichen Keypoints (blobs: markanten Punkten in Bildern natürlicher Szenen) (Bay 2008) und eine rechenzeitoptimierte Implementierung in C (Chen 2007), die unter der Voraussetzung weiterer Optimierungen eine gute Basis für Echtzeitimplementierung auf einem kompakten bordbasierten Rechner darstellte.

OCamCalib Toolbox zur Bestimmung von Kalibrierungsparametern einer Spiegelkamera. Diese MATLAB basierte Toolbox (OCamCalib 2012, Scaramuzza 2006) wurde als Grundlage für den entwickelten Kalibrierungsstand genommen und geeignet für Fischaugenkameras modifiziert.

5 Zusammenarbeit mit anderen Stellen

5.1 Zusammenarbeit der Verbundpartner

Eine wichtige Voraussetzung für Erfolg des Vorhabens stellt eine optimale Gestaltung der Zusammenarbeit zwischen den Verbundpartner dar, die einerseits auf einer klaren Definition und Abgrenzung von bearbeiteten Aufgabenbereichen basiert und andererseits durch ein aktives grenzübergreifendes Management, das System-Engineering und konsequente Verwendung relevanter technischer Mittel dafür sorgt, dass die Aktivitäten im Kontext des gesamten Projektes durchgeführt werden und somit das Erreichen der Projektziele in der angestrebten Systemsicht sichert.

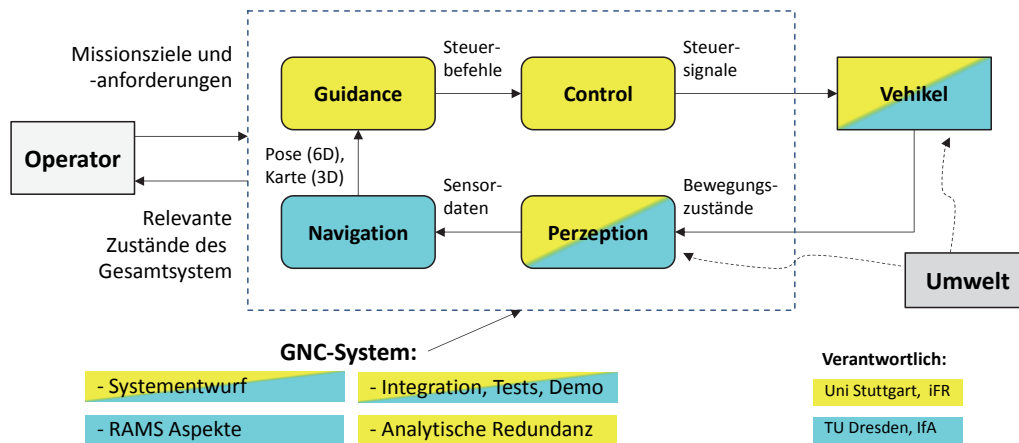


Abbildung 5.1 Aufgabenbereiche der Verbundpartner im gesamten Systemkontext.

Die von den Verbundpartnern eigenverantwortlich übernommenen Aufgabenbereiche sowie die gemeinsam vorgenommenen Tätigkeiten sind in Abbildung 5.1 vorgestellt. Die gemeinsamen Tätigkeiten beziehen sich auf die Aufgaben im systemtechnischen Kontext: Systementwurf, Aufbau der Flugplattform (mit Sensorik) sowie Integration, Tests und Demonstration des Gesamtsystems (Flugplattform und Bodensubsysteme).

Integrierter virtueller Engineering-Ansatz: Aufgrund der räumlichen Separation Stuttgart – Dresden waren die beiden Teams des iFR und IfA gefordert, für die vorliegende enge inhaltliche Verzahnung der Softwarefunktionen einen geeigneten Engineering-Ansatz zu etablieren. Diese Aufgabe wurde erfolgreich mittels eines integrierten virtuellen Engineering-Ansatzes gelöst. Ausgehend von definierten Projektstandards wurde ein gemeinsames internetbasiertes Repository mit Configuration Management Funktionalitäten für Software und Dokumente geschaffen. Regelmäßige Videokonferenzen (Skype) ermöglichten eine kontinuierliche, kurzfristige und insgesamt effiziente Abstimmung kritischer Entwicklungsfragen.

5.2 Zusammenarbeit mit Dritten

Institut für Flugmechanik und Flugregelung: Alle notwendigen Arbeiten wurden innerhalb der Universität Stuttgart erledigt, weshalb keine Beauftragung Dritter stattfand.

Institut für Automatisierungstechnik: Bei der TU Dresden wurden keine Aufträge an Dritte vergeben, da alle notwendigen Arbeiten innerhalb der Technischen Universität Dresden durchgeführt werden konnten.

6 Verwendung der Zuwendung und erzielt Ergebnis

6.1 Erarbeitung eines detaillierten Systemkonzeptes

Es wurden folgende wesentliche Ergebnisse durch Verbundpartner in Zusammenarbeit erzielt (in AP1):

- Definition von Systemanforderungen und Randbedingungen hinsichtlich Einsatzszenario, Systemfunktionen, Leistungen, Sicherheitsaspekten sowie Validierung,
- UML-basiertes High-Level-Systemmodell (inkl. Architekturen der Hardware und Software der Flugplattform und des Bodensegmentes) unter Berücksichtigung relevanter RAMS-Aspekte,
- Auswahl geeigneter algorithmischer Ansätze,
- Definition von Schnittstellen zu HiL- & SiL-Umgebungen für Tests und Verifikation von Hardware- und Softwarekomponenten am Boden,
- wichtige Konzepte für Tests (Testumgebungen, Abläufe und Vorgehensweisen).

6.2 Systemtestbench mit Flugplattform

Hauptfunktionalität und Testumgebung

Die Testbench mit Flugplattform (Projektcode SYS-FP-GNC) soll zu Test, Verifikation und Demonstration der entwickelten Autonomiefunktionen dienen und zwar in einer repräsentativen Testumgebung gemäß den angestrebten terrestrischen und extraterrestrischen Referenzszenarien. Die Referenzmission des Testbenches besteht aus den in Abbildung 6.1 vorgestellten Phasen, wobei die Flugphasen „Aufstieg“ und „Transferflug“ unmittelbar das terrestrische Szenario ansprechen und die letzte Flugphase „Landing“ auch für das extraterrestrische Landungszenario relevant ist.

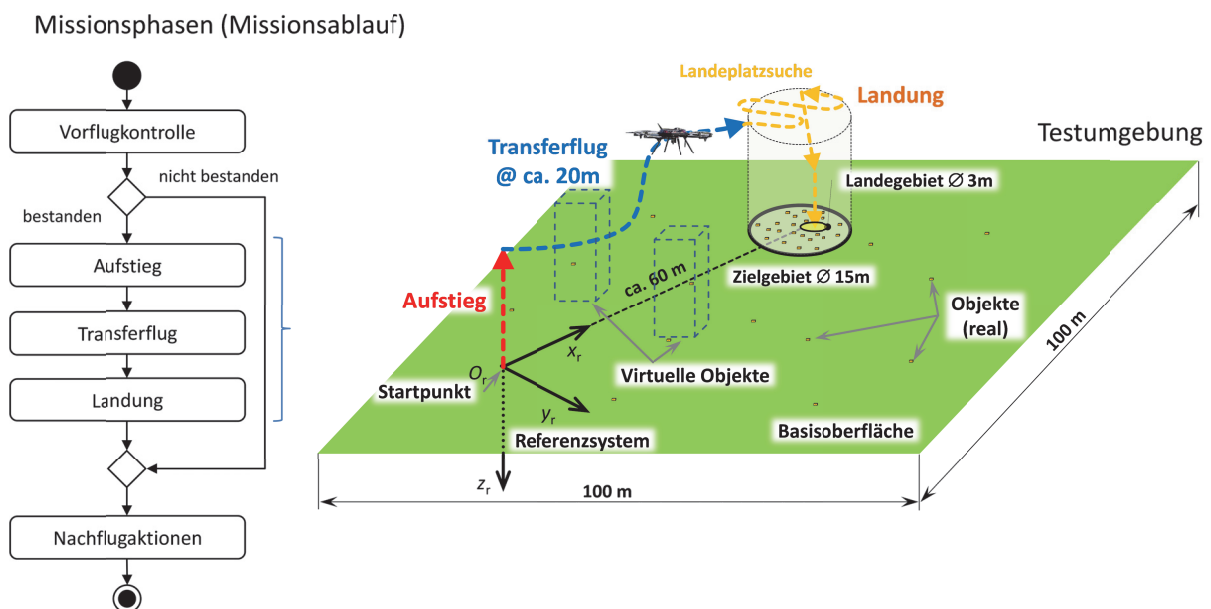


Abbildung 6.1 UML-Diagramm mit Hauptaktivitäten des Testbenches (Missionsablauf) und Testumgebung.

Die Testumgebung wird in freiem Gelände unter Verwendung von natürlichen Elementen, wie die Basisoberfläche, und künstlichen Elementen gestaltet. Die künstlichen Elemente umfassen kleine boxenartige Objekte, die vor allem als Hindernisse bei der Landung dienen sollen. Unter Beachtung

von Projekteinschränkungen wird die Verwendung von großen Hindernissen (20 m Höhe) für den Transferflug als virtuelle Objekte ausnahmsweise zugelassen.

Als Flugplattform wird ein Flugvehikel mit 6 Freiheitsgraden vom Typ Octocopter eingesetzt. Die Flugplattform stellt dabei ein hinreichend repräsentatives funktionelles Modell eines planetaren düsenangetriebenen Landfahrzeuges dar und kann somit eine gebietsübergreifende Interpretation der Ergebnisse ermöglichen. Eckdaten der Flugplattform: Rahmensystem Goliath der Firma Cadmic GmbH mit acht Rotoren und zwei LiPo-Akkus, äußerer Durchmesser ca. 1m, Startgewicht ca. 5,4 kg, maximaler Schub ca. 10 kg.

Hardwarearchitektur

Abbildung 6.2 zeigt ein internes UML-Blockdiagramm des Testbenches SYS-FP-GNC in SysML-Sprache mit hinzugefügten Fotos von wichtigen Elementen. Das Testbench beinhaltet sowohl bordbasierte als auch bodenbasierte Subsysteme und Komponenten. Die Zuordnung der speziellen englischen UML-Bezeichnungen ist in einer Konsistenzliste in Anhang gezeigt.

Die meisten Hardware-Komponenten des Systems wurden angeschafft und ggf. geeignet angepasst (in AP3 und AP4.2). Der Universelle Projekt-Rechner (UPC) ist ein handelsübliches Notebook (Laptop) mit Dual-Core Intel-Prozessor 2,6 GHz. Der Navigationscomputer (IfA) wurde auf Grundlage einer kompakten Entwicklungsplattform Pandaboard ES aufgebaut (Eckdaten: Prozessor TI OMAP4460, inkl. ARM Cortex-A9 Dual-Core 2x1,2 GHz, 2x NEON SIMD-Co-Prozessor, 1 GB DDR2-RAM, Leistung max. 4 W).

Die Komponenten des kompakten Bordrechners wurden hingegen auf Basis der Vorarbeiten des iFR entwickelt und angefertigt (in AP3.1) (Eckdaten: Prozessor TI OMAP3530, inkl. ARM Cortex-A8 720 MHz, 512 MB DDR-RAM, Leistung max. 1W; FPGA Xilinx Spartan 6, ca. 147.000 Logikzellen, 4 MB Blockram). Die inertielle Messeinheit (IMU) ist im Bordrechner eingebaut, da die Messdaten auch für die Flugregelung in schnellen Regelkreisen verwendet werden. Der GPS-Empfänger ermöglicht vor allem die Erstellung von „ground truth“ Daten für die Testauswertung.

Der Zusammenbau der Flugplattformen (AP4.2) und die Systemintegration (in AP4.3) wurden in Teams der Verbundpartner in Zusammenarbeit durchgeführt.

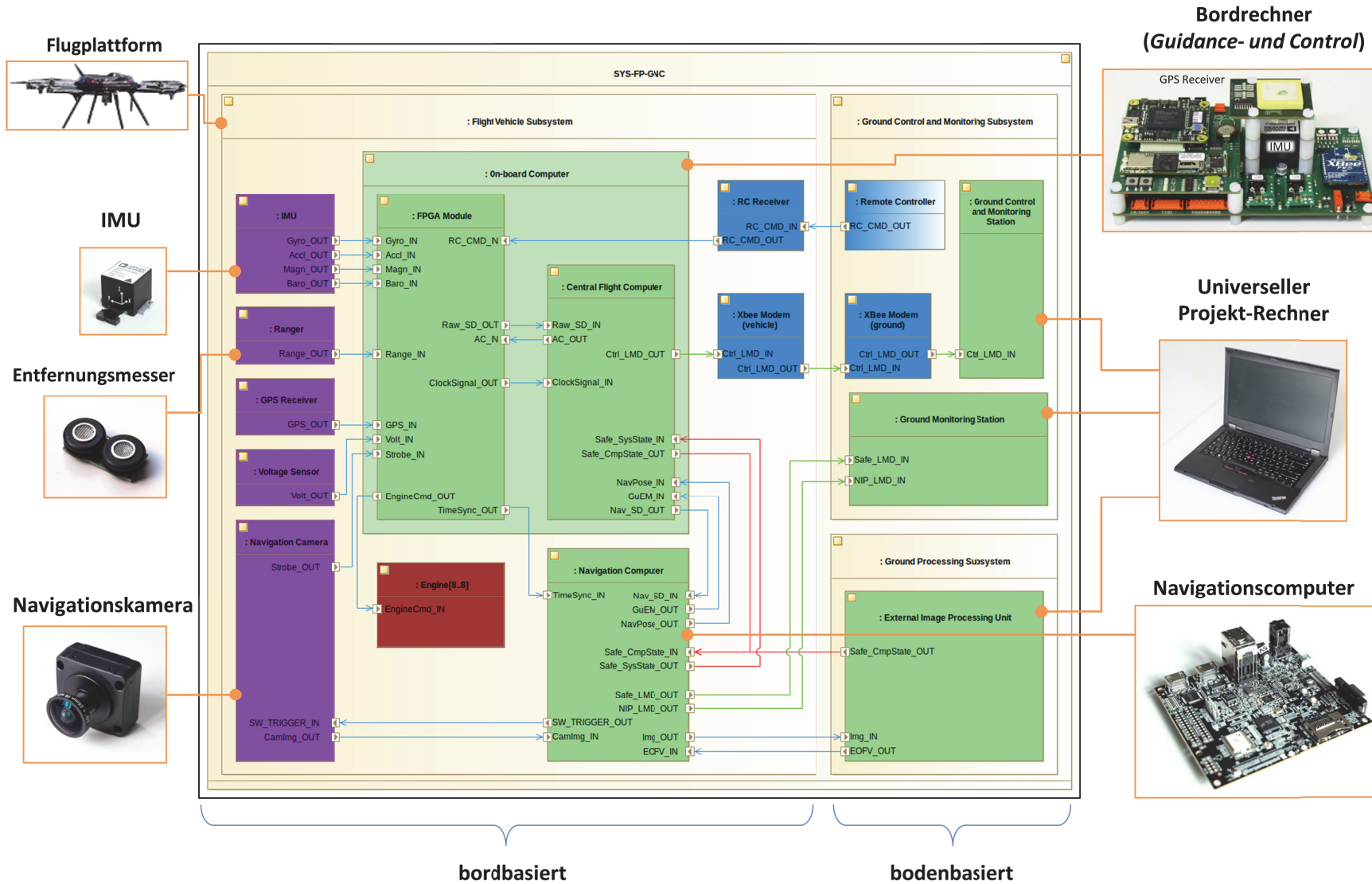


Abbildung 6.2 Internes Blockdiagramm von SYS-FP-GNC in SysML (mit Elektronik, Sensorik und Aktorik) und Fotos wichtiger Blöcke.

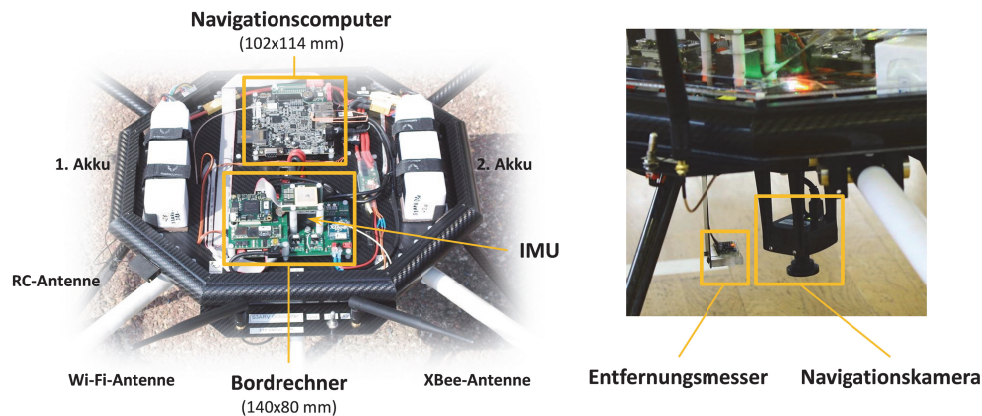


Abbildung 6.3 Kompakte Bordelektronik und Bordsensorik einer Flugplattform (RC-Empfänger auf der unteren Seite, die Motorregler in der ersten Einbauebene versteckt).

Softwarearchitektur

Die sechs grünen Blöcke in Abbildung 6.2 repräsentieren Rechereinheiten des Testbenches, die zum Teil unter verschiedenen Betriebssystemen laufen, die im Projekt entsprechend den Systemanforderungen angepasst und konfiguriert wurden (in AP3).

Der Schwerpunkt der Softwareentwicklung lag bei der Betriebssoftware der Rechereinheiten, vor allem bei der Implementierung der Guidance-, Navigation- und Control-Algorithmen (in AP2). Die Algorithmen der Guidance- und Control-Funktionen wurden im Bordrechner realisiert. Der Navigationscomputer stellte eine bordbasierte Rechnerplattform für die Algorithmen der Navigation dar. Für rechenaufwendige Berechnung des korrelationsbasierten optischen Flusses war die bodenbasierte Externe Bildverarbeitungseinheit auf dem Universellen Projekt-Rechner (Laptop) vorgesehen.

Speziell zu erwähnen sind die Software-Frameworks für die Algorithmen (AP3.3). Das Framework des Bordrechners für die Guidance und Control Algorithmen wurde auf Grundlage der Vorarbeiten vom iFR weiterentwickelt. Das Framework für die Algorithmen der bildgestützten Navigation erforderte vor allem intensive plattformübergreifende Übertragung von Bilddaten zwischen parallellaufenden Algorithmen und Überwachungsfunktionen und wurde im Projekt als eine Multi-Prozess/Multi-Rechner-Software über den TCP/IP/Wi-Fi-Netzwerkprotokollen entwickelt („Distributed Network based Framework“, DNF). Das DNF soll somit eine Grundlage für eine verteilte Betriebssoftware dreier verschiedener Rechereinheiten darstellen: Navigationscomputer, Bodenstation für Überwachung und Externe Bildverarbeitungseinheit.

Die auf dem Universellen Projekt-Rechner (Laptop) implementierten bodenbasierten Rechereinheiten des Testbenches stellen den Systemoperatoren außerdem Benutzerschnittstellen zur Verfügung.

Im Projekt wurden zwei bodenbasierten Rechereinheiten implementiert: die Bodenstation für Kontrolle und Überwachung (der Control- und Guidance-Funktionen) und die Bodenstation für Überwachung (der Navigationsfunktion und Systemsicherheit) (in AP3.4). Die Entwicklung der Externen Bildverarbeitungseinheit wurde aufgrund der angepassten Zielsetzung bzgl. der Verwendung des optischen Flusses auf wichtige wiederverwendbare Bildauswertungsfunktionen reduziert.

Wiederverwendung in anderen Testbenches

Wichtige Hardware- und Softwarekomponenten des Systemtestbenches können auch in laborbasierten SiL- und HiL-Testbenches wiederverwendet werden. Das SiL- und das HiL(PiL)-

Testbench für die Navigationsalgorithmen basieren zum Beispiel auf dem Navigationscomputer, der Bodenstation für Überwachung und der DNF-Software.

Die Hardwarekomponenten des Bordrechners (siehe Abbildung 6.2) können bis auf eine Interfaceplatine unverändert im HiL-Testbench verwendet werden. Softwarekomponenten des Bordrechners werden in dieser Testbench unverändert eingesetzt. Außerdem kann die Bodenstation zur Überwachung der Control- und Guidance-Algorithmen unverändert im HiL-Testbench eingesetzt werden.

Aufgebaute Flugplattformen

Es wurden insgesamt vier Flugplattform im Laufe des Vorhabens gemäß der im Projekt vorbereiteten Dokumentation aufgebaut.

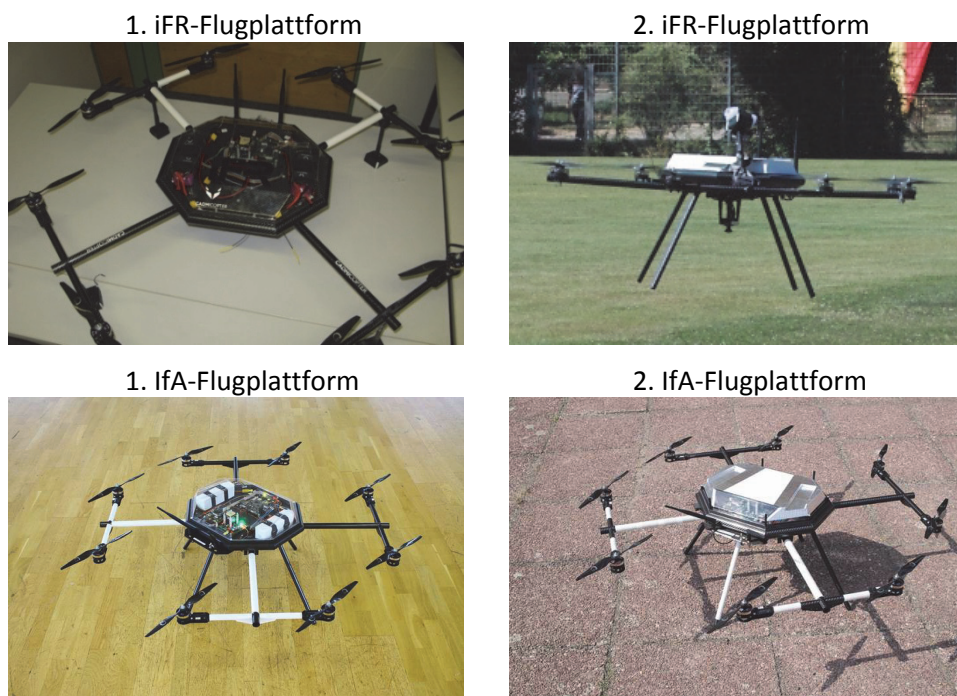


Abbildung 6.4 Im Laufe des Vorhabens aufgebaute Flugplattformen.

6.3 Systemtestbench mit Lander-Demonstrator

Hauptfunktionalität und Testumgebung

Der Zweck dieses Systemtestbenches (Projektcode SYS-TB-N) ist Test und Verifikation der entwickelten Algorithmen der optischen Navigation im Kontext der planetaren Landung. Die Funktionalität umfasst vor allem Erzeugung und Erfassung wiederholbaren Bewegungen eines Lander-Demonstrators in einer simulierten Umgebung zur Durchführung von Simulationsexperimenten in Echtzeit. Dieses laborbasierte Testbench wurde auf Basis des am IfA verfügbaren Satelliten-Rendezvous-Simulator „MiPOS“ (Miniature Proximity Operation Simulator, 6 DoF) aufgebaut. Es ermöglicht eine skalierte Simulation eines Landefluges in einer skalierten Testumgebung (in Modellmaßstab). Die Vorgehensweise nutzt bestimmte Skalierungsinvarianz der optischen Abbildung und der Drehratenmessung aus und kann im Rahmen von funktionellen Tests und Leistungsanalyse aussagenkräftige Ergebnisse liefern. Der Lander-Demonstrator mit der kompletten Bordelektronik und Bordsensorik der Flugplattform wird entlang einer Flugtrajektorie geführt, die zu dem als „ground truth“ für die Testauswertung gespeichert wird.

Architektur

Dieses Systemtestbench besteht aus dem Lander-Demonstrator, der Bodenstation für Überwachung und dem MiPOS-basierten Planetary Landing Testbench. In den zwei ersten Elementen wurden die jeweiligen Komponenten des Systemtestbenches SYS-FP-GNC wiederverwendet.

Der Lander-Demonstrator besteht aus dem Rumpf der Flugplattform mit der kompletten Elektronik und Sensorik und wird über dieselbe Datenschnittstelle an das Bodensegment angebunden.

Das Planetary Landing Testbench (PL-TB) wurde auf Basis des MiPOS durch eine Anpassung des Endeffektors (Rotationseinheit) und eine Erweiterung der Steuer-Software aufgebaut (in AP4.1). Der Endeffektor wurde für die Montage des Lander-Demonstrators umgebaut. Die Steuer-Software wurde mit erforderlichen Datenschnittstellen und Datenauswertungsfunktionen (dynamische Modellierung, Koordinatentransformation) erweitert. Die Eckdaten des PL-TB: maximaler Arbeitsraum ca. 2,5x1,5x0,8 m (für den Lander-Demonstrator); Tragfähigkeit bis 2,5 kg; nominale Messfehler der Flugtrajektorie bis 3 mm (translatorisch), 0,005° (rotatorisch) (1σ).

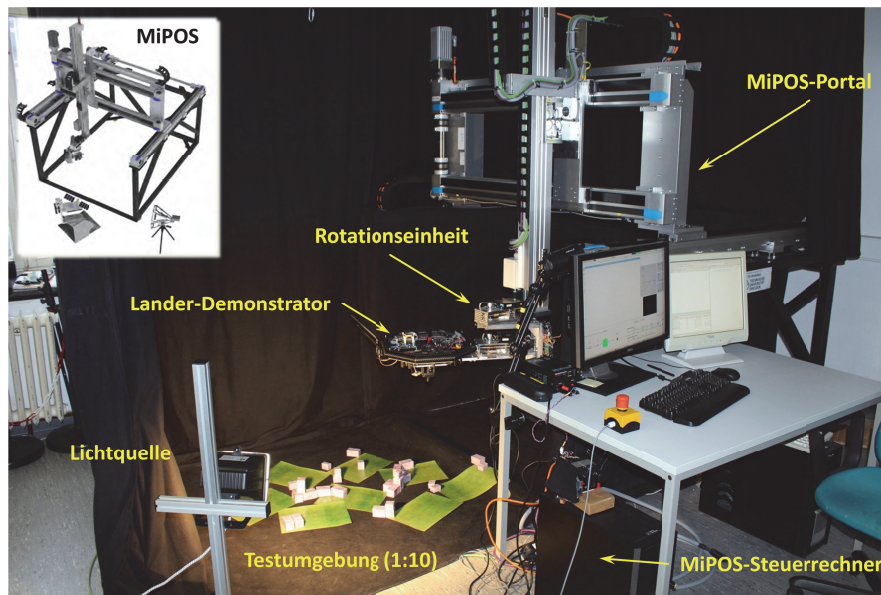


Abbildung 6.5 Planetary Landing Testbench mit Lander-Demonstrator („Testbench“ von AP4.1).

6.4 Perzeption

Als Referenzszenarien werden autonome Landungen in einem unbekanntem Umfeld (keine a priori Informationen) zugrunde gelegt. Damit das Gesamtsystem entsprechend der Gesamtzielsetzung in seiner Komplexität beschränkt bleiben kann, wurden folgende primäre Sensorkomponenten konzipiert: eine Mono-Kamera, eine Inertiale Messeinheit (dreiachsiger Drehratensensor, 3 Beschleunigungssensoren), sowie ein optischer Laser-Distanzsensor (LiDAR). Aus technologischer Sicht stehen diese Sensoren bereits jetzt zur Verfügung, sowohl in miniaturisierter Form für UAV Anwendungen (Katastrophenhilfe), als auch für Raumfahrtanwendungen in entsprechender Form. Zusätzlich wurden sekundäre Sensoren wie Luftdruckmesser und dreiachsiges Magnetometer betrachtet. Ein GPS-Receiver war zum Zwecke der Erstellung von Referenzdaten für Testauswertung vorgesehen.

Bis auf die Mono-Kamera und LiDAR wurden die Sensorkomponenten in früheren Arbeiten am iFR bereits eingesetzt, sodass hier nur die Hardware neu zu beschaffen und einzubauen war (AP3.2, iFR). Die elektrische und softwareseitige Anbindung wurden aus den Vorarbeiten übernommen.

In diesem Kapitel werden Sensorkomponenten erläutert, die im Projekt von IfA implementiert wurden bzw. in den IfA-Aufgaben verwendet wurden.

Mono-Kamera

Für die Mono-Kamera wurden das Kameramodell CMLN-13S2M-CS von Point Grey Research und das Fischaugenobjektiv Lensagon BF2M12520 ausgewählt (AP3.2, IfA). Es wurden nicht nur die erforderlichen optischen und photometrischen Leistungen, sondern Maße und Gewicht aber auch Protokolle für die Datenübertragung und Ansteuerung der Kamera berücksichtigt.

Die Kamera ist konfigurierbar und bietet unter anderem die Möglichkeiten für eine automatische Belichtungssteuerung. Bei den sorgfältig ausgewählten Einstellungen kann die Kamera bis zu ca. 30 Bilder mit 496x482 Pixel pro Sekunde liefern, die bei typischen Licht- und Bewegungsverhältnissen optimal beleuchtet sind und scharf bleiben.

An Bord der Flugplattform wurde diese Navigationskamera von unten in einer speziellen Plattform montiert (siehe Abbildung 6.3). Die Blickrichtung der Kamera stimmt mit der vertikalen Achse der Flugplattform überein. Ein sehr großer Öffnungswinkel vom Fischaugenobjektiv ermöglicht dabei im Nominalfall die Erfassung der kompletten Szene unter der Flugplattform.

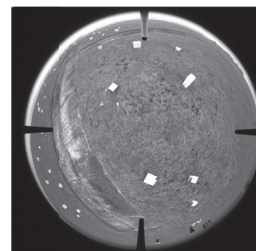
Die Navigationskamera wurde über die Datenschnittstelle USB 2.0 unter Verwendung der libdc1394-Bibliothek an den Navigationscomputer angeschlossen. Zusätzliche Signalleitungen für die Triggerung der Bildaufnahme und die Markierung der Bilder mit Zeitstempeln wurden dem Computer über eine speziell entwickelte Adapter-Elektronik zugeführt.

Es wurden erforderliche mathematische Modelle der Navigationskamera erstellt: Inverses Modell (aus Bild in Umwelt), Direktes Modell (aus Umwelt in Bild), sowie ein Kalibrierungsverfahren mit einem Kalibrierungsstand zur Bestimmung der Modellparameter entwickelt (unter Verwendung der MATLAB basierten OCamCalib - Toolbox, siehe Kapitel 4.2).



Kamera:
 CMLN-13S2M-CS
 max 1296x964 px @ 18 fps
 0,01 ms ... 10s automatisch
 44 x 25,5 x 41 mm
 37 Gram

Objektiv (Fischaugen):
 Lensagon BF2M12520
 Sichtfeld 185°
 14 Gram



Beispielbild (aus Flugdemo)

Abbildung 6.6 Navigationskamera.

Inertiale Messeinheit

Die Inertiale Messeinheit wurde mit dem „Ten Degrees of Freedom Inertial Sensor“ ADIS16407 implementiert (AP3.2, iFR). Das kompakte Modell ermöglicht vor allem die dreiachsige Erfassung der Drehraten und der Beschleunigungen und beinhaltet noch zusätzliche Sensoren zur dreiachsigen Messung des Magnetfeldes (Magnetometer) und zur Messung des Luftdruckes (Luftdruckmesser).

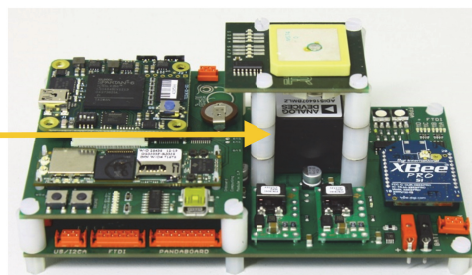
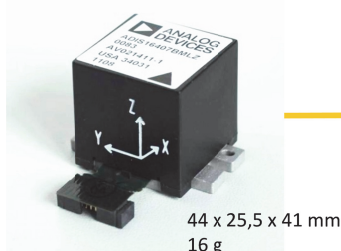


Abbildung 6.7 Sensor ADIS16407 und Einbau im Bordrechner.

6.5 Navigation

6.5.1 Ansatz zur autonomen bildbasierten Navigation

Der Navigationsalgorithmus soll vor allem eine autonome Lokalisierung des Fluggerätes in einer unbekanntem Umgebung sicherstellen. Das zweite Ziel ist eine autonome Erstellung von Umgebungsmodellen (Karten). Ausgangspunkte hierfür sind vor allem die entsprechenden Referenzszenarien. Desweiteren müssen bei der Algorithmenentwicklung insbesondere die relevanten Soft- und Hardwareanforderungen berücksichtigt werden. Resultierend aus den Aspekten „small“, „safe“ und „space“ wirken sich diese Anforderungen speziell in Form begrenzter Sensorik und Rechenleistung aus.

Im Projekt wurde die Entwicklung der optischen Navigation auf folgendem zweistufigen Ansatz basiert (Abbildung 6.8).

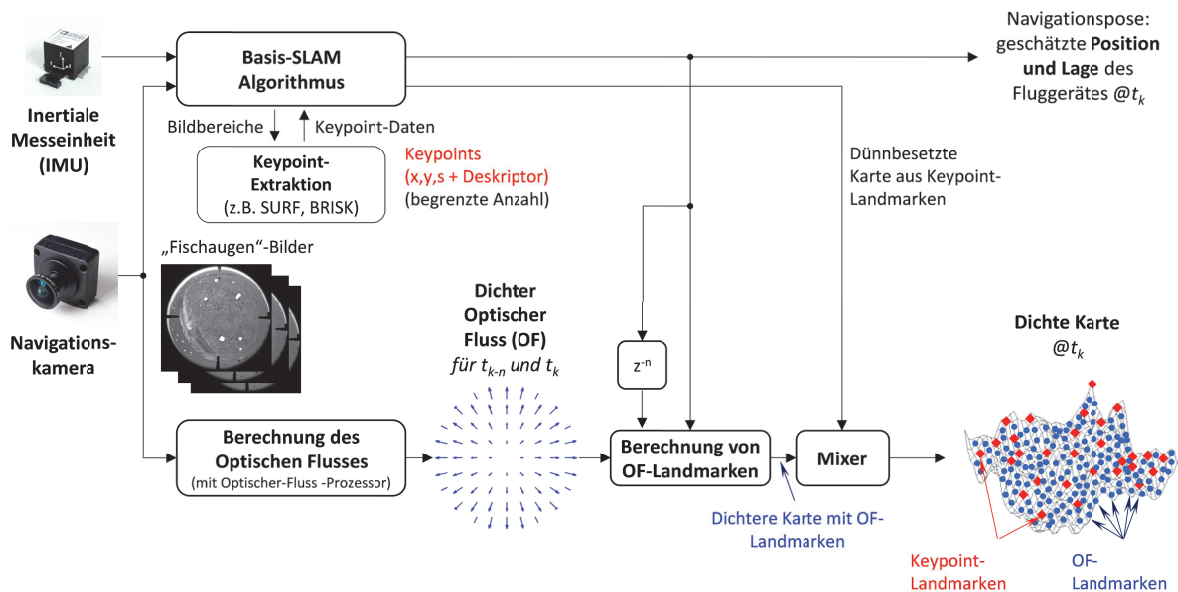


Abbildung 6.8 Zweistufiger Ansatz zur autonomen optischen Navigation.

Die erste Stufe besteht aus einem keypointbasierten Navigationsalgorithmus, der vor allem Position und Lage des Fluggerätes autonom unter Verwendung von Daten einer Kamera und einer inertialen Messeinheit liefert. Außerdem führt der Algorithmus simultane Bestimmung von Landmarken (3D-Punkten) der Umgebung durch. Daher zählt der Algorithmus zur Gruppe von SLAM-Algorithmen und wird hier als Basis-SLAM-Algorithmus bezeichnet.

Der Basis-SLAM-Algorithmus wird unter Beachtung der limitierten Rechenleistung auf eine begrenzte Anzahl von Keypoints ausgelegt („small“). Dadurch wird die Menge der bestimmten Landmarken reduziert, was in einer relativ dünn besetzten Umgebungskarte resultieren kann.

Für eine sichere Landung in komplexen Umgebungen mit kleinen Hindernissen kann jedoch eine dichtere Karte erforderlich sein („safe“). Um den damit verbundenen erhöhten Rechenaufwand zu begrenzen, wird die Bestimmung der dichteren Karte in einem begleitenden Prozess reduzierter Komplexität implementiert (2. Stufe). Als Ausgangsdaten für die Erstellung dichter Karten werden die Vektoren des optischen Flusses verwendet, welche in einem dedizierten Optischer-Fluss-Prozessor mit einer hinreichend hohen Auflösung aus Kamerabildern bestimmt werden können. Die Landmarken des optischen Flusses (OF-Landmarken) werden mithilfe der bestimmten Lage und Position mit einem reduzierten Rechenaufwand berechnet („small“). Die endgültige Karte wird anschließend aus den Landmarken der beiden Typen zusammengesetzt. Der gesamte Navigationsalgorithmus kann auch als Erweiterter SLAM-Algorithmus bezeichnet werden.

Im Projekt wurde der primäre Teil des vorgestellten Ansatzes untersucht: der Basis-SLAM Algorithmus. Die Entwicklung der 2. Stufe (mit dem optischen Fluss) wurde aufgrund der angepassten Zielsetzung (Fortschrittgespräch Nr. 4) auf Umsetzung wichtiger wiederverwendbarer Bildauswertungsfunktionen bei Berechnung des optischen Flusses reduziert.

6.5.2 Basis-SLAM Algorithmus

Die primäre Aufgabe des Basis-SLAM Algorithmus besteht in der Bestimmung der Navigationspose mit einer für die Flugplanung und -regelung erforderlichen hohen Datenrate (100 Hz bei der Flugplattform). Die erforderliche Navigationspose enthält Position, Geschwindigkeit und Lage der Flugplattform in einem festgelegten Referenz Bezugssystem: Referenzsystem für Flugführung und Navigation $\{r\}$. Darüber hinaus soll der Algorithmus auch eine Karte aus Landmarken im Referenzsystem bestimmen.

Das Referenzsystem $\{r\}$ wird in der Projektlösung als ein kartesisches Rechtssystem mit dem Ursprung auf dem Startpunkt der Flugplattform bestimmt. Die Achsen des Referenzsystems werden auf Grundlage des körperfesten Systems des Fluggerätes am Startpunkt sowie der z-Achse vom geodätischen System auf dem Startpunkt definiert. Die z-Achse stimmt dabei mit der z-Achse des geodätischen Systems überein. Die Flugplattform wird am Startpunkt so ausgerichtet, dass die x-Achse ungefähr zum Zentrum des Zielgebietes zeigt (siehe Abbildung 6.1).

Bei reinen kamerabasierten SLAM-Ansätzen (Mono-SLAM) werden die Ergebnisse ausschließlich unter Verwendung der Kamerabilder produziert und unterliegen somit einer Reihe von prinzipiellen Einschränkungen. So werden sie in einem Koordinatensystem bestimmt, das lediglich mit den Koordinaten der mitgeschätzten Landmarken verknüpft wird. Ein Bezug auf externe Referenzen, wie z.B. die z-Achse des geodätischen Systems, ist nicht möglich (Lot-Problem). Zudem können die ortsbezogene Daten der Pose und der Karte nur bis auf einen Skalierungsfaktor bestimmt werden (Skalierungsproblem). Beide Probleme können jedoch im Kontext der autonomen Navigation (d.h. ohne Nutzung externer Datenquellen bzw. Infrastrukturen) gelöst werden. Ein möglicher Lösungsansatz beruht dabei auf Verwendung von inertialen Messdaten (Winkelgeschwindigkeiten und Beschleunigungen). Ein 3-DOF-Beschleunigungsmesser erfasst auch die Gravitationsbeschleunigung mit und kann zur Rekonstruktion der z-Achse verwendet werden. Die Beschleunigungsdaten stellen weiterhin eine Grundlage zur Bestimmung der kombinierten Navigationslösung mit einer normalen Skalierung, d.h. im metrischen System, dar.

Eine weitere Randbedingung bei diesem kombinierten SLAM-Ansatz besteht in unterschiedlichen Raten der Bilder und der inertialen Messdaten, die im Kontext einer angestrebten rechenaufwandminimalen Lösung („small“) optimal berücksichtigt werden soll.

Unter Beachtung dieser Anforderungen und Randbedingungen wurde der Basis-SLAM Algorithmus als Kombination von zwei parallel laufenden Algorithmen gestaltet (Abbildung 6.9).

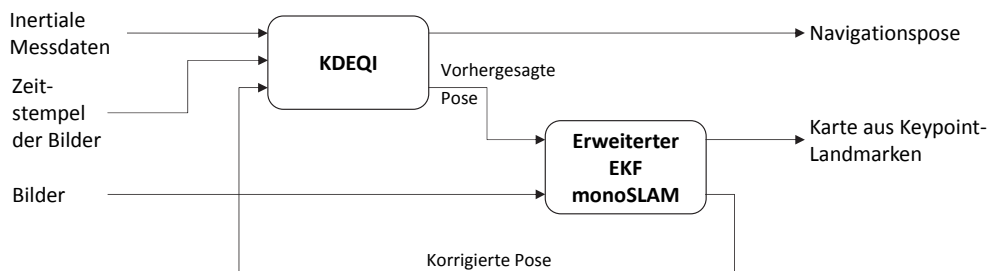


Abbildung 6.9 Prinzipielle Struktur des Basis-SLAM Algorithmus.

Die Eingangsdaten, die Zwischendaten und die Ausgangsdaten werden im Basis-SLAM Algorithmus mit Zeitstempeln markiert. Das ermöglicht Sicherstellung der zeitlichen Konsistenz aller Daten auch bei asynchronen und unterschiedlich schnellen Prozessen der Datenlieferung und der Datenauswertung, was im Basis-SLAM Algorithmus berücksichtigt wurde.

Der erste Navigationsalgorithmus KDEQI ist ein Integrator von Kinematischen Differenzialgleichungen (Kinematic Differential Equation Integrator). Der Algorithmus bestimmt den kompletten Bewegungszustand der Flugplattform (Positionsvektor, Geschwindigkeitsvektor, Lagequaternion, Winkelgeschwindigkeitsvektor) in $\{r\}$ primär aus den inertialen Messdaten (Winkelgeschwindigkeitsvektor, Beschleunigungsvektor) mit einer erforderlichen hohen Datenrate und liefert einen Teilzustand als die Navigationspose an die Funktionen Guidance und Control. Die Datenrate wird durch die Rate der inertialen Messdaten definiert.

Die Anfangswerte für den Integrator werden am Startpunkt bestimmt bzw. festgelegt. Zur Fehlerbegrenzung wird der Algorithmus zu regelmäßigen Zeitabständen initiiert (zurückgesetzt). Dies geschieht unter Verwendung der korrigierten Pose aus dem anderen Algorithmus.

Die Komplexität des KDEQI ist relativ gering, so kann der Algorithmus endliche und kleine Rechenzeiten („safe“) auch mit limitierten Rechenressourcen („small“) sicherstellen.

Der zweite Navigationsalgorithmus Erweiterter EKF monoSLAM repräsentiert eine erweiterte optische Navigation mit Landmarken, bei welcher das Lot-Problem und das Skalierungsproblem dadurch gelöst werden, dass die Zustandspropagation mithilfe des in $\{r\}$ arbeitenden KDEQI erfolgt (vorhergesagte Pose in Abbildung 6.9). Der Algorithmus liefert die Karte aus den Landmarken im Referenzsystem $\{r\}$ (inkl. Qualität der Landmarken) und bestimmt außerdem die korrigierte Pose für das Zurücksetzen des KDEQI.

Die Komplexität des Erweiterten EKF monoSLAM ist verhältnismäßig hoch, daher können die Ergebnisse mit einer geringeren Datenrate als bei KDEQI bereitgestellt werden. Endliche Rechenzeiten sowie Echtzeiteigenschaften sollen beim Algorithmus vor allem durch den rekursiven EKF-Ansatz mit einer begrenzten Anzahl der Landmarken sichergestellt werden.

EKF basierte SLAM-Algorithmen stellen einen Teil von aktuellen Forschungsaktivitäten auf dem Gebiet der optischen Navigation dar. Die EKF-Algorithmen besitzen zwar ein gutes Potenzial für eine Echtzeitimplementierung in kompakten Rechnern, aber benötigen in der Regel eine Robustifizierung mit zusätzlichen Maßnahmen, bei welchen Minimierung des zusätzlichen Rechenaufwandes eine große Rolle spielt. Ein Lösungsansatz (Civera 2010) verwendet eine elegante Kombination aus dem echtzeitfähigen EKF-Ansatz und dem robusten RANSAC-Ansatz. Das Erzeugen einer RANSAC-Hypothese aus einer Stichprobe von Keypoint-Matches wird mithilfe der Aktualisierung des Zustandes im EKF-Filter durchgeführt. Dadurch lässt sich die Anzahl von Datenpunkten in der Stichprobe auf eins festlegen, was eine Reduzierung der Anzahl von RANSAC-Iterationen ermöglicht. Die endgültige Aktualisierung des EKF-Zustandes wird anschließend ähnlich dem LO-RANSAC-Prinzip durchgeführt (LO-RANSAC: local optimised RANSAC), was zur Verbesserung der Schätzgenauigkeit beitragen kann. Aufgrund von vielversprechenden Echtzeitleistungen wurde dieser Ansatz als Basis für die EKF-basierte optische Navigation in S3ARV genommen.

Speziell wurde der Erweiterte EKF monoSLAM unter Verwendung eines Mono-SLAM-Algorithmus aufgebaut, welcher in derselben Veröffentlichung (Civera 2010) beschrieben wurde. Der Originalalgorithmus verwendet dabei eine perspektivische Kamera und schätzt die Pose sowie die Landmarkenkarte im aktuellen Kamerabezugssystem. Dadurch kann die Fortpflanzung der Schätzfehler minimiert werden. Die geringeren Schätzfehler haben einen positiven Einfluss auf Funktion des EKF, in dem sie für eine Begrenzung der Unsicherheiten bei Verwendung des Messmodells des Filters sorgen und die Wirkung der Linearisierungsfehler (in Jacobi-Matrizen) reduzieren.

Im Erweiterten EKF monoSLAM wurden im Vergleich mit dem Original-Algorithmus Anpassungen in folgenden Bereichen durchgeführt:

- Initialisierung von Keypoints mit einem Modell einer Kamera mit Fischaugenobjektiv,
- neues Messmodell für eine Kamera mit Fischaugenobjektiv,
- Jacobi-Matrix für das neue Messmodell,
- Koordinatentransformationen der Zustandsgrößen (mit Bezug auf das Referenzsystem {r}),
- SURF-Detektor als Keypoint-Detektor (Detektion von natürlichen Keypoints „blobs“),
- Beschreibung der Keypoints mit SURF-Deskriptoren anstatt Bildsegmente.

Außerdem wurde der Erweiterte EKF monoSLAM auf die Auswertung der Daten mit Zeitstempeln ausgelegt, was seine Verwendung im datengetriebenen Basis-SLAM mit asynchronen Prozessen ermöglichen kann.

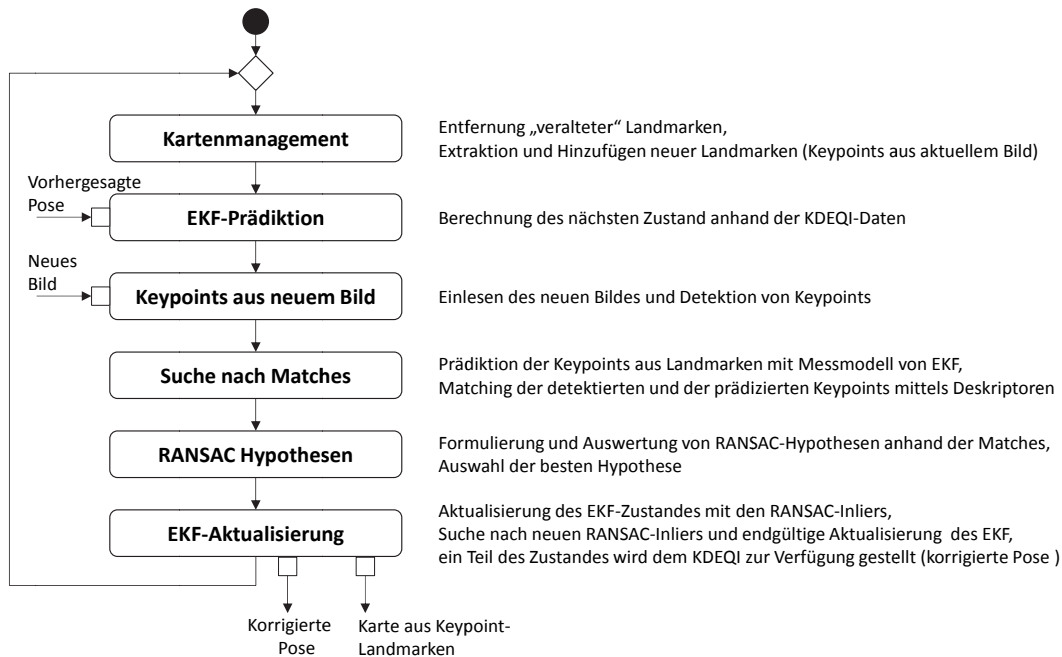


Abbildung 6.10 Prinzipielles Aktivitätsdiagramm für Erweiterten EKF monoSLAM.

Die Implementierung des Basis-SLAM Algorithmus wurde als eine MATLAB-Version und eine C++-Version vorbereitet. Die MATLAB -Version wurde im ersten Schritt unter Verwendung der durch die Autoren (Civera 2010) veröffentlichten Matlab-Version des Original-Mono-SLAM-Algorithmus entwickelt und getestet. Im nächsten Schritt wurde die C++-Version vorbereitet.

6.5.3 Laborbasierte Testbenches für die Navigationsalgorithmen

Für Tests der Navigationsalgorithmen wurden folgende laborbasierte Testbenches vorbereitet (Abbildung 6.11).

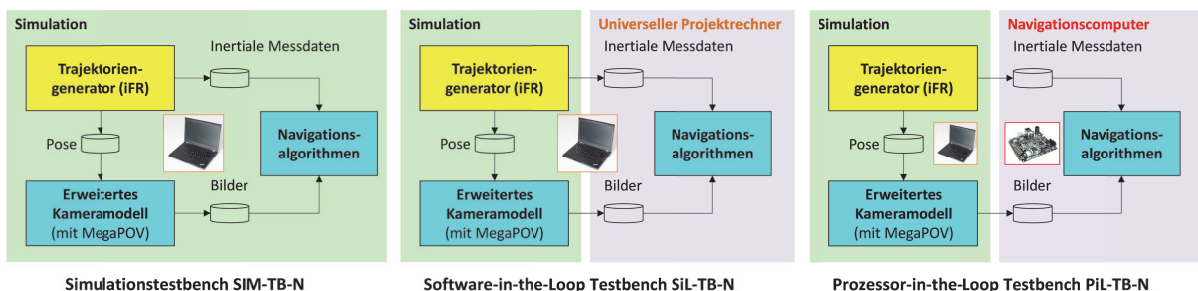


Abbildung 6.11 Laborbasierte Testbenches für Navigationsalgorithmen im Überblick.

Das Simulations-Testbench SIM-TB-N wurde unter MATLAB/Simulink in zwei Varianten implementiert: für MATLAB-Implementierung und C++-Implementierung der Navigationsalgorithmen und diente vor allem für funktionelle Tests. Die Bilder wurden dabei mit einem Erweiterten Kameramodell generiert. Das Erweiterte Kameramodell wurde unter MATLAB auf Basis vom Open-Source-Bildrenderer MegaPOV entwickelt. Die inertialen Messdaten und die Pose wurden mithilfe eines Testbenches des iFR zur Verfügung gestellt. Die Navigationsalgorithmen wurden dabei in das Framework vom Simulink über spezielle Schnittstellenblöcke (S-Funktionen) eingebettet.

Das Software-in-the-Loop Testbench SiL-TB-N diente für die ersten Echtzeittests und Optimierung der Algorithmen (C++) und wurde auf Basis des DNF-Frameworks aufgebaut. Die Navigationsalgorithmen wurden dafür in das spezielle dafür entwickelte DNF-Framework (siehe Kapitel 6.2) eingebettet und können dabei auf dem bodenbasierten Universellen Projektrechner (Laptop) mit denselben Eingangsdaten wie bei SIM-TB-N getestet werden.

Im Prozessor-in-the-Loop Testbench PiL-TB-N können die Algorithmen (C++) auf dem Navigationscomputer („Prozessor“) im DNF-Framework mit denselben Daten ausgeführt, getestet und hinsichtlich endgültiger Echtzeitleistungen optimiert werden.

Es bestand auch die Möglichkeit, die während des Fluges gewonnen Daten zu speichern und in späteren Experimenten in den laborbasierten Testbenches zu verwenden.

6.5.4 Test und Verifikation

Die Erprobung der Navigationsalgorithmen mit simulierten Bildern und inertialen Messdaten in der MATLAB/Simulink-Umgebung hat algorithmische Leistungen demonstriert, die den Einsatz der Algorithmen im angestrebten GNC-System ermöglichen können.

Abbildung 6.12 zeigt Ergebnisse eines Simulationsexperimentes: Referenztrajektorien (rot); geschätzte Trajektorien der translatorischen Bewegung (auf Achsen x_r , y_r , z_r des Referenzsystem $\{r\}$, blau) und der rotatorischen Bewegung (Eulerwinkel in $\{r\}$).

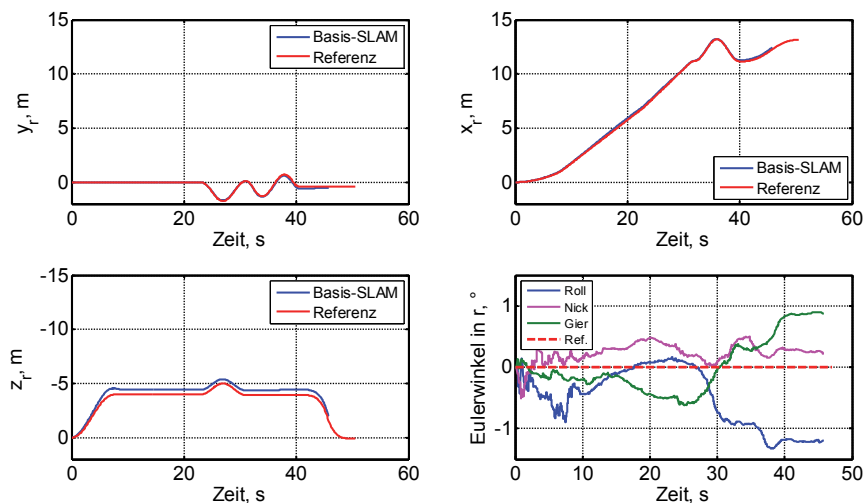


Abbildung 6.12 Navigationsleistungen mit simulierten Daten (Experiment in SIM-TB-N).

Snapshot aus der Simulation

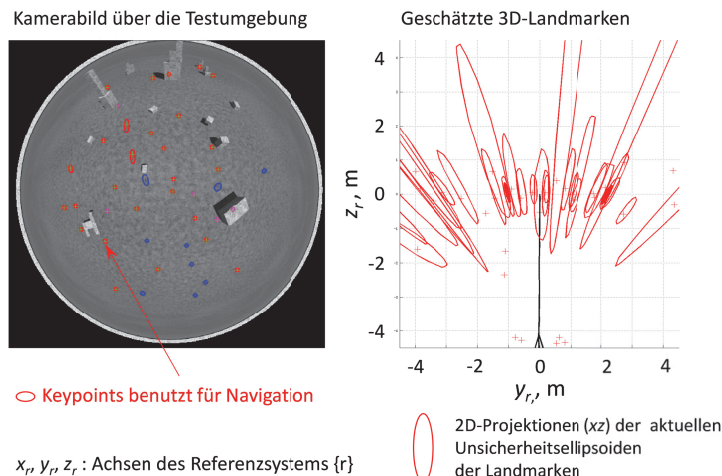


Abbildung 6.13 Snapshot aus der Simulation.

Es wurden Experimente mit realen Bildern und inertialen Messdaten durchgeführt. Die Daten wurden mithilfe der entwickelten Flugplattform im Flug in einer freien Testumgebung mit künstlichen Elementen (simulierte Felsen und Steinbrocken) gewonnen. Die Experimente bestätigten die prinzipielle Machbarkeit der optischen Navigation im Rahmen des ausgewählten Basis-SLAM-Ansatzes. Es wurde außerdem festgestellt, dass die aktuelle Umsetzung des Navigationsansatzes Limitierungen im Bereich der Robustheit aufweist, welche vor allem durch die Problematik der asynchronen Lieferung der Messdaten im verwendeten Testsystem begründet sein können. Die Bilder und die inertialen Messdaten werden in der Kamera und in der IMU der Flugplattform unabhängig voneinander erfasst und durch ein komplexes System aus Hardware- und Softwarekomponenten des Bordrechners und des Navigationscomputers zu Komponenten der Navigationssoftware weitergeleitet. Die dabei (sporadisch) entstehenden Pausen und gegenseitigen Verzögerungen haben negativen Einfluss auf die Funktion der Navigationsalgorithmen, die dank der rekursiven Natur zwar rechenzeitoptimal sind aber solche Störungen auch mit den speziellen entwickelten Kompensationsmechanismen nur in einem begrenzten Maße tolerieren können. Zur Verbesserung können nicht nur weitere dedizierte algorithmische Anpassungen vorgenommen werden, sondern kann auch eine Echtzeit-Optimierung der gesamten Datenübertragungskette im System erfolgen.

6.5.5 Echtzeitbildverarbeitung

Es wurden zwei Klassen der Bildverarbeitungsalgorithmen definiert, die für den verwendeten Navigationsansatz erforderlich sind: Keypoint-Extraktion und Optischer Fluss.

Bei Verfahren im Bereich der Keypoint-Extraktion wurde auf den Stand der Technik zurückgegriffen und versucht eine ausgewählte Lösung hinsichtlich der Rechenleistungen zu optimieren. So wurde eine Implementierung von SURF-Verfahren in C (Chen 2007) primär unter Verwendung von spezifischen Parallelisierungsmechanismen des Navigationscomputers verbessert und im Flug erprobt (Abbildung 6.14).

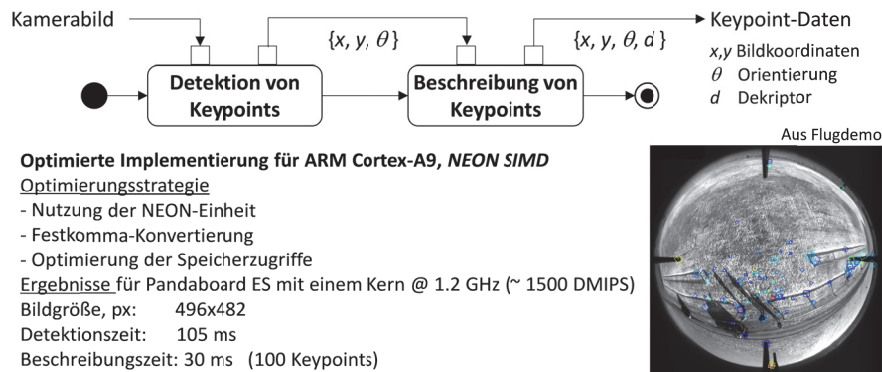


Abbildung 6.14 SURF-Keypoint-Extraktion optimiert für Pandaboard ES.

Im Bereich des Optischen Flusses wurde ein am IfA entwickeltes Verfahren für die Verwendung in der geplanten Externen Bildverarbeitungseinheit umgesetzt. Beim Verfahren handelt es sich um korrelationsbasierte Berechnung des optischen Flusses und eine C++-Implementierung für Intel-Prozessoren unter Verwendung mehrerer Prozessorkerne und SIMD-Einheiten. Die optimierte Implementierung kann in Echtzeitanwendungen im Rahmen der Externen Bildverarbeitungseinheit zur Berechnung der dichten Umgebungskarte eingesetzt werden (Intel Core2 Duo E8400, 3GHz, SSE4, zwei Kerne, Korrelationssegment 16x16 px: ~6400 Segmente/s; bei Bildgröße 496x480 px: ca. 7 Bilder/s).

Durch ein spezielles optoelektronisches Design kann auf Basis eines optischen Korrelators ein extrem leistungsfähiger kompakter und raumfahrttauglicher Optischer-Fluss-Prozessor realisiert werden (aktuelles Ziel bei IfA: 50.000 Segmente/s bei Baugröße 50x20x8 mm, Masse 20 g, Leistung 2W), wie es für autonome visuelle Navigationssysteme benötigt wird.

6.6 RAMS-Analysen

Der Aspekt Sicherheit („safe“) ist auf das Engste mit den Aspekten Systemverfügbarkeit und -zuverlässigkeit verknüpft (engl. RAMS - Reliability, Availability, Maintainability, Safety). Die in diesem Projekt angestrebten Architekturlösungen zeichnen sich primär durch eingebettete Softwarelösungen in einer geräteminimalen Hardwareumgebung.

Zu den wichtigsten RAMS-Aspekten zählen vor allem Fehlererkennung und Redundanz, wobei bei kleinen Systemen auf eine optimale funktionale Redundanz anstelle reiner Hardwareredundanz zurückgegriffen werden muss. Diese Themengebiete sowie relevante Einflüsse auf die Softwarearchitektur wurden im Kontext einer optimalen Kapselung von ausfallkritischen Softwareeinheiten und im Kontext von Erkennung und Behandlung von kritischen Fehlern untersucht.

6.6.1 Optimale Kapselung von ausfallkritischen Softwareeinheiten

Das unmittelbare sicherheitsrelevante Ziel der optimalen Kapselung besteht in Erhöhung der funktionellen Zuverlässigkeit („safe“) bei gleichzeitiger Minimierung des Hardwareaufwandes („small“).

Einige negative Umwelteinflüsse wie Temperaturerhöhung, Abfall der Versorgungsspannung und kosmische Strahlung, können Ausfälle in Rechnerhardware verursachen. Diese lokalen Ausfälle können zur Verfälschung von verarbeiteten Daten führen und somit das Entstehen inkorrektur Ergebnisse am Systemausgang hervorrufen. Das Prinzip wird in Abbildung 6.15 verdeutlicht. Bei einer Umweltstörung wird hier eine Funktion des Datenspeichers beeinträchtigt, was zu einer unerwünschten Änderung des gespeicherten Wertes einer Variable führt und damit einen Fehler verursacht, der bei der Verarbeitung der Daten in den Softwarefunktionen zu einem fehlerbehafteten kritischen Systemausgang propagiert wird.

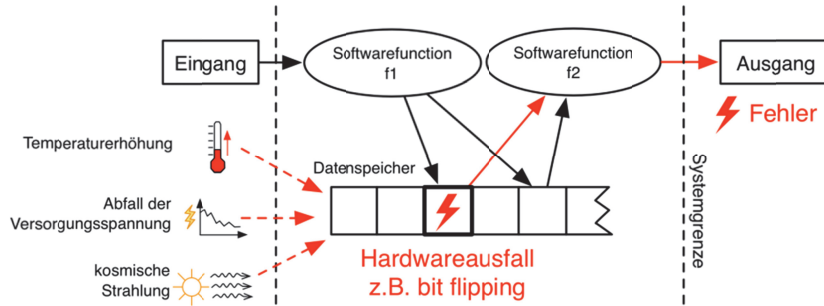


Abbildung 6.15 Erklärung zum lokalen Hardwareausfall und dessen Ergebnissen.

Zum Beispiel wurde ein durch Sonnenstrahlung verursachter Speicherdefekt zur Ursache für das Versagen der russischen Mission „Phobos Grunt“ (Februar 2012).

Der traditionelle Ansatz zur Sicherstellung einer hohen Systemverfügbarkeit und einer hohen Zuverlässigkeit bei solchen Ereignissen wie bit flipping verwendet bei sicherheitskritischen Systemen speziell geschützte Hardwarekomponenten. Diese Schutzmaßnahmen erfordern zusätzliche Hardwareressourcen und erschweren somit den Einsatz in geräteminimalen Umgebungen, die bei diesem Vorhaben angestrebt werden.

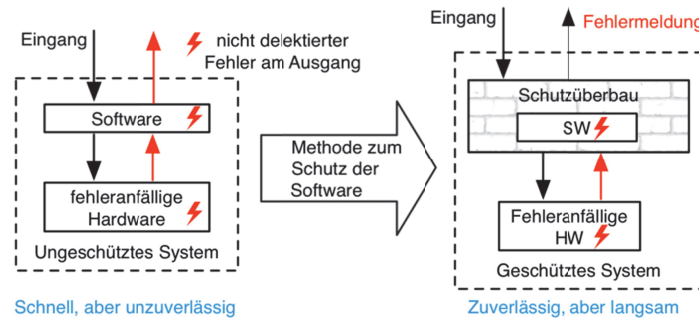


Abbildung 6.16 Prinzip des softwarebasierten Schutzes unter Verwendung von SFD (im Bild wird ein durch Hardwareausfall verursachter Datenfehler auf der Softwareebene detektiert und gemeldet).

Eine attraktive alternative Lösung stellt die Anwendung von Ausfalldetektoren dar, die ausschließlich in der Software implementiert werden (engl. software-implemented hardware fault detectors, SFD). Wichtige Beispiele sind SWIFT, SWIFT ECF, Software Encoded Processing (AN, ANB, ANBD-codes). Prinzipiell ermöglichen SFD die Verwendung kompakter und evtl. kostengünstiger Hardware mit reduzierten Zuverlässigkeitseigenschaften bei gleichzeitiger Aufrechterhaltung der Systemzuverlässigkeit auf erforderlichem Niveau (Abbildung 6.16). Allerdings führt die Anwendung der SFD zur Erzeugung vom zusätzlichen Quellcode, was einen beachtlichen zusätzlichen Rechenaufwand (Overhead) bedeuten kann und eine Minderung der Leistungen (z.B. Laufzeiteigenschaften) zur Folge haben kann. Der im Projekt entwickelte Ansatz (Abbildung 6.16) bezweckt eine Minimierung der SFD-bedingten Leistungsminderung bei gleichzeitiger Aufrechterhaltung einer hohen Zuverlässigkeit des Systems („small“ und „safe“). Es wurde gezeigt, dass eine selektive und balancierte Anwendung der SFD ausschließlich auf die kritischsten Teile der Software in einem effizienten Systemdesign mit minimaler Fehlerfortpflanzung resultieren kann.

Bei diesem Ansatz wird eine Kombination von Methoden der Zuverlässigkeitsanalyse und der Leistungsanalyse verwendet. Die Basis stellen dabei eine erweiterte Version eines Dual-Graph-Fehlerfortpflanzungsmodells und zeitdiskrete Modelle der Markov-Ketten dar, die in früheren Arbeiten am IfA eingeführt wurden. Mit der Methode der Zuverlässigkeitsanalyse wird mittlere Anzahl von nicht detektierten Fehlern an kritischen Systemausgängen bestimmt. Die Methode der Leistungsanalyse evaluiert mittlere Ausführungszeit von Systemfunktionen. Beide Methoden werden bei einer quantitativen Untersuchung verschiedener Strategien der selektiven Anwendung der SFD

eingesetzt und ermöglichen somit die Bestimmung der Balance zwischen den Systemleistung und der Systemzuverlässigkeit.

Der Ansatz wurde auf die Software des Eingebetteten Flugreglers (Teil des Bordrechners) angewandt, getestet und demonstriert (in AP2.5).

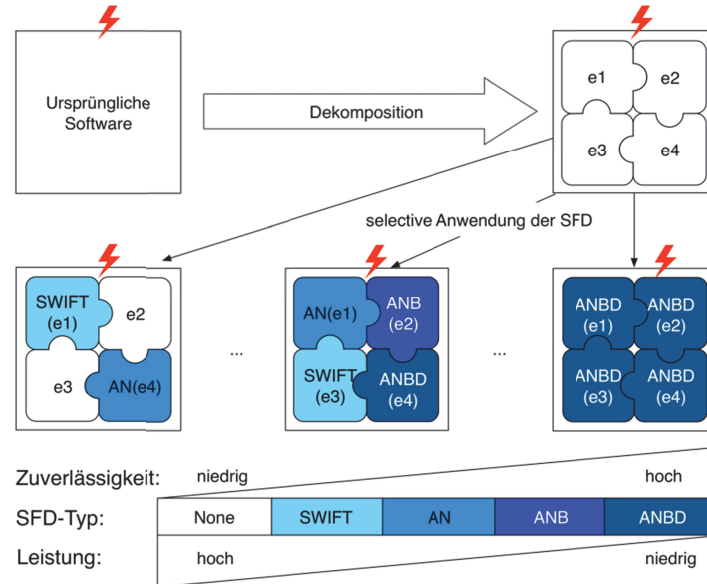


Abbildung 6.17 Systemdekomposition und selektive Anwendung der SFD.

Ein Teil der sicherheitskritischen Software des Eingebetteten Flugreglers wurde für diese Fallstudie durch iFR zur Verfügung gestellt. Diese Software wurde gemäß der Aktivitätsdiagramm im linken Teil von Abbildung 6.18 in die Programmiersprache C umgesetzt. Bei der Untersuchung wurde eine dreifache Redundanz der Funktionen berücksichtigt. Eine Schutzstrategie wird dadurch definiert, dass eine begrenzte Menge der Funktionen durch die Redundanz geschützt wird. Die verbleibenden Funktionen werden nicht geschützt und verursachen somit kein Overhead. Bei der Anwendung dieses Ansatzes auf fünf Funktionen (mit Ausnahme von „read-input“) entstehen 32 unterschiedliche Schutzstrategien, die in einer Simulationsumgebung ausgewertet werden können. Im rechten Teil der Abbildung 6.18 repräsentiert jeder Punkt die Zuverlässigkeit und die Rechenleistung einer ausgewerteten Schutzstrategie. Die grüne Kurve ist eine Pareto-Front aus sinnvollen Schutzstrategien. Eine optimale Schutzstrategie kann aus der Pareto-Front durch eine geeignete Festlegung einer der beiden Anforderungen gefunden werden.

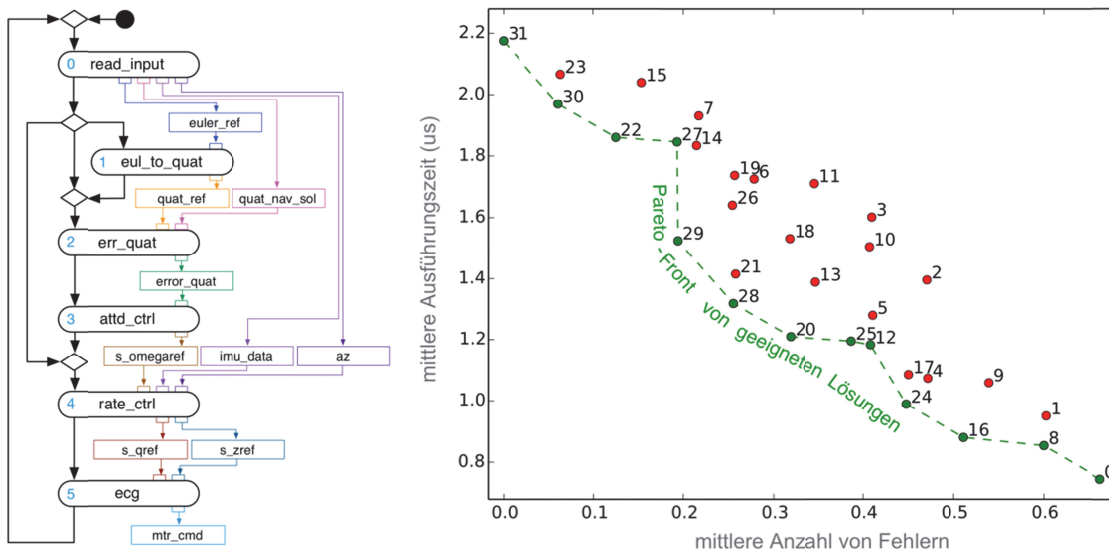


Abbildung 6.18 Links: UML-Aktivitätsdiagramm der Hautschleife der Low-Level-Flugregelungssoftware (schwarze Verbindungen zeigen Kontrollfluss zwischen den Aktivitäten/Funktionen, Farbverbindungen zeigen Datenfluss zwischen den Funktionen und Variablen). Rechts: Ergebnisse der Auswertung der Schutzstrategien.

6.6.2 Erkennung und Behandlung von kritischen Fehlern

Die Erkennung und Behandlung von kritischen Fehlern (auf Software- und Hardwareebene) bezweckt eine autonome Systemtoleranz gegenüber Defekten („safe“).

Bei der Fehlererkennung wird das System im Betrieb beobachtet, um mithilfe geeigneter Maßnahmen eine Konsistenz mit erwartetem Systemverhalten sicherzustellen. Gemäß den IEEE-Normen stellt die Fehlererkennung ein Mittel dar, um die Systemtoleranz gegenüber Defekten im Kontext der Systemsicherheit zu erreichen. Daher ist die Fehlererkennung ein erforderlicher Bestandteil eines sicherheitskritischen Systemdesigns.

Gleichzeitige Fehlererkennung und autonome Fehlerbehandlung wurde vor allem in AP2.5 untersucht. Dabei wurde ein spezielles an die Aufgabe angepasstes Framework für Sicherheitsüberwachung eingeführt, das auf simultaner Fehlererkennung und Fehlerbehandlung für verteilte Systeme basiert. Der Schwerpunkt des Ansatzes liegt auf Überwachung der verteilten Systemsoftware in Echtzeit. Eine spezielle Überwachungssoftware empfängt ständig Sicherheitszustände von kritischen Softwarekomponenten des Systems, bestimmt den Sicherheitszustand für das Gesamtsystem aus diesen Informationen und sendet den Sicherheitszustand des Systems zu einer speziellen Steuersoftwarekomponente, die das Fluggerät im Falle eines Fehlers in einen sicheren Schwebeflug versetzt oder eine Sicherheitslandung einleitet. In parallel unterstützt die Überwachungssoftware die Übertragung der Telemetriedaten mit relevanten Sicherheitsinformationen zu einer Bodenstation.

Im Projekt wurde das Konzept im Rahmen des Systemtestbenches mit der Flugplattform (SYS-FP-GNC) umgesetzt. Die Architektur des entwickelten Frameworks ist in Abbildung 6.19 vorgestellt.

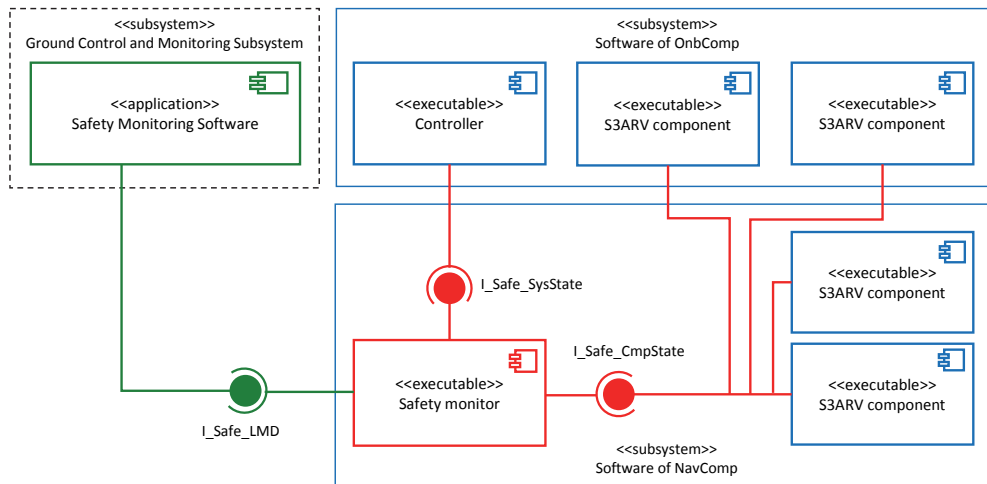


Abbildung 6.19 UML-Komponentendiagramm des Frameworks für Sicherheitsüberwachung.

Die Softwarekomponenten des verteilten Frameworks gehören zu zwei bordbasierten Rechereinheiten und zu einem bodenbasierten Rechner. Die Implementierung des Frameworks beinhaltet einen Satz von Softwarekomponenten für Fehlererkennung und Fehlerbehandlung, welcher aus folgenden Teilen besteht:

- *Sicherheitsklienten* sind spezielle Softwaremodule, die in den Systemkomponenten integriert sind und die Sicherheitszustände an den Sicherheitsmonitor senden. Die Sicherheitszustände werden dabei in den Systemkomponenten generiert.
- *Sicherheitsmonitor* ist eine eigenständige bordbasierte Softwarekomponente, welche die Fehlerwarnungen empfängt und einen Sicherheitszustand für das Gesamtsystem unter Beachtung der Echtzeit des Systems generiert. Dieser Sicherheitszustand wird dann an Fehlerhandler übermittelt. Gleichzeitig sendet das Sicherheitsmonitor Logging- und Monitoring-Daten an die (bodenbasierte) Software zur Sicherheitsüberwachung.
- *Fehlerhandler* ist ein Softwaremodul, das in die Flugsteuerungssoftware integriert ist und erforderliche Notaktionen für das System einleitet, wie z.B. Umschalten in den Modus des Schwebefluges und in den Modus der Sicherheitslandung.
- *Software zur Sicherheitsüberwachung* ist eine bodenbasierte Anwendung, die den Systemoperatoren die Sicherheitszustände aller überwachten Systemkomponenten, den Sicherheitszustand des Gesamtsystems sowie Warnungen über eine Benutzerschnittstelle darstellt.

Die Software des Frameworks wurde während der Entwicklung in einer speziell vorbereiteten Simulationsumgebung getestet und verifiziert (in AP2.5). In weiteren Schritten wurde das Framework in die Bordsoftware der Flugplattform integriert (in AP4.3) und anschließend im Rahmen der Flugerprobung erfolgreich demonstriert (in AP4.4).

6.7 Wissenschaftliche IfA-Highlights im Überblick

Ein kombiniertes Konzept der autonomen optischen Navigation mit minimalistischer Sensorik (Kamera, IMU): absolute Lokalisierung mit keypointbasiertem SLAM, unter Verwendung von IMU Daten, dichte Karten mithilfe des optischen Flusses.

Ein intelligenter Ansatz zur Kapselung fehlerkritischer Software: Markow-basierte Methoden zur Leistungs- und Zuverlässigkeitsanalyse, Optimierte Verwendung von Software für Detektion von Hardware-Fehlern.

6.8 Bearbeitungsstand nach Arbeitspaketen: Zusammenfassung

AP	Name	Stand
AP 1	Systementwurf	
1.1	Anforderungsdefinitionen	abgeschlossen
1.2	Systemkonzept	abgeschlossen
1.3	Testplan	abgeschlossen
1.4	Auswertung	abgeschlossen
AP 2	Algorithmenentwurf und Entwicklung	
2.1	Navigation	zum Teil abgeschlossen
2.2	Control	abgeschlossen
2.3	Guidance	abgeschlossen
2.4	SPLAM	Zielstellung geändert
2.5	RAMS	abgeschlossen (Software-Entwurf)
2.6	Analytische Redundanz	abgeschlossen (mit Modifikationen)
AP 3	Implementierung und Integration	
3.1	Bordrechner	abgeschlossen
3.2	Navigations-Elektronik	abgeschlossen (mit Modifikationen)
3.3	Bordrechner-Software	abgeschlossen
3.4	Ground Control + Monitoring	abgeschlossen
AP 4	Demonstration	
4.1	Testbench	abgeschlossen (mit Modifikationen)
4.2	Flugplattform	abgeschlossen
4.3	Test und Verifikation	abgeschlossen (für Navigation zum Teil abgeschlossen)
4.4	Flugdemonstration	abgeschlossen (für Navigation zum Teil abgeschlossen)

7 Wichtigste Positionen des zahlenmäßigen Nachweises

7.1 0812 „Beschäftigte E12-E15 / BAT IIa bis I“ und 0822 „Beschäftigungsentgelte“

Die multidisziplinäre Projektaufgabe wurde am IfA in einem forschungsorientierten Team bearbeitet. Wie geplant bestand das Team aus wissenschaftlichen Mitarbeitern. Unterstützend wurde studentische Hilfskraft (unter 0822) eingesetzt. Die gesamte Aufwandsplanung wurde weitgehend eingehalten.

7.2 0831 Gegenstände bis zu 410 / 400 EUR

Zu dieser Gruppe gehören acht zusätzliche LiPo-Akkus für die Flugplattformen und ein LiPo-Tresor zur sicheren Aufbewahrung von LiPo-Akkus.

7.3 0843 Sonstige allgemeine Verwaltungsaufgaben

Die Ausgabe dieser Gruppe bestehen hauptsächlich aus Kosten für die Reparatur direkter Folgen aus Unfall einer Flugplattform (das beschädigte Rahmensystem wurde durch Cadmic GmbH repariert) und Kosten für Verbrauchsmaterial zur Anfertigung der künstlichen Elemente der Testumgebung.

7.4 0846 Dienstreisen

Die Dienstreisen vom IfA wurden im Vorhaben für folgende Projektereignisse durchgeführt: Kick-Off Meeting, Fortschrittsgespräche (Nr. 1, 2 und 4), Projektreffen (vor allem Definition technischer Schnittstellen, Systemintegration und -tests), Flugtests sowie Endpräsentation. Außerdem wurde die 2. Nationale Konferenz zur Raumfahrt-Robotik in Berlin besucht.

7.5 0850 Gegenstände und andere Investitionen von mehr als 410 / 400 EUR im Einzelfall

Die Positionen gemäß der Anlage zum Antrag „Gegenstände und andere Investitionen (Geräteliste)“ stehen im Kapitel in Anführungszeichen.

Für die Testbenches wurden insgesamt drei "Kameras" mit drei geeigneten "Objektiven" (inkl. einiger kleiner Befestigungselemente) gekauft (für die Navigationskamera).

Für die "Bodenstation (Laptop)" wurde ein Notebook beschafft, der außerdem als Universeller Projekt-Rechner für alle Softwarekomponenten der bodenbasierten Subsysteme diene.

Für die Flugplattformen wurden zwei Rahmensysteme Goliath der Firma Cadmic GmbH (ehemals Cadmicopter GmbH) mit Zubehör (jeweils ein mechanischer Rahmen mit acht eingebauten Motoren und Propellern, zwei elektronischen Platinen der Motorregler, zwei Akkus, acht Ersatzpropellern und Befestigungsschrauben - "Mehrröhrensystem").

Sonstige bordbasierte und bodenbasierte Hardwareelemente und -komponenten der entwickelten Testbenches (eingeordnet unter der Position "4 Bordrechner inkl. 2 Echtzeit-Testbeds") wurden ebenfalls wie geplant angeschafft.

8 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die beantragte Zuwendung diene dem Zweck, den Förderschwerpunkt Raumfahrtrobotik im Bereich der wichtigen Autonomiefunktionen wie Bewegungsplanung, Navigation, Umgebungserkennung, Regelung mit Berücksichtigung von operationellen Grenzen / Hardwaregrenzen sowie Fehlertoleranz zu unterstützen.

Das Vorhaben verfolgte das Ziel, diese erforderlichen Autonomiefunktionen im Rahmen einer ganzheitlichen Systembetrachtung von realistischen Raumfahrtanwendungen planetarer Landung zu entwickeln und an einem Funktionsmodell zu demonstrieren. Die Hauptaspekte waren dabei: limitierte Masse, Volumen und Leistungsaufnahme der bordbasierten Recheneinheiten und Sensoren („small“); sichere Algorithmen, die eine Lösung der Navigation und Bewegungsplanung in endlicher Zeit liefern können und fehlertolerantes Verhalten aufweisen („safe“) sowie das Potential besitzen, in raumfahrttaugliche Hardware umgesetzt zu werden („space“).

Die Zuwendung war notwendig, da für diese Aktivitäten keine Eigenmittel der Verbundpartner zur Verfügung stehen.

Die Ergebnisse dieses Vorhabens sind nicht direkt am Markt nutzbar. In erster Linie wurden Ansätze erörtert und Verfahren erprobt, um neue Lösungsmöglichkeiten für die Problemstellungen zu untersuchen. Eine marktnahe Verwendung der Ergebnisse wird erst mit weiteren Entwicklungsschritten seitens der Verbundpartner möglich sein.

Mittels der Zuwendung wurden Technologien erschlossen, die für sichere autonome Landung kleiner Flugvehikel auf Planeten notwendig sind, und auch im Bereich atmosphärischen Flugs z.B. im Katastrophenschutz Anwendung finden können. Die damit verbundenen Möglichkeiten haben ein hohes Entwicklungspotenzial, um zukünftig entsprechend eines Knowledge-Based Projektmanagements die Risiken und Kosten von Projekten zu reduzieren.

9 Voraussichtliches Nutzen und Verwertbarkeit des Ergebnisses im Sinne des fortgeschriebenen Verwertungsplans

Schutzrechte

Keine Schutzrechte wurden im Rahmen des Vorhabens in Anspruch genommen. Eine Anmeldung von Schutzrechten wird nach Beendigung des Vorhabens nicht geplant.

Wirtschaftliche Erfolgsaussichten

Eine direkte wirtschaftliche Anwendung der Ergebnisse dieses Forschungsvorhabens war nicht geplant. An beiden beteiligten Forschungseinrichtungen (iFR und IfA) wurde im Rahmen dieses Vorhabens jedoch innovatives Know-how im Bereich kleiner unbemannter Raumfahrzeuge (und als Spin-Off: Flugzeuge/Drehflügler) generiert, was ein in Zukunft immer wichtiger werdendes Gebiet auch auf nationaler Ebene darstellt. Die entwickelten Systemkonzepte (Hardware, Software, Funktionen) sowie algorithmische Lösungen sind speziell für in Planung befindliche bzw. zukünftige Raumfahrtmissionen (planetare Landung, Exploration) geeignet. Die Ausbildung der beteiligten Doktoranden und Studenten auf diesem Gebiet kann somit als volkswirtschaftlich relevant eingeschätzt werden. Am iFR und am IfA waren bereits vier Doktoranden an Arbeiten im Themengebiet des Vorhabens beteiligt (zwei am iFR, zwei am IfA) und es wurden zwölf studentische Arbeiten unterstützt (sieben am iFR und fünf am IfA).

Wissenschaftliche und/oder technische Erfolgsaussichten

Es wird geplant, die Ergebnisse des Vorhabens auf internationalen Konferenzen (wie Aerospace: AIAA Guidance, Navigation, and Control Conference, CEAS Guidance, Navigation, and Control Conference, IFAC Automatic Control in Aerospace Symposium, ESA GNC Conference; Robotik: IROS, ICRA) und in entsprechenden Journalen (wie AIAA Journal of Guidance, Control and Dynamics, Control Engineering Practice, CEAS Space Journal, IEEE Control Systems Technology, IEEE Aerospace and Electronic Systems) überwiegend im Zeitraum 2015/2016 einer kritischen Fachöffentlichkeit zu präsentieren. Speziell sind die geplanten Veröffentlichungen zu nennen, die in Kapitel 11 vorgestellt sind. Das internationale Interesse an diesem Thema zeigt sich bei der Betrachtung der Programme der oben genannten internationalen Konferenzen. Dies dient dem wissenschaftlichen Ansehen des Standorts Deutschland genauso wie den beiden beteiligten Forschungseinrichtungen.

Wissenschaftliche und wirtschaftliche Anschlussfähigkeit

Mit den Projektergebnissen ergeben sich für beide beteiligten Forschungseinrichtungen beste Möglichkeiten für weiterführende Projektaktivitäten vor allem im Planetary-Landing-Kontext. Die angestrebten Folgeaktivitäten am IfA im Überblick:

- Weiterentwicklung des vorgeschlagenen Navigationsansatzes,
- weiterführende RAMS-Untersuchungen,
- Weiterentwicklung und Optimierung bezüglich Raumfahrttauglichkeit, bezüglich spezifischer Missionsanforderungen (z. B. Asteroid).

Die entwickelten Lösungen stellen weiterhin Schlüsselemente für Multi-Vehikel-Systeme dar, z. B. für verteilte und situationsangepasste Exploration. Für diese Szenarien sind zusätzlich zu den Autonomiefunktionen noch Koordinationsfunktionen (z. B. Adaption der Konfiguration, wechselnde Aufgaben) und deren technologische Umsetzung (Rekonfiguration, Kommunikation) durch beide beteiligten Forschungseinrichtungen in Kooperation zu entwickeln.

Es wird angestrebt, die Möglichkeiten zu den potenziellen Folgeaktivitäten im Anschluss an das Vorhaben in nachfolgenden Jahren zu ersuchen.

10 Während der Durchführung des Vorhabens dem Zuwendungsempfänger bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Morpheus Lander mit ALHAT

Während der Projektlaufphase gab es mehrere erfolgreiche Testflüge des sogenannten Morpheus Landers, der als Testträger für das ALHAT (Autonomous Landing and Hazard Avoidance Technology) genannte Lande- und Hindernisvermeidungssystem dient. Beide von der NASA und dem JPL seit mehreren Jahren in Entwicklung befindlichen Systeme sind für zukünftige Mond- bzw. Marsmissionen zur autonomen Landung auf dem jeweiligen Himmelskörper gedacht. Der technologische Lösungsansatz zur Kartierung des Ziellandegebiets ist jedoch ein anderer als er bei S3ARV gewählt wurde: Anders als beim vorliegenden Projekt wird auf Laserentfernungsmessung (LiDAR) mittels dreier Sensoren zur Kartierung der Landezone zurückgegriffen, was aufgrund der zulässigen Zuladung des Morpheus Landers von bis zu 499 kg machbar ist.

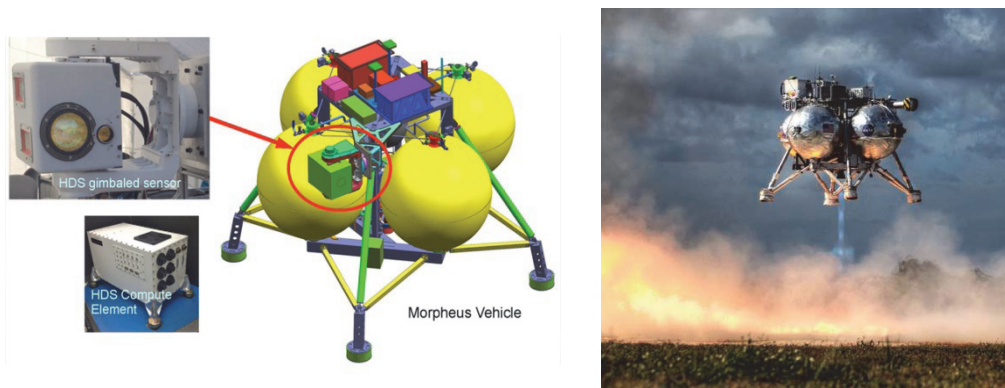


Abbildung 10.1 Morpheus Lander und ALHAT (Autonomous Landing and Hazard Avoidance Technology) mit HDS (Hazard Detection System). Fotos/Bilder mit freundlicher Genehmigung von NASA/JPL-Caltech (links) bzw. NASA/Project Morpheus (rechts).

Aufgrund der starken Zuladungsbeschränkung von kleinen Fluggeräten („small“) wurde bei S3ARV jedoch auf die Kartierung mittels überwiegend optischer Verfahren zurückgegriffen, da die Kameras in Masse und Formfaktor für die Größe der kompakten Flugplattformen geeigneter sind.

11 Erfolgte oder geplante Veröffentlichungen des Ergebnisses

Folgende Beiträge und Papers wurden veröffentlicht:

Rashidov, A. ; Morozov, A. ; Janschek, K. ; Yusupova, N.: Concurrent Error Detection for Distributed Systems: A Case Study. In: Proceedings of the Conference “Information Technologies for Intelligent Decision Making Support”, Ufa, Russia (18-21 May 2014)

Morozov, A. ; Janschek, K.: Optimized Software-Implemented Fault Tolerance. In: 1st PERASPERA Workshop, Poster session, ESTEC, Noordwijk, The Netherlands (11-12 February 2015)

Geplante Publikationen sind:

Morozov, A. ; Janschek, K.: Reliability and Performance Optimization for Software-implemented Hardware Fault Detectors. (eingereicht bei Springer Software Engineering for Self-Adaptive Systems, February 2015).

12 Quellen

- Bay 2008** Bay, H. ; Ess, A. ; Tuytelaars, T. ; Gool, L. V.: SURF: Speeded up robust features. In: Computer Vision and Image Understanding 110 (2008), Nr. 3, S. 346-359
- Chen 2007** Chen, W. C. ; Xiong, Y. ; Gao, J. ; Gelfand, N. ; Grzeszczuk, R.: Efficient Extraction of Robust Image Features on Mobile Devices. In: Mixed and Augmented Reality, ISMAR 2007, 6th IEEE and ACM International Symposium on (13-16 Nov. 2007), S.287, 288
- Civera 2010** Civera, J. ; Grasa, O. G. ; Davison, A. J. ; Montiel, J. M. M.: 1-Point RANSAC for EKF Filtering: Application to Real-Time Structure from Motion and Visual Odometry. In: Journal of Field Robotics (2010), Vol. 27, S. 609-631
- Janschek 2005** Janschek, K. ; Tchernykh, V. ; Beck, M.: An Optical Flow Approach for Precise Visual Navigation of a Planetary Lander. In: 6th International ESA Conference on Guidance, Navigation and Control Systems, Loutraki, Greece (17-20 October 2005)
- Joos 2011a** Joos, A. ; Fichter, W.: Parallel Implementation of Constrained Nonlinear Model Predictive Controller for an FPGA-based Onboard Flight Computer. In: F. T. Holzapfel, Advances in Aerospace Guidance, Navigation and Control, Berlin : Springer Verlag, S. 273-286
- Joos 2011b** Joos, A. ; Weimer, F. ; Fichter, W.: Path Planning with FPGAs for UAV Applications. In: Proc. of the 8th International ESA Conference on Guidance, Navigation & Control Systems, Karlovy Vary, Czech Republic (5 - 10 June 2011)
- Morozov 2014** Morozov, A. ; Janschek, K.: Probabilistic Error Propagation Model for Mechatronic Systems. In: Mechatronics (Elsevier) (2014), Vol. 24, Issue 8, pp. 1189 - 1202
- OCamCalib 2012** <https://sites.google.com/site/scarabotix/ocamcalib-toolbox> (abgerufen am 11. Dezember 2012)
- Scaramuzza 2006** Scaramuzza, D. ; Martinelli, A. ; Siegwart, R.: A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In: Proceedings of IEEE International Conference of Vision Systems (ICVS'06), New York (January 5-7, 2006)
- Schlicht 2011** Schlicht, M.: Entwicklung einer Software zur schnellen parallelen Berechnung des optischen Flusses mit Intel-Prozessoren. Technische Universität Dresden. Studienarbeit, 2011
- Schnitzer 2011** Schnitzer, F. ; Sonnenburg, A. ; Janschek, K. ; Willich, G.: SLAM-based 3D Shape Estimation for Rendezvous with Uncooperative and Unknown Target Spacecraft. In: 8th Int'l ESA Conf. on Guidance and Navigation Control Systems, Carlsbad, Czech Republic (5 - 10 June 2011)
- Weimer 2010** Weimer, F. ; Trittler, M. ; Joos, A. ; Gros, M. ; Posch, A. ; Fichter, W.: FPGA-Based Onboard Computer System for Mini Aerial Vehicles. In: International Micro Air Vehicle Conference and Flight Competition, Braunschweig, Germany (2010)

13 Anhang

13.1 Modifikation von wichtigen Systembegriffen aus dem Antrag

Einige Begriffe und Bezeichnungen einiger Systemkomponenten wurden während der Entwicklung zum Zwecke einer optimierten Systemarchitektur modifiziert. In diesem Kapitel werden solche Modifikationen dargestellt. Die betroffenen Begriffe (Antragsnamen) stehen im Kapitel in Ausführungszeichen, die relevanten aktuellen Begriffe (Projektnamen) sind kursiv.

Navigationselektronik (Arbeitspaket AP3.2)

An der Stelle eines „GPU-Rechners“ bzw. „GPU-beschleunigten Rechners“ (der Optischer-Fluss-Prozessor zur Berechnung des optischen Flusses) wurde eine *Externe Bildverarbeitungseinheit* eingesetzt. Die Bildverarbeitungseinheit stellt ein übergeordnetes Systemelement dar und ist aktuell auf Basis eines mobilen und für Feldtests geeigneten Notebooks implementiert.

Dem „Bildverarbeitungsrechner“ gehörte ursprünglich nur die Bildauswertung. Der Rest des Algorithmus der optischen Navigation sollte auf dem „Bordrechner“ implementiert werden. Um den schwächeren „Bordrechner“ zu entlasten, wurde der Rest des Navigationsalgorithmus dem „Bildverarbeitungsrechner“ zugeteilt. Der „Bildverarbeitungsrechner“ wurde somit zum *Navigationscomputer*.

Die *Navigationskamera* enthält die „Kamera“ und das „Objektiv“ (inkl. einiger kleiner Befestigungselemente).

Das „RF-Link“ wurde mit einem Wi-Fi-Netzwerk realisiert und wird daher häufig als *Wi-Fi-Verbindung* oder einfach *Wi-Fi* bezeichnet.

Bordrechner (Arbeitspaket AP3.1)

Das Gerät „Bordrechner“ entspricht im Ganzen dem aktuellen *Bordrechner* im System (Funktionen Guidance und Control) und besteht hardwaremäßig aus drei elektronischen Platinen, auf welchen *Zentraler Flugcomputer* und *Eingebetteter Flugregler* realisiert sind.

In der Anlage zum Antrag „Gegenstände und andere Investitionen (Geräteliste)“ steht das Gerät *Bordrechner* (genau gesagt nur die elektronischen Komponenten) unter Position Nr. 5 „4 Bordrechner inkl. 2 Echtzeit-Testbeds“.

Es ist jedoch zu beachten, dass unter den „Bordrechnern“ der Position Nr. 5 die Kosten aller elektronischen Komponenten der Bordelektronik und Bordsensorik der Flugplattform zusammenfasst eingeplant wurden (mit Ausnahme der *Motorregler* und der *Navigationskamera*). Neben dem *Bordrechner* gehören dazu vor allem der „Bildverarbeitungsrechner“ (*Navigationscomputer*), die „Inertialsensorik“ (IMU), der GPS-Empfänger (*GPS-Modul*, *GPS-Antenne*), der *Fernbedienungsempfänger* (siehe z.B. Abbildung 6.2).

Bodenstation (Arbeitspaket AP3.4)

Die geplanten Funktionen der „Bodenstation“ wurden auf zwei bodenbasierten (logischen) Rechereinheiten implementiert. Die erste Einheit *Bodenstation für Kontrolle und Überwachung* (verantwortlich iFR) läuft als eine Anwendung unter dem Betriebssystem Windows 7. Die zweite Einheit *Bodenstation für Überwachung* (verantwortlich IfA) wurde konform zum *Navigationscomputer* für das Betriebssystem Linux entwickelt. Der zweiten Einheit gehört jetzt die „Telemetrieschnittstelle zu optischen Navigationssystem“. Beide Einheiten wurden physikalisch auf einem mobilen und für Feldtests geeigneten Notebook implementiert.

Dasselbe Notebook diente als *Universeller Projekt-Rechner* auch für die *Externe Bildverarbeitungseinheit* und stellte somit eine (erweiterte) Bodenstation für die Implementierung aller bodenbasierten Steuerungs- und Auswertungsfunktionen dar. In der Anlage zum Antrag „Gegenstände und andere Investitionen (Geräteliste)“ wurde das Notebook unter Position Nr. 3 „Bodenstation (Laptop)“ eingeplant.

Bordrechner-Software

Zweck der „Bordrechner-Software“ war vor allem einen Rahmen für die algorithmischen Komponenten (aus AP2) zu schaffen. Für den *Bordrechner* wurde diese Software auf Grundlage der Vorarbeiten für Flächenflugzeug vom iFR weiterentwickelt. Für die anderen neuen stark zusammenhängenden Komponenten *Navigationscomputer*, *Bodenstation für Überwachung* und

Externe Bildverarbeitungseinheit wurde eine plattformübergreifende Rahmensoftware entwickelt (IfA), die als *Distributed Network based Framework (DNF, DNF-Framework)* bezeichnet wurde.

Flugplattform (Arbeitspaket AP4.2)

Die „Fernsteuerung“ besteht aus der bodenbasierten *Fernbedienung* (inkl. *Programmierungsbox*) und dem bordbasierten *Fernbedienungsempfänger*.

Die Flugplattform basiert auf einem Rahmensystem Goliath der Firma Cadmic GmbH (früher Cadmicopter GmbH). Das Rahmensystem Goliath besteht hauptsächlich aus einem mechanischen tragenden Rahmen (inkl. Rumpf, Ausleger, Landebeine usw.) und beinhaltet noch acht Motoren mit Propellern, zwei elektronische Platinen der Motorregler und zwei Akkus.

In der Anlage zum Antrag „Gegenstände und andere Investitionen (Geräteliste)“ wurde das Rahmensystem Goliath mit wenigem Zubehör (acht Ersatzpropeller, Befestigungsschrauben) unter die Position Nr. 4 „Mehrrohrsystem“ eingeordnet.

13.2 Zuordnung spezieller Bezeichnungen in UML-Modellen

Bezeichnungen in Englisch

Central Flight Computer
 Embedded Flight Controller
 External Image Processing Unit
 Flight Vehicle
 Ground Control and Monitoring Station
 Ground Control and Monitoring Subsystem
 Ground Monitoring Station
 Ground Processing Subsystem
 On-board Computer
 Ranger
 RC Receiver
 Remote Controller (RC)
 Safety Monitoring Software
 Safety Monitor
 Software of OnbComp

Systembegriffe in Deutsch

Zentraler Flugcomputer
 Eingebetteter Flugregler
 Externe Bildverarbeitungseinheit
 Flugplattform, Fluggäret
 Bodenstation für Kontrolle und Überwachung
 Subsystem für bodenbasierte Kontrolle und Überwachung
 Bodenstation für Überwachung
 Subsystem für Bodenbasierte Datenverarbeitung
 Bordrechner
 Entfernungsmesser
 Fernbedienungsempfänger
 Fernbedienung
 Software zur Sicherheitsüberwachung
 Sicherheitsmonitor
 Software des Bordrechners

Berichtsblatt

1. ISBN oder ISSN geplant	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel S3ARV – Demonstration robuster autonomer GNC-Funktionen für unbemannte Luft- und Raumfahrzeuge mit reduzierter Baugröße (Small & Safe Space Autonomous Robotic Vehicles) - Teilvorhaben: Navigation und RAMS-Analysen	
4. Autor(en) [Name(n), Vorname(n)] Prof. Dr. techn. Janschek, Klaus Dr.-Ing. Morozov, Andrey Dipl.-Phys. Losinger, Thomas Dr.-Ing. Dyblenko, Sergey	5. Abschlussdatum des Vorhabens 31.12.2014
	6. Veröffentlichungsdatum 30.06.2015
	7. Form der Publikation Schlussbericht
8. Durchführende Institution(en) (Name, Adresse) Technische Universität Dresden - Fakultät Elektrotechnik und Informationstechnik Institut für Automatisierungstechnik	9. Ber.-Nr. Durchführende Institution S3ARV-IFA-PM-FR-00 Ver. 1.0
	10. Förderkennzeichen 50RA1208
	11. Seitenzahl 40
12. Fördernde Institution (Name, Adresse) Raumfahrtmanagement des Deutschen Zentrums für Luft- und Raumfahrt e.V. Königswinterer Str. 522-524 53227 Bonn	13. Literaturangaben 12
	14. Tabellen 0
	15. Abbildungen 24
16. Zusätzliche Angaben k.A.	
17. Vorgelegt bei (Titel, Ort, Datum) Technische Informationsbibliothek (TIB) – Deutsche Forschungsberichte Welfengarten 1B, 30167 Hannover, August 2015	
18. Kurzfassung Das Verbundvorhaben verfolgte das Ziel, zukünftig erforderliche Autonomiefunktionen für Guidance, Navigation und Control für die präzise, sichere und autonome Landung eines Raumfahrzeugs (Referenzmission) im Rahmen einer ganzheitlichen Systembetrachtung unter wichtigen Randbedingungen eines Raumfahrteinsatzes wie „small“, „safe“, „space“ und „autonom“ zu entwickeln. Im Hinblick auf die Randbedingungen „small“ und „autonom“ wurden dabei kompakte optische und inertielle Sensoren sowie kompakte Rechentechnik vorgesehen. Neben der Referenzmission „Planetary Landing“ wurde auch ein terrestrisches Szenario im Bereich des Katastrophenschutzes betrachtet. Im Teilvorhaben „Navigation und RAMS-Analysen“ konzentrierten sich die Arbeiten des Institutes für Automatisierungstechnik (IfA) auf folgende Aufgabenbereiche: - bildbasierte Navigation (inkl. Mapping) unter Beachtung von Keypoints, optischem Fluss und inertialen Messdaten, - Analysen zu Reliability, Availability, Maintainability, Safety (RAMS) bei eingebetteten Softwarelösungen in einer geräteminimalen Hardwareumgebung (optimale Kapselung ausfallkritischer Softwareeinheiten; simultane Erkennung und Behandlung kritischer Fehler). Die entwickelten Algorithmen und Funktionen wurden mit simulierten und experimentellen Daten getestet. Die Experimente mit Multikopter-Flugplattform wurden in Kooperation mit dem Projektpartner Institut für Flugmechanik und Flugregelung (iFR), Universität Stuttgart, realisiert. Die Grundlage für eine effektive Zusammenarbeit der räumlich getrennten Teams des iFR und IfA stellte ein integrierter virtueller Engineering-Ansatz, der auf einem gemeinsamen internetbasierten Repository mit Configuration Management Funktionalitäten für Software und Dokumente und regelmäßigem Wissensaustausch über Videokonferenzen basierten. Die im Rahmen des Projekts entwickelten Technologien besitzen das Potential, eine sichere Landung zukünftiger Landefahrzeuge auf fremden Planeten zu ermöglichen.	
19. Schlagwörter planetare Landung, bildbasierte Navigation, SLAM, optischer Fluss, RAMS, Software-Kapselung, Fehlererkennung, Fehlerbehandlung	
20. Verlag k.A.	21. Preis k.A.

Nicht änderbare Endfassung mit der Kennung 277294-5

Document control sheet

1. ISBN or ISSN planned	2. type of document (e.g. report, publication) Veröffentlichung (Publikation)
3. title S3ARV – Demonstration of robust, autonomous GNC functions for unmanned air- and spacecraft with reduced size (Small & Safe Space Autonomous Robotic Vehicles) - Sub-project: Navigation and RAMS Analysis	
4. author(s) (family name, first name(s)) Prof. Dr. techn. Janschek, Klaus Dr.-Ing. Morozov, Andrey Dipl.-Phys. Losinger, Thomas Dr.-Ing. Dyblenko, Sergey	5. end of project 31.12.2014 6. publication date 30.06.2015 7. form of publication Sonstiges
8. performing organization(s) name, address Technische Universität Dresden Faculty of Electrical and Computer Engineering Institute of Automation	9. originators report no. S3ARV-IFA-PM-FR-00 Ver. 1.0 10. reference no. 50RA1208 11. no. of pages 40
12. sponsoring agency (name, address) Raumfahrtmanagement des Deutschen Zentrums für Luft- und Raumfahrt e.V. Königswinterer Str. 522-524 53227 Bonn	13. no. of references 12 14. no. of tables 0 15. no. of figures 24
16. supplementary notes n/a	
17. presented at (title, place, date) Technische Informationsbibliothek (TIB) – Deutsche Forschungsberichte Welfengarten 1B, 30167 Hannover, August 2015	
18. abstract The goal of the joint project was to develop necessary autonomous guidance, navigation and control functions to enable a precise, safe and autonomous landing of a future spacecraft on another planet (reference mission). These functions should be designed considering an integrated system approach with the following constraints typical for a space mission: "small", "safe", "space", "autonomous". Compact optical and inertial sensors as well as compact computing units were considered to take care of the constraints "small" and "autonomous" during the project. A terrestrial scenario in the scope of disaster control was considered in addition to the reference mission "planetary landing". The activity of the Institute of Automation (IfA) was focused in the sub-project „Navigation and RAMS-Analysis“ on the following working areas: - image based navigation (incl. mapping), considering keypoints, optical flow and inertial measurements, - analysis in the scope of Reliability, Availability, Maintainability and Safety (RAMS) in embedded software solutions in a minimal hardware environment (optimal encapsulation of failure critical software units; simultaneous detection and handling of critical errors). The developed algorithms and functions were tested with simulated and experimental data. The experiments with the multicopter flight platform were realised in cooperation with the project partner Institute of Flight Mechanics and Control (iFR), University of Stuttgart. Effective cooperation of spatially separated teams of iFR and IfA was based on an integrated virtual engineering approach using a common Internet-based repository with Configuration Management functionality for software and documents and regular exchange of knowledge via video conferences. The technologies developed in the scope of this project can potentially enable a safe planetary landing on other planets in future missions.	
19. keywords planetary landing, image based navigation, SLAM, optical flow, RAMS, software encapsulation, error detection and handling	
20. publisher n/a	21. price n/a

Nicht änderbare Endfassung mit der Kennung 277300-8