

AUFBAU EINER HARDWARE-IN-THE-LOOP – SIMULATION FÜR DAS UNBEMANNT FIEGENDE SYSTEM IFSYS

M. Rabe,
Technische Universität Berlin, Institut für Luft- und Raumfahrt,
Marchstraße 12, 10587 Berlin, Deutschland

Zusammenfassung

Im Studierendenprojekt IFSys wird am Fachgebiet Flugmechanik, Flugregelung und Aeroelastizität der Technischen Universität Berlin ein *Intelligentes Fliegendes System* entwickelt – IFSys genannt. Studierende entwickeln in diesem Projekt ein unbemanntes fliegendes System. Sie erlernen dabei interdisziplinäres Arbeiten und erlangen praktische Erfahrungen. Zur Erprobung des automatischen Flugsteuerungssystems am Boden wurde eine Testanlage in Form eines Hardware-in-the-Loop (HIL) – Simulators aufgebaut. Mit ihm kann die entwickelte Hard- und Software des Flugsteuerungssystems unter realitätsnahen Bedingungen getestet werden, ohne dass die Gefahr von Beschädigungen besteht. Der vorliegende Bericht gibt einen Überblick über das Projekt IFSys und beschreibt das Funktionsprinzip, den Aufbau und das Bedienkonzept des HIL-Simulators.

SYMBOLE

H	Höhe
Θ	Längslagewinkel
Φ	Hängewinkel

ABKÜRZUNGEN UND INDIZES

ACARE	Advisory Council for Aeronautics Research in Europe
ADT	Air Data
AFCS	Automatic Flight Control System
AHRS	Attitude Heading Reference System
ALEXIS	Airborne Laboratory for Experiments on Inflight Systems
CAN	Controller Area Network
CAS	Calibrated Air Speed
DLC	Direct Lift Control
FCC	Flight Control Computer
FCL	Flight Control Laws
FMRA	Fachgebiet Flugmechanik, Flugregelung und Aeroelastizität (TU Berlin)
GCS	Ground Control Station
GPS	Global Positioning System
HIL	Hardware-in-the-Loop
IFSys	Intelligentes Fliegendes System
I/O	Input/Output
LSC	Local Servo Controller
MSL	Mean Sea Level
PWM	Pulse Width Modulation
RC	Radio Control
SUT	System Under Test
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
USH	Ultraschallhöhenmesser

1 EINLEITUNG

Die Entwicklung und Erprobung neuer Flugzeuge und ihrer Systeme ist ein langwieriger und kostenintensiver Prozess. Neben finanziellen Risiken birgt er auch Gefahren für die Sicherheit der Versuchspiloten. Hohe finanzielle Kosten

entstehen vor allem bei Programmverzögerungen, wie sie bei großen Verkehrsflugzeugprojekten in der nahen Vergangenheit auftraten.

Eine Halbierung der Zeit bis zur Markteinführung wurde vom *Advisory Council for Aeronautics Research in Europe* (ACARE) als eines der Hauptziele für die europäische Luftfahrtindustrie bis zum Jahr 2020 ausgerufen. Gleichzeitig sollen aktuelle Sicherheitsstandards gehalten oder verbessert werden. Realisiert werden soll dies unter anderem durch den Einsatz standardisierter Verfahren und computergestützter Methoden bei Produktentwicklung, -erprobung und -fertigung [1]. Ein Verfahren, den Entwicklungsprozess kürzer, kostengünstiger und sicherer zu machen, ist die Nutzung der Hardware-in-the-Loop – Simulation.

Die HIL-Simulation ist ein Testverfahren bei der Entwicklung eingebetteter Systeme. Das sind Systeme, bei denen ein Computer einen Teil eines komplexen technischen Systems bildet [2]. Anwendung findet die HIL-Simulation in der Entwicklung von Flugregelungssystemen in der Luftfahrt aber auch von Fahrzeugelektronik im Automobilbau.

Die HIL-Simulation ermöglicht die realitätsnahe Überprüfung der Funktionalität von Steuergeräten außerhalb ihrer physikalischen Systemumgebung. Das zu testende Gerät oder System wird mit einem Simulationsrechner verbunden, auf dem die entsprechende Systemumgebung modelliert wird. Dieses Verfahren bietet verschiedene Vorteile. Zum einen sind HIL-Tests hoch automatisierbar und liefern reproduzierbare Ergebnisse. Zum anderen verkürzen sie die Entwicklungszeit der Gesamtsysteme. In modernen Entwicklungsprozessen existieren lange vor der Fertigstellung kompletter Prototypen genaue Computermodelle. Mit Hilfe der HIL-Simulation ist es möglich, das Flugverhalten eines Flugzeuges im Zusammenspiel mit dem realen Steuerungssystem in frühen Entwicklungsphasen zu überprüfen. Darüber hinaus bieten HIL-Simulationen die Möglichkeit, Regelungssoftware ohne Gefahr für beteiligte Personen und Maschinen zu testen.

Ziel des Projektes war es einen HIL-Simulator für ein unbemanntes Luftfahrzeug (UAV – *Unmanned Aerial Vehicle*) und für das zu seinem Betrieb erforderliche System (UAS – *Unmanned Aerial System*) aufzubauen. Der HIL-Simulator soll ein gefahrenfreies Testen von Hard- und Software-Modifikationen des Automatischen Flugsteuerungssystems (AFCS – *Automatic Flight Control System*) des UAVs sowie des Datenlink zur Kommunikation zwischen UAV und Bodenkontrollstation ermöglichen.

2 INTELLIGENTES FLIEGENDES SYSTEM IFSYS

IFSys ist der Name eines Studierendenprojektes an der Technischen Universität Berlin und steht für *Intelligentes fliegendes System*. Ziel des Projektes sind Entwicklung, Bau und Betrieb eines unbemannten fliegenden Systems (UAS) für zivile Einsatzzwecke.

Unbemannte Flugzeuge (UAVs) werden für Missionen eingesetzt, die ohne menschliche Piloten durchgeführt werden müssen, weil sie entweder sehr lange dauern, in verseuchter Luft stattfinden oder gefährlich sind (*dull, dirty or dangerous*). Kostengünstiger als bemannte Flugzeuge sind UAVs dann, wenn eine Mission eine leichte Nutzlast hat, die per Funk fernbedient werden kann. All dies kann bei zivilen und bei militärischen Anwendungen der Fall sein. Zu den Anwendungsfällen zählen beispielsweise Messflüge in sehr großer oder sehr geringer Höhe, Flüge zur Erdbeobachtung oder Verkehrsüberwachung sowie Feuerlöscheinsätze und Suchmissionen.

Ein UAS besteht aus einer Bodenkontrollstation, von der aus das Fluggerät ferngesteuert werden kann, und einem unbemannten Luftfahrzeug (UAV). Im Projekt IFSys heißt das Flugzeug ALEXIS (*Airborne Laboratory for Experiments on Inflight Systems*).

Um eine durch die Bodenkontrollstation vorgegebene Mission autonom abfliegen zu können, verfügt ALEXIS über ein elektronisches, automatisches Flugsteuerungssystem [3]. Dieses besteht aus einem Flugsteuerungscomputer (FCC – *Flight Control Computer*), Aktuatorik zur Ansteuerung der Stellgrößen sowie Sensorik zur Bestimmung von Position, Lage und Umströmungsbedingungen. Die Software des FCC wurde von den Studierenden selbst entwickelt [4]. Um Änderungen an den Regelungsfunktionen ohne Gefahr für den Versuchsträger testen zu können, wurde ein sogenannter Iron Bird aufgebaut. Der Iron Bird ist einen Prüfstand, auf dem ein Großteil des Flugsteuerungssystems (Aktuatorik, FCC, Signalübertragung) mit Originalkomponenten aufgebaut ist.

Durch die Mitarbeit im Projekt IFSys erhalten Studierende die Chance, ihre theoretischen Kenntnisse in den Gebieten Flugmechanik, Flugregelung, Aeroelastik, Aerodynamik, Leichtbau, Elektrotechnik, Informatik, Systementwicklung und vielen anderen Bereichen durch praktische Erfahrungen zu ergänzen und dabei Erfahrungen mit Projekt- und Teamarbeit zu machen.

2.1 Entwicklungsprozess

Die Entwicklung des UAS im Projekt IFSys erfolgt nach dem V-Modell, siehe BILD 1. Sie gliedert sich in drei Phasen: den Entwurf, die Implementierung und die Integration.

Entlang des linken Schenkels erfolgt der Entwurf. Begonnen wird mit der Festlegung der Spezifikationen für das Gesamtsystem Flugzeug (oben links). Bis hinunter zu den einzelnen Funktionen der Software sowie den Hardwareelementen des Flugsteuerungscomputers werden die Anforderungen immer weiter verfeinert (*top-down*-Prozess). Hierbei ist darauf zu achten, dass jede Spezifikation einer oberen Abstraktionsebene durch ein oder mehrere detailliertere Spezifikationen der darunter liegenden Ebene wiedergegeben wird. Umgekehrt muss sicher gestellt werden, dass Detailspezifikationen immer einer Spezifikation der darüber liegenden Ebene zugeordnet werden können („*traceability and compliance*“ - Prozess vgl. [5]). Der Fuß des V-Modells beschreibt die Implementierung der Systembestandteile. Hier wird die Software hergestellt und Hardwarekomponenten gefertigt. Der rechte Schenkel stellt den Integrations- und Verifikationsprozess dar. Dieser verläuft umgekehrt vom Entwurfsprozess. Die Integration geschieht vom Kleinen, den Einzelkomponenten, hinauf zum Großen, dem Gesamtsystem (*bottom-up*-Prozess) [6].

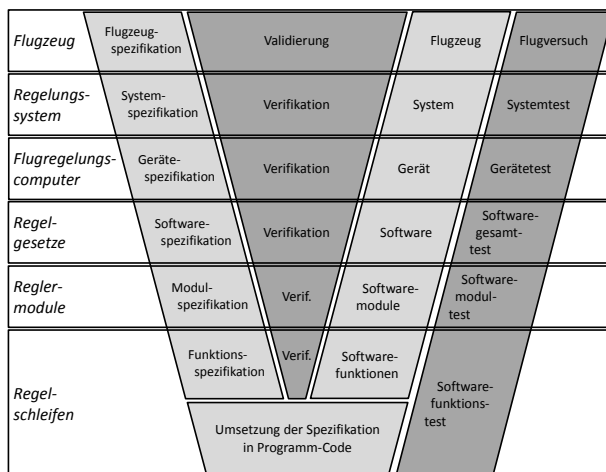


BILD 1. V-Modell für die Entwicklung eines Flugregelungssystems

Parallel zur Implementierung und Integration findet die Verifikation und Validierung des Gesamtsystems und seiner Komponenten statt. Die Erfüllung jeder Spezifikation ist durch einen Test mit definierten Soll-Ergebnissen zu überprüfen, zu bestätigen und zu dokumentieren [5].

Durch den *top-down*-Ablauf in der Entwurfsphase wird sichergestellt, dass alle globalen Anforderungen an das Gesamtprojekt in Detailspezifikationen übersetzt werden. Das ist Voraussetzung damit das Endprodukt allen Anforderungen gerecht werden kann. In der Integrationsphase gewährleistet der Verifikations- und Validierungsprozess, dass jede Spezifikation erfüllt ist. Der *bottom-up*-Verlauf der Integrationsphase bietet weitere Vorteile. So garantiert der Testprozess, dass immer mit fehlerfreien Teilprodukten weitergearbeitet wird. Dies verringert den Aufwand bei der Fehlersuche im Vergleich zu einem einzigen, abschließenden Test des Gesamtsystems. Außerdem wird die Wahrscheinlichkeit versteckter Fehler im Endprodukt reduziert, die bei einem einzelnen Abschlusstest unentdeckt bleiben könnten.

2.2 Flugversuchsträger ALEXIS

Der Flugversuchsträger ALEXIS ist ein etwa 11 kg schweres Hochdecker-Flugzeug mit einer Spannweite von 4 m. Besondere äußere Merkmale von ALEXIS sind sein hochgestreckter, ungefeilter Flügel und das umgedrehte V-Leitwerk (siehe BILD 2).



BILD 2. Flugversuchsträger ALEXIS

ALEXIS verfügt am Flügel über Querruder, Brems- und Wölbklappen. Letztere ermöglichen durch ihre hohe Stellgeschwindigkeit eine direkte Auftriebsbeeinflussung (DLC – *Direct Lift Control*). Ein kombiniertes Höhen- und Seitenruder (*Ruddervator*) ist am Leitwerk eingebaut und dient der Steuerung des Flugzeugs um dessen Hoch- und Querachse. Für Nickbewegungen werden beide Ruderflächen gleichsinnig, für Gierbewegungen gegensinnig ausgeschlagen.

2.3 Automatisches Flugsteuerungssystem

Für den automatischen Flug ist ALEXIS mit einem AFCS ausgestattet. Den Aufbau des Flugsteuerungssystems zeigt BILD 3 schematisch. Das AFCS besteht aus folgenden Komponenten:

- Flugsteuerungscomputer (FCC) mit einer Schnittstellenplatine,
- Sensorik:
 - Fluglage-/Kursanzeigesystem (AHRS – *Attitude Heading Reference System*),
 - GPS-Navigationssystem,
 - Luftdatensystem (ADT – *Air Data*),
 - Ultraschall-Höhenmesser (USH),
- Aktuatorik (Antriebsmotor, Servomotoren)
- einer Platine je Aktuator zur Aktuatoransteuerung und -überwachung (LSC – *Local Servo Controller*),
- einem RC-Empfänger zur Steuerung des UAVs über eine Funkfernbedienung und
- einem Funkmodem, über das der Flugsteuerungscomputer mit der Bodenkontrollstation kommunizieren kann.

Im Betrieb werden vom FCC die Werte der Sensoren eingelesen (Frequenz: 100 Hz). Sie liefern die Eingangsgrößen der Flugregelgesetze (FCL – *Flight Control Laws*). Mit Hilfe der FCL werden die zur Erfüllung der Flugaufgabe benötigten Stellflächenkommandos berechnet (Update-rate 50 Hz). Vom FCC werden diese Kommandos auf den CAN-Bus übermittelt. Die LSCs jedes Aktuators sind mit dem CAN-Bus verbunden. Sie wandeln die Stellkommandos vom CAN-Bus in ein pulsweitenmoduliertes Signal (PWM – *Puls Width Modulation*) um. Das PWM-Signal steuert dann die Aktuatoren. Zusätzlich liegen an den LSCs das Signal zur Umschaltung zwischen FCC und

Funkfernbedienung, sowie die Stellkommandos der Fernbedienung an [7]. Schaltet der Sicherheitspilot an der Fernbedienung von automatischem auf manuellen Flug um, so leiten die LSCs nicht mehr die Signale des FCC an die Aktuatoren weiter, sondern die des RC-Empfängers. Der Ausschlag einer Stellfläche wird von einem Winkelmesser bestimmt und per CAN-Bus von dem jeweiligen LSC an den FCC zurückgesendet. So kann der FCC beispielsweise feststellen, wenn ein Fehler an einem Aktuator oder der dazugehörigen Stellfläche vorliegt.

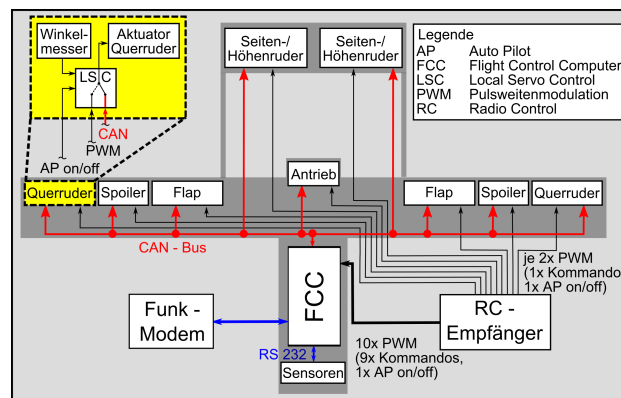


BILD 3. Schematische Systemübersicht des AFCS

2.4 Iron Bird openALEXIS

Als Iron Bird wird eine bodengebundene Testanlage bei der Entwicklung von Flugzeugen bezeichnet. Auf dieser Anlage werden die mechanischen, elektrischen und hydraulischen Steuerungssysteme eines Flugzeugmusters im Verbund getestet. Ein Iron Bird wird bereits in Entwicklungsphasen eingesetzt, in denen die Flugzeugzelle noch nicht vorhanden ist. Die Anordnung der einzelnen Bauteile erfolgt wie im späteren Flugzeug, so dass das Systemverhalten realitätsnah nachgebildet wird [8].

Im Projekt IFSys wurde ein Iron Bird aufgebaut. Er dient der Erprobung und Verifikation des Flugsteuerungssystems von ALEXIS. Auf ihm sind alle Aktuatoren inkl. der LSCs und der Flugsteuerungscomputer integriert wie BILD 4 zeigt. Dort ist ebenfalls der Simulationsrechner des HIL-Simulators dargestellt. Im folgenden Kapitel wird auf der Aufbau des HIL-Simulators detailliert beschrieben.

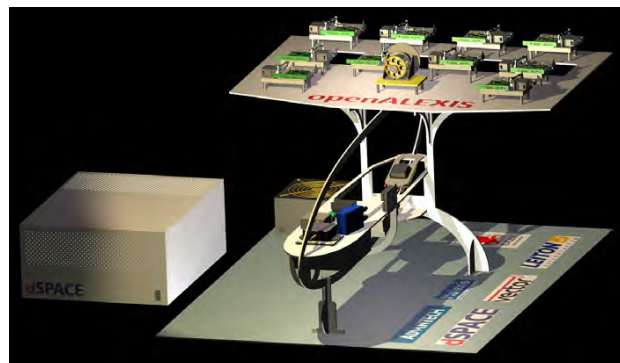


BILD 4. IronBird openALEXIS

3 HARDWARE-IN-THE-LOOP-SIMULATION

Ein Hardware-in-the-Loop – Simulator besteht aus zwei Hauptkomponenten: dem Simulationsrechner und dem zu testenden Steuergerät oder Teilsystem, im Folgenden als Testsystem (SUT – *System Under Test*) bezeichnet. Zur Kommunikation mit dem Testsystem ist der Simulationsrechner an dessen Ein- und Ausgänge angeschlossen. Es ist daher unabdingbar, dass der Rechner zum Testsystem kompatible Schnittstellen besitzt.

Auf dem Simulationsrechner wird das Verhalten der Systemumgebung nachgebildet. Dafür liest dieser die Testsystemausgänge aus, simuliert die Reaktionen der Umgebung auf diese Signale und stimuliert die Eingänge des SUT so, dass sich das SUT wie im realen Betrieb verhält. Die Simulation muss in Echtzeit erfolgen. Sowohl das Simulationsmodell als auch der -rechner müssen diese Anforderung erfüllen können.

Ein Bediensystem ermöglicht einerseits die Steuerung des Simulationsrechners durch den Testingenieur. Andererseits dient es der Darstellung und Speicherung der beobachteten und simulierten Parameter. Im Gegensatz zum Simulationsrechner unterliegt das Bediensystem nicht den Echtzeitanforderungen.

HIL-Simulatoren kommen in mehreren Stufen des in BILD 1 dargestellten Entwicklungsprozesses zum Einsatz. In den einzelnen Stufen kann sich der Anteil real vorhandener Systembestandteile unterscheiden. Die an die Ein- und Ausgänge des SUT direkt angeschlossenen Systeme werden als Lasten bezeichnet. Dazu zählen Sensoren, Aktuatoren oder andere Steuergeräte. Es wird zwischen zwei Fällen unterschieden. Bei Gerätetests wird die Systemumgebung des Steuergerätes, inkl. aller Lasten simuliert. Bei Systemtests hingegen, werden diese virtuellen Lasten teilweise oder vollständig durch Echtlasten (reale Bauteile) ersetzt.

Gegenüber Software-in-the-Loop – Tests bieten HIL-Tests den Vorteil, hardware-spezifische Eigenschaften eines Systems zu berücksichtigen. Dazu gehören beispielsweise Signalverarbeitungszeiten im Testsystem oder den angeschlossenen Geräten. Physikalische Effekte wie Reibung in einer angeschlossenen Aktuatorik können ebenfalls einen Einfluss auf das Verhalten des Testsystems haben. Bei der Verwendung von Echtlasten müssen diese Effekte nicht simuliert werden, sondern sind real vorhanden und beeinflussen das Testsystem wie im späteren Betrieb.

In den nachfolgenden Abschnitten wird der Hardware-in-the-Loop Simulator für die Entwicklung des Flugsteuerungs- und Flugregelungssystems des UAVs ALEXIS (vgl. Kapitel 2.3) beschrieben.

3.1 Konzept

An den HIL-Simulator im Projekt IFSys werden verschiedene Anforderungen gestellt. Primär wird der Simulator für Tests bei der Entwicklung der Hard- und Software für das AFCS des UAV ALEXIS eingesetzt. Er muss daher kompatibel zu den aktuellen Schnittstellen des Systems sein, aber auch an zukünftige Modifikationen anpassbar sein. Weiterhin soll der HIL-Simulator zur Entwicklung der Bo-

denstation in unterschiedlichen Ausbaustufen (nur Software, oder mit Hard- und Software) eingesetzt werden. Der HIL-Simulator dient außerdem dem Training des Sicherheitspiloten für die Phasen manueller Steuerung des UAV. Zusätzlich muss der HIL-Simulator einfach zu bedienen sein, da Studierende mit unterschiedlichem Wissenstand mit ihm arbeiten.

Aus diesen Einsatzzwecken und Randbedingungen ergeben sich folgende Anforderungen:

- Kompatibilität zu aktuellen Schnittstellen des SUT
- Anpassbarkeit an Modifikationen der Hardware des SUT
- Erzeugung benötigter Sensordaten
- einfache Bedienbarkeit
- hohe Automatisierbarkeit von Tests und deren Auswertung
- Visualisierung des Fluges
- Anbindung an die Bodenkontrollstation (GCS – *Ground Control Station*) in unterschiedlichen Ausbaustufen.

3.2 Funktionsprinzip

Beim AFCS des UAV ALEXIS bilden die Sensoren und die Aktuatoren die Schnittstellen mit der Umgebung. Die Sensoren messen die Bewegungsgrößen des UAV und stellen diese dem FCC zur Verfügung. Die Aktuatoren wandeln die elektrischen Steuersignale des FCC in mechanische Bewegungen der Stellflächen um und beeinflussen so die Flugbewegung des UAVs. Der HIL-Simulator greift daher an diesen Schnittstellen an. BILD 5 zeigt eine schematische Übersicht die Verbindung des AFCS mit dem Simulationsrechner.

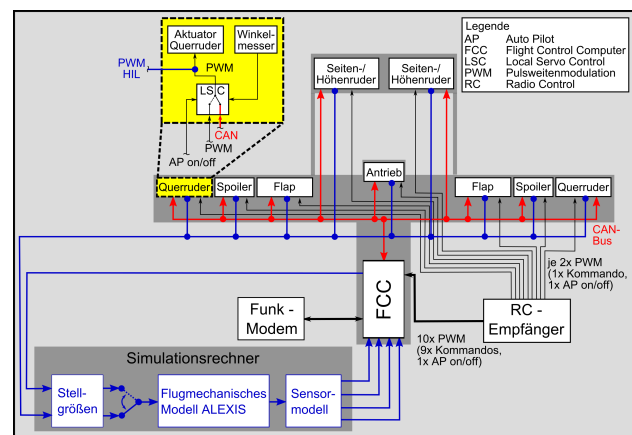


BILD 5. Schematische Systemübersicht des AFCS im Betrieb des HIL-Simulators

Die gesamte Sensorik ist während eines HIL-Tests außer Betrieb. Stattdessen stimuliert der Simulationsrechner die Sensorschnittstellen am FCC mit simulierten Sensorwerten. Anhand der Daten berechnet der FCC wie im Flugbetrieb die Steuerkommandos und sendet diese über den CAN-Bus an die LSCs. Dort werden die Kommandos als PWM-Signale an die Aktuatoren weitergeleitet. Diese PWM-Signale werden am Simulationsrechner eingelesen. Zusätzlich werden die gemessenen Stellflächenausschläge an einer seriellen Schnittstelle des FCCs eingelesen (siehe Kapitel 4.2).

Auf dem Simulationsrechner wird die Systemumgebung des AFCS simuliert. Anhand der Stellkommandos oder der Stellflächenausschläge wird die flugmechanische Bewegung des UAVs berechnet. Die Bewegungsgrößen werden in Signale umgerechnet, die denen der realen Sensoren gleichen und wieder an den FCC übertragen. So wird die Simulationsschleife geschlossen.

3.3 Aufbau

Der Simulationsrechner des IFSys HIL-Simulators ist ein erweiterbares Echtzeitsystem der Firma dSPACE, welches aus folgenden Komponenten besteht:

- eine DS1006 Prozessorkarte mit einer seriellen Schnittstelle,
- eine DS4004 Digital-I/O-Karte mit 96 Ein- und Ausgängen und
- eine DS4201-S Schnittstellen-Karte mit 4 seriellen Schnittstellen.

Zusätzlich zum Simulationsrechner wird ein Bediensystem (im Folgenden Host-PC genannt) benötigt. Dieser PC kommuniziert über eine optische Schnittstelle mit dem Simulationsrechner. Auf dem Host-PC läuft die Software zur Steuerung des Simulationsrechners und nach Bedarf weitere Programme zur Visualisierung eines Tests. BILD 6 zeigt schematisch den Aufbau des HIL-Simulators

Der Kabelbaum zur Verbindung von Simulationsrechner und Flugsteuerungssystem ist spezifisch für den IFSys-HIL, kann aber um zusätzliche Leitungen erweitert werden.

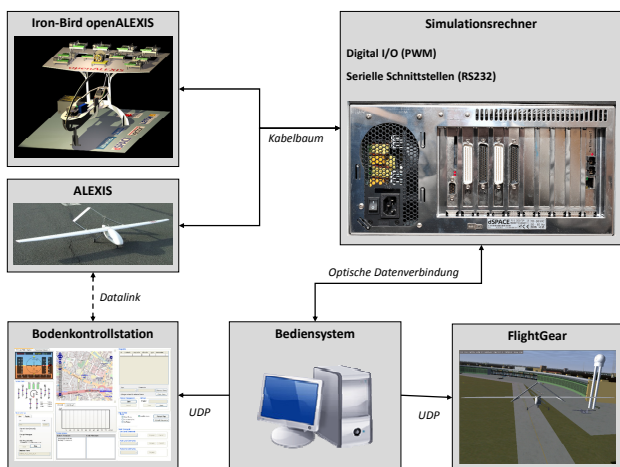


BILD 6. Schematischer Aufbau des HIL-Simulators

4 SIMULATIONSMODELL

Mit dem HIL-Simulator soll das Flugsteuerungssystem des UAV ALEXIS, bzw. vorrangig der Flugsteuerungscomputer und seine Software getestet werden. Das dafür benötigte Simulationsmodell muss die Eingänge des FCC so stimulieren, wie es im Flug der Fall wäre und gleichzeitig dessen Ausgangssignale verarbeiten.

Für die Entwicklung des Simulationsmodells von ALEXIS werden MATLAB und Simulink verwendet. Ein umfangreiches flugmechanisches Modell von ALEXIS wurde in [9] entwickelt und dient als Grundlage für das HIL-Simulationsmodell. Zur dSPACE Hard- und Software ge-

hört eine Simulink-Bibliothek, die eine einfache und zuverlässige Einbindung der Hardwareschnittstellen in die Simulation ermöglicht. Mit dem zu MATLAB gehörenden Real-Time Workshop Embedded Coder wird aus dem Simulink-Modell C-Code generiert. Aus diesem wird mit dem GNU C Compiler ein ausführbares Programm für den Simulationsrechner generiert.

Das Simulationsmodell von ALEXIS besteht aus drei Hauptmodulen (vgl. BILD 5): dem flugmechanischen Modell, das das Verhalten des UAV im Flug nachbildet, einem Modul zum Einlesen der Stellgrößen, das die benötigten Eingangssignale für das flugmechanische Modell zu Verfügung stellt sowie der Sensoremulation, die aus den Ausgangsgrößen der flugmechanischen Modellierung realitätsnahe Sensorwerte erzeugt und diese an den FCC sendet. Der Rechenzyklus (Gesamtschrittweite) der Simulation wird so gewählt, dass sie mit der Emulation der vier Sensorsysteme kompatibel ist. Alle drei Module werden in den nachfolgenden Abschnitten näher behandelt.

4.1 Flugmechanisches Modell

Die für das Simulationsmodell verwendete flugmechanische Simulation von ALEXIS beinhaltet folgende Effekte:

- nichtlineare Bewegungsgleichungen der Flugmechanik mit sechs Freiheitsgraden,
- Ein-Punkt-Aerodynamik,
- Triebwerkscharakteristik,
- Fahrwerkskräfte sowie
- statische und dynamische Atmosphäreneffekte.

Die Aerodynamik wurde mit Hilfe empirischer Methoden und eines 3D-Panelverfahrens modelliert [9]. Eine Validierung des Modells bzw. eine Identifizierung der Aerodynamik von ALEXIS durch Flugversuche ist geplant, aber bislang nicht umgesetzt.

4.2 Einlesen der Stellgrößen

Zur Steuerung des flugmechanischen Modells müssen der Simulation die Stellflächenausschläge bekannt sein. Dabei wird zwischen den Soll- und den Istwerten der Stellgrößen unterschieden.

Die Sollwerte sind die Kommandos, welche vom Flugsteuerungsrechner an die Aktuatoren gesendet werden. Jedes Kommando liegt in Form eines pulsweitenmodulierten (PWM-) Signals an dem LSC eines jeden Aktuators an. Die PWM-Signale werden an jedem LSC abgegriffen (siehe BILD 5) und über die Digital-I/O-Karte in den Simulationsrechner eingelesen.

Die Istwerte sind die tatsächlich eingestellten Stellflächenpositionen. Sie werden an jeder Fläche durch einen Winkelmesser bestimmt und per CAN-Bus an den FCC geschickt. Dieser stellt die Stellgrößen als Signal (Frequenz: 50 Hz) an einer seriellen Schnittstelle zur Verfügung. Von dort aus werden sie an den Simulationsrechner übertragen (siehe ebenfalls BILD 5).

Vom Simulationsrechner werden sowohl Soll- als auch Istwerte der Stellgrößen eingelesen, um bei den HIL-Tests auch Komponentenausfälle abbilden zu können. Dazu zählen beispielsweise:

- der Wegfall eines oder mehrerer Istwerte durch Störungen der Winkelmessung,
- der Wegfall aller Istwerte durch Ausfall der I/O-Platine des FCC,
- der Wegfall eines oder mehrerer Sollwerte durch Störungen an den LSCs sowie
- Abweichungen von Soll- und Istwerten durch Fehler der Aktuatoren.

Durch die Messung der Sollwerte an den LSCs kann zusätzlich das Verhalten bei einem Ausfall des FCC getestet werden. In diesem Fall soll eine automatische Umschaltung der Kommandos vom FCC auf die Fernsteuerung des Sicherheitspiloten erfolgen, so dass die Aktuatoren weiterhin angesteuert werden können.

4.3 Modellierung von Sensorsignalen

Für die Regelung von ALEXIS benötigt der FCC die Messwerte der UAV-Sensoren. Alle Sensoren kommunizieren über je eine serielle Schnittstelle mit dem FCC. Aufgrund der fehlenden Flugzeugbewegung können die realen Sensoren keine nutzbaren Daten liefern. Sie müssen daher durch eine Simulation nachgebildet werden. Die simulierten Messwerte werden dann an den FCC übertragen.

Das Verhalten der vier Sensorsysteme (vgl. Kap. 2.3) wird mit Simulink nachgebildet. Es werden einfache, für das jeweilige Sensorsystem typische Fehlermodelle implementiert. Zusätzlich wird auf eine exakte Modellierung der Übertragungsprotokolle und Signaleigenschaften geachtet. Das Luftdatensystem und der Ultraschallhöhenmesser senden ihre Messwerte kontinuierlich, sobald sie vom FCC mit Strom versorgt werden. Das AHRS und das GPS hingegen werden vom FCC zunächst konfiguriert und ändern im Laufe der Konfiguration Eigenschaften der Signale (z.B. die Baud-Rate) bevor die Übertragung der Messwerte beginnt. Dieses Verhalten wird ebenfalls durch die Simulation nachgebildet.

Die Updaterate des Simulationsmodells beträgt 1200 Hz, damit die Emulationen der Sensoren mit den jeweiligen Updateraten der realen Sensoren (AHRS: 100 Hz, GPS: 4 Hz, USH: 16 Hz, ADT: 30 Hz) ablaufen können.

4.4 Untersuchung der Latenzzeiten

Die Latenzzeit der HIL-Simulation wird als die Zeit definiert, die zwischen dem Senden der Stellgrößenkommandos vom FCC und dem Empfang der Sensorwerte durch den FCC vergeht. Sie setzt sich aus drei Hauptprozessen zusammen:

- 1) Übertragung der Stellgrößen vom AFCS an den Simulationsrechner,
- 2) Berechnung des Simulationsmodells und
- 3) Übertragung der Sensorwerte an den FCC.

Die Dauer des Prozesses 1) hängt davon ab, ob die Soll- oder die Istwerte der Stellgrößen in der HIL-Simulation verwendet werden. Die Signale der Sollwerte werden über den CAN-Bus an die LSCs übertragen, wofür maximal 10 ms [7] benötigt werden (dieser Anteil der Latenzzeit ist auch im realen System vorhanden). Von den LSCs werden die Signale an den Simulationsrechner übertragen und dort verarbeitet, was im Bereich von Nanosekunden ge-

schieht und hier vernachlässigt wird. Die Gesamtübertragungszeit der Kommandos vom FCC an die Simulation beträgt somit maximal 10 ms. Bei Betrachtung der Istwerte addiert sich zu der Prozesszeit bei der Übertragung der Sollwerte die Dauer der Messung der Stellflächenposition und deren Übertragung über den CAN-Bus an den FCC (max. 20 ms, siehe [7]). Zusätzlich addiert sich die Totzeit aus der Phasenverschiebung zwischen Ankunft einer CAN-Nachricht und dem Senden der Daten an der seriellen Schnittstelle (Updaterate: 50 Hz) sowie die Dauer der Datenübertragung an den Simulationsrechner (2,2 ms für die Übertragung von 25 Byte mit 115,2 kBaud). Je nach Phasenverschiebung benötigt Prozess 1) somit zwischen 2,2 ms und 52,2 ms.

Prozess 2) benötigt für einen Simulationsdurchlauf 0,8 ms (Updaterate 1,2 kHz).

Der Prozess 3) (am Beispiel des AHRS) benötigt mindestens 2 ms (Übertragung von 23 Byte mit 115,2 kBaud). In Abhängigkeit der Phasenverschiebung addieren sich dazu im ungünstigsten Fall 10 ms aus der Updaterate der Sensoremulatation und weitere 10 ms aus der Updaterate der Software zum Einlesen der Sensordaten auf dem FCC.

Die mögliche Latenzzeit der HIL-Simulation kann (am Beispiel der Übertragung der AHRS Sensorwerte) somit bei Verwendung der Sollwerte der Stellgrößen zwischen 2,8 ms und 32,8 ms bzw. bei Verwendung der Istwerte zwischen 5 ms und 75 ms variieren.

5 VALIDIERUNG

Um den HIL-Simulator zur Entwicklung der Flugregelgesetze einsetzen zu können, muss der Simulator das Verhalten des UAVs exakt nachbilden. Dazu gehören ein korrektes Einlesen der Stellgrößen, eine exakte flugmechanische Nachbildung von ALEXIS und die genaue Implementierung der Sensoremulatation.

Die Validierung des flugmechanischen Modells von ALEXIS wird mit Daten aus Flugversuchen geschehen, ist aber nicht Bestandteil dieses Berichtes. Um zu zeigen, dass das flugmechanische Modell mit der gewählten Updaterate auf dem Simulationsrechner betrieben werden kann, wurden manuelle Flüge durchgeführt und die Daten aufgezeichnet. Ein Zeitschrieb der Position, sowie von Höhe über MSL (*Mean Sea Level*) H_{MSL} und dem Längslagewinkel Θ eines dieser Flüge ist in BILD 7 aufgetragen.

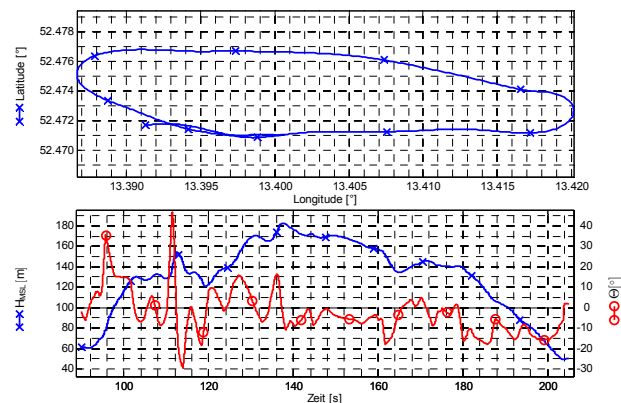


BILD 7. Position, Höhe H_{MSL} und Längslagewinkel Θ

Bei der Durchführung und Auswertung der Flüge fiel auf, dass das UAV in der Simulation dynamischer reagiert, als es in Flugversuchen der Fall war. Da für Test der FCCs jedoch eine möglichst genaue Nachbildung der Flugzeugbewegung nötig ist, sollte die in Kapitel 4.1 angesprochene, experimentelle Validierung des flugmechanischen Modells nachgeholt werden.

Die Validierung der Einleseroutine der Stellgrößen erfolgt durch einen Test, bei dem nacheinander alle Stellflächen durch Eingaben an der Funkfernbedienung betätigt und Vollausschläge vorgenommen werden. Dies wird zwei Mal durchgeführt. Im ersten Durchlauf wird der Auslegungsfall überprüft, in dem die Istwerte der Stellgrößen an der seriellen Schnittstelle des FCCs ausgelesen werden. Der zweite Durchlauf bildet den Ausfall der I/O-Platine des FCC nach. In diesem Fall entfallen alle Istwerte und es erfolgt im Simulationsmodell eine Umschaltung zu den PWM-Signalen der LSCs.

In beiden Fällen wird die gleiche Kommandoabfolge gewählt. Es erfolgen nacheinander Maximalausschläge an Querruder, Höhenruder, Seitenruder, Klappen, Spoilern und dem Schub. In BILD 8 sind exemplarisch die Zeitschriebe der Istwerte der einzelnen Stellgrößen dargestellt. In beiden Fällen sind die auf Eins normierten Werte der Stellgrößen gezeigt. Diese werden anschließend in Ausschlagwinkel umgerechnet und dem flugmechanischen Modell zur Verfügung gestellt. Es ist zu erkennen, dass sämtliche Stellgrößen korrekt eingelesen werden. Die Validierung der Einleseroutine der Stellgrößen wurde mit solchen Tests erfolgreich abgeschlossen.

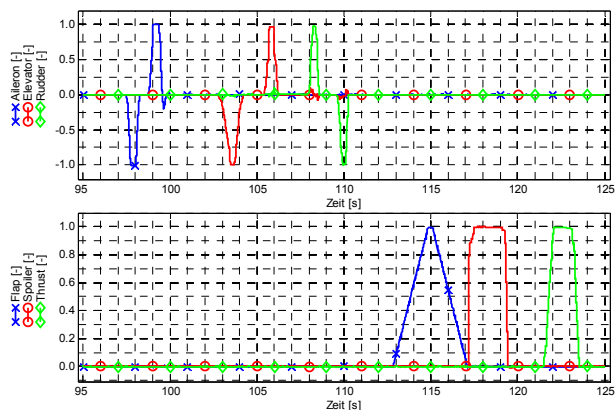


BILD 8. Validierung der Einleseroutine der Stellgrößen

Die Validierung der Sensoremulation beinhaltet zwei Untersuchungen. Erstens muss sichergestellt werden, dass der FCC nicht erkennt, ob mit seinen Schnittstellen die realen oder die simulierten Sensoren verbunden sind. Um dies zu zeigen wird auf dem FCC das Programm ausgeführt, das den Datenaustausch zwischen den Sensoren und den FCCs durchführt. Zuvor wird der HIL-Simulator mit den Schnittstellen des FCCs verbunden. Nach dem Start des Kommunikationsprogramms werden vom FCC für alle vier Sensoren Meldungen ausgegeben, dass der jeweilige Sensor erfolgreich an der dazugehörigen Schnittstelle initialisiert werden konnte. Dadurch ist gezeigt, dass der FCC alle Sensoremulationen als reale Sensoren erkennt und die Übertragungsprotokolle korrekt implementiert wurden.

Weiterhin muss garantiert werden, dass der FCC exakt die Informationen erhält, die aus der Simulation gesendet werden. Um dies zu zeigen wird ein virtueller Testflug mit dem HIL-Simulator durchgeführt. Dabei werden die simulierten Sensorwerte auf dem Simulationsrechner, sowie die vom FCC empfangenen Daten gespeichert. In BILD 9 sind die Sensorwerte aus beiden Datenaufzeichnungen exemplarisch für Längslagewinkel Θ und Hängewinkel Φ aus dem AHS dargestellt.

Bei der Analyse der Zeitschriebe sind zwei Abweichungen zwischen den beiden Datenaufzeichnungen ersichtlich: ein Zeitversatz zwischen den beiden Signalen und Unregelmäßigkeiten in der FCC-Datenaufzeichnung. Mit Hilfe von Kreuzkorrelationen der beiden Zeitschriebe von Längslage- und Hängewinkel wurde der Zeitversatz genauer untersucht. Er beträgt im Mittel 23 ms. Dies liegt im Bereich der in Kapitel 4.4 beschriebenen Latenzzeiten bei der Übertragung der AHS Signale. Bei den Unregelmäßigkeiten in der FCC-Datenaufzeichnung handelt es sich um ca. 50-200 ms andauernde Phasen, in denen keine Aktualisierung des Sensorwertes stattfindet. Diese Phase treten mit einer Frequenz von ca. 1 Hz auf. Die Ursachen dafür liegen nicht in der Kommunikation zwischen FCC und HIL-Simulator. Vergleiche mit Flugversuchsdaten zeigen, dass dort vergleichbare Unregelmäßigkeiten auftreten. Analysen der Timing-Genauigkeit der Software auf dem FCC in [4] haben gezeigt, dass die nominelle Updaterate der Software von 10 ms nicht exakt eingehalten wird. Die Gründe werden hier dem FCC-Betriebssystem Windows XPe (XP embedded) zugeschrieben und müssen weiter untersucht werden. Damit wurde bereits ein erstes Ergebnis für die AFCS-Entwicklung durch die HIL-Tests erzielt.

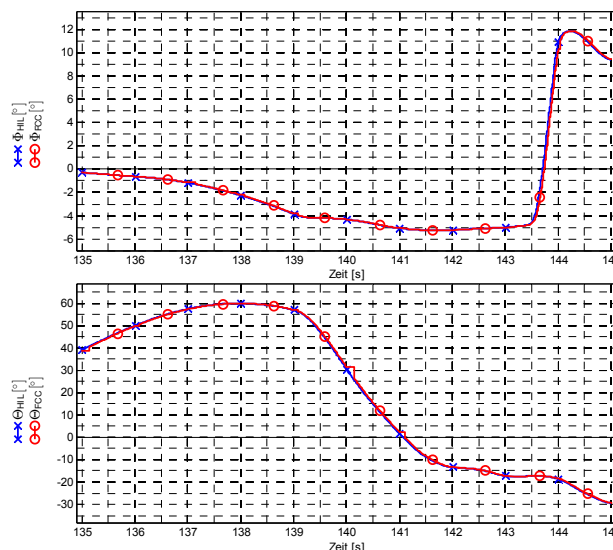


BILD 9. Validierung der Sensoremulationen

Trotz der beschriebenen Abweichungen zwischen den Datenaufzeichnungen wird die Übertragung der Sensor-signale zwischen Simulation und FCC als korrekt angesehen.

6 DURCHFÜHRUNG VON HIL-TESTS

Dieses Kapitel beschreibt die Durchführung von Tests mit dem Hardware-in-the-Loop – Simulator des IFSys Projektes. Das Testkonzept verfolgt das Ziel, HIL-Tests weitestgehend mit MATLAB zu automatisieren. Die Automatisierung ermöglicht es, reproduzierbare Ergebnisse zu generieren. Eine einfach zu bedienende Benutzeroberfläche soll es den Studierenden erleichtern, den komplexen Ablauf von HIL-Tests schnell zu verstehen und umzusetzen. Dies reduziert den Einarbeitungsaufwand und die Wahrscheinlichkeit von Bedienfehlern. Da alle Anschlüsse am Iron Bird gut zugänglich sind, werden für die Kopplung mit dem Simulationsrechner weniger als 5 min benötigt.

6.1 Bedienkonzept

Es wurde darauf geachtet, dass möglichst wenig manuelle Schritte zur Durchführung von HIL-Tests notwendig sind. Die Anzahl manueller Schritte hängt jedoch stark davon ab, ob Änderungen an dem Simulationsmodell der Systemumgebung vorgenommen werden. Bei erstmaliger Verwendung eines neuen Simulationsmodells müssen folgende Arbeitsschritte vom Testingenieur manuell durchgeführt werden:

- Kompilieren des Simulationsmodells,
- Anlegen eines Testprojektes und Einbinden des kompilierten Modells in dieses, sowie
- Definition von Aufzeichnungsparametern.

Die letzten beiden Schritte lassen sich vereinfachen, indem Projekte von früheren Modellversionen kopiert und an das neue Simulationsmodell angepasst werden.

Wurde der HIL-Simulator wie zuvor beschrieben eingerichtet, erfolgt die Durchführung der HIL-Tests weitestgehend automatisiert. Bei Modifikationen an der Aktuatorik ist vor einem HIL-Test eine Kalibration der Stellgrößen notwendig. Dies ist ebenfalls mit hierfür entwickelten MATLAB-Funktionen automatisiert. Der Testingenieur muss nur eine Funktion ausführen und den Anweisungen auf dem Bildschirm des Host-PCs folgen. Die Steuerung des Simulationsrechners, die Datenaufzeichnung und die Berechnung der Kalibrationswerte erfolgt automatisch.

Der Testingenieur kann einen HIL-Test über eine grafische Benutzeroberfläche in MATLAB bedienen (siehe BILD 10). In dieser Oberfläche kann er sämtliche Einstellungen vornehmen:

- Auswahl eines Testprojektes,
- Auswahl aufzuzeichnender Parameter,
- Konfiguration der Datenauswertung mit dem Plot-Werkzeug TUBPlot des Fachgebietes,
- Auswahl der möglichen Visualisierungsoptionen:
 - FlightGear und/oder
 - Bodenkontrollstation.



BILD 10. Grafische Benutzeroberfläche für HIL-Tests

Nach Abschluss aller Einstellungen durch den Testingenieur wird der HIL-Test mit den folgenden Schritten automatisch durchlaufen:

- Aufruf der Kontrollsoftware für den Simulationsrechner,
- Laden des Testprojektes,
- Konfiguration der Datenübertragung zwischen Simulationsrechner und Host-PC,
- Start der Visualisierungsprogramme,
- Start der HIL-Simulation und der Datenübertragung sowie
- Durchführung der Datenauswertung nach Beendigung des HIL-Tests.

Ein manueller Eingriff des Testingenieurs ist nur zum Senden von Befehlen an den FCC notwendig, z.B. zum Start einer Flugaufgabe.

6.2 Visualisierungsoptionen

Zur Visualisierung der Flugzeugbewegung während der Hardware-in-the-Loop – Tests stehen zwei Optionen zur Verfügung. Zum einen kann die Bewegung des UAV wie im realen Flugbetrieb auf der in [10] entwickelten Bodenkontrollstation nachverfolgt werden. Deren Darstellung ist in BILD 11 gezeigt. Neben der Visualisierung des Flugweges auf einer Karte und der Flughöhe in dem darunterliegenden Diagramm, bietet die Bodenkontrollstation auch ein Primary Flight Display. Auf diesem werden die EULER-Winkel, die Flughöhe, die Fluggeschwindigkeit und die Vertikalgeschwindigkeit angezeigt. Zusätzlich werden die Stellgrößenausschläge von ALEXIS dargestellt. Zur Visualisierung kann entweder die Bodenkontrollstation inklusive ihrer Hardware verwendet werden, wobei die Daten vom FCC über das Funkmodem (siehe BILD 3) an den Bodenkontrollrechner gesendet werden, oder es wird nur die Software der Bodenkontrollstation verwendet, wobei Daten vom Simulationsrechner an den Host-PC des HIL-Simulators übertragen und dargestellt werden.

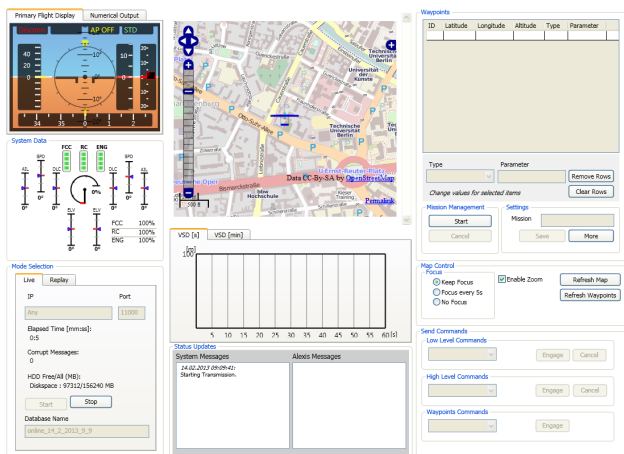


BILD 11. Visualisierung eines HIL-Tests auf der Bodenkontrollstation

Als zweite Visualisierungsoption steht die Flugsimulationssoftware FlightGear zur Verfügung. Mit FlightGear wird die simulierte Bewegung des Fluggerätes mit einem 3D-Modell von ALEXIS dargestellt. Dabei kann zwischen verschiedenen Perspektiven gewählt werden: die Cockpitsicht, die Verfolgungsansicht des UAV sowie eine Ansicht, wie sie der Sicherheitspilot vom Boden aus hätte. Somit eignet sich diese Darstellung zum Training des Sicherheitspiloten, beispielsweise für den Fall eines Ausfalls des Flugsteuerungscomputers.



BILD 12. Visualisierung eines HIL-Tests mit FlightGear

7 FAZIT UND AUSBLICK

Im Rahmen des Studierendenprojektes IFSys wurde ein Hardware-in-the-Loop – Simulator aufgebaut. Dieser wird erfolgreich zum Testen der Flugregelgesetze und des automatischen Flugsteuerungssystems eingesetzt. Erste HIL-Tests in Vorbereitung von Flugversuchen haben Erkenntnisse zur Weiterentwicklung des flugmechanischen Modells von ALEXIS geliefert. So ist es nötig, das Modell insbesondere in Bezug auf die Flugzeuglängsbewegung zu überarbeiten, da das Modell hier wesentlich dynamischer reagiert (Reaktion auf Stellkommandos und Amplitude der Eigenschwingungen) als das reale Flugzeug.

Das Konzept des HIL-Simulators hat sich bewährt. Das für die Durchführung von HIL-Tests entwickelte Bedienkonzept hat sich als leicht verständlich und gut zu bedienen

erwiesen, so dass Versuche nach nur kurzer Einarbeitungszeit durchgeführt werden können. Der gesamte HIL-Simulator kann leicht transportiert werden und wurde auch auf Messen zur Demonstration des IFSys-Projektes eingesetzt.

Der modulare Aufbau des HIL-Simulators ermöglicht eine Anpassung an zukünftige Erweiterungen des automatischen Flugsteuerungssystems von ALEXIS.

SCHRIFTTUM

- [1] Advisory Council for Aeronautics Research in Europe: *EUROPEAN AERONAUTICS: A VISION FOR 2020*, European Commission, 2001
- [2] Marwedel, Peter: *Eingebettete Systeme*, Springer-Verlag Berlin Heidelberg, 2008
- [3] Walde, Georg; Hamann, Alexander: *ALEXIS - Friedliche Dohne für die Lehre*, DGLR Luft- und Raumfahrt 3 (2012), Mai - Juni, S. 28-29
- [4] Lieb, Johannes: *Entwicklung der Flugsteuerungssoftware für den Bordrechner des UAV ALEXIS*, Technische Universität Berlin, Studienarbeit, September 2009
- [5] Brockhaus, Rudolf; Alles, Wolfgang; Luckner, Robert: *Flugregelung*, 3. Auflage, Springer-Verlag, Berlin Heidelberg, 2011
- [6] Büttig, Björn: *Der Einsatz von Hardware-in-the-Loop-Technologie in Virtuellen Test Centern*, Shaker Verlag, 2010
- [7] Klass, Sven: *Konzipierung und Hardwareentwicklung eines Umschaltsystems für die automatische und manuelle Flugsteuerung des unbemannten Luffahrzeuges ALEXIS*, Technische Universität Berlin, Studienarbeit, 2013
- [8] AIRBUS S.A.S: *Iron Bird*, <http://www.airbus.com/innovation/proven-concepts/in-design/iron-bird/>, Version: 04. Januar 2013
- [9] Unruh, Oliver: *Flugmechanisches Modell des UAV ALEXIS zur Analyse der Flugleistungen*, Technische Universität Berlin, Studienarbeit, Oktober 2008
- [10] Krishnamurthy, Vikram: *Bodenstation für das Unmanned Aerial Vehicle (UAV) ALEXIS - Konzept und erste Realisierung*, Technische Universität Berlin, Masterarbeit, 2012