

Diplomarbeitspräsentation



Optimierte Verifikation des Airbus A380 Slat Flap Control Computers

Diplomand: Timo Hauschildt

1. Prüfer: Prof. Dr.-Ing. Dieter Scholz, MSME
2. Prüfer: Dipl.-Ing Roland Gad



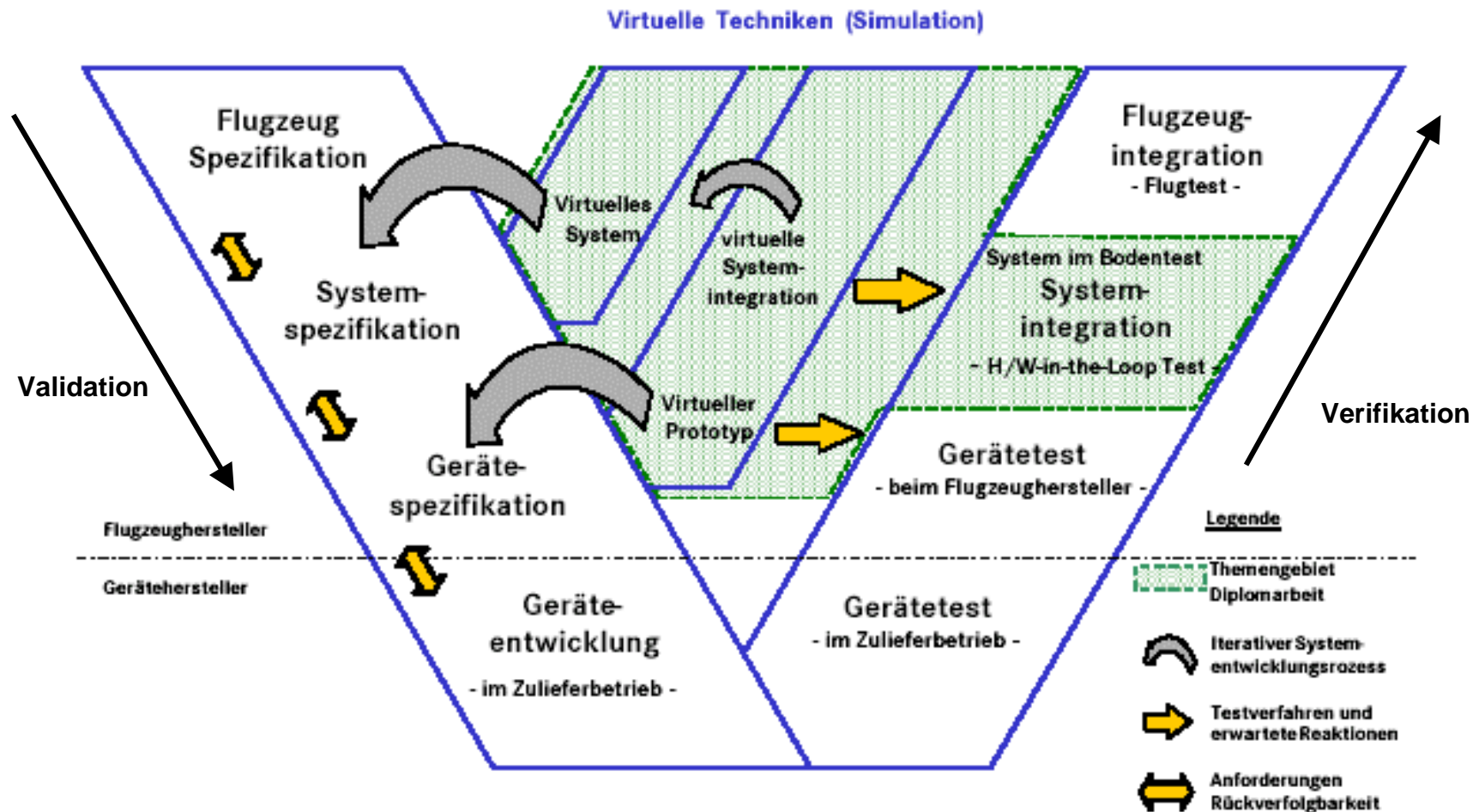
Inhalt

1. Einleitung
2. Slat Flap Control Computer (SFCC)
3. High Lift System Development Simulator (HLSDS)
4. Erstellung eines Testvektors (Validation)
5. Testvektorimplementierung
6. Beschreibung von Verifikationstests mit Hilfe des erstellten C-Programms
7. Darstellung und Diskussion der Testergebnisse

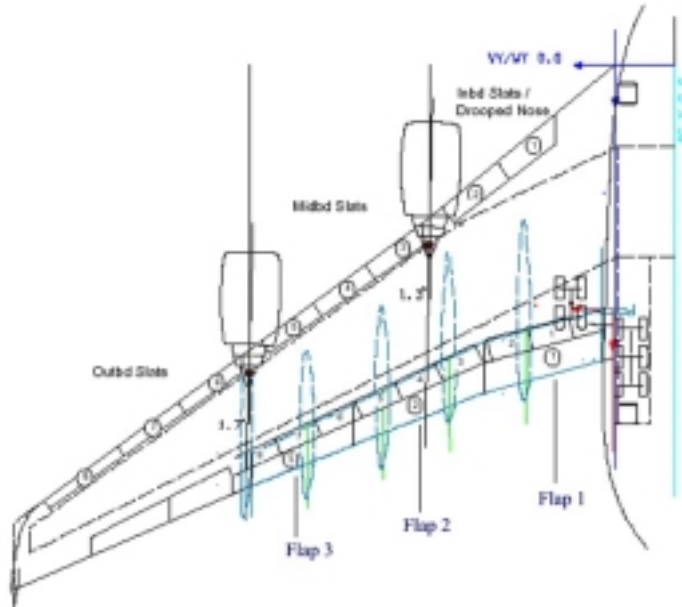


1. Hintergrund zur Diplomarbeit

- Erstellung einer Verbindung zwischen Validation und Verifikation



1. Hochauftriebssystem des Airbus A380



- Allgemein:
 - Betriebsdruck der Hydraulikmotoren = 5000 psi
 - CFK Antriebswellen
 - 250 Systemkomponenten
- Vorderkante:
 - 6 x Slat + 2 x „Droop Nose“ pro Flügel
 - Hybridantrieb (E-Motor/H-Motor)
- Hinterkante:
 - 3 x Flap (Einfachspaltklappen) je Flügel
 - Fläche der Landeklappen = 120 m²

2. Slat Flap Control Computer (SFCC)



- „Fly-by-Wire“ Rechner
- Zuständig für die Steuerung und Überwachung des Hochauftriebssystems
- Ansteuerung über die „Command Sensor Unit“ (CSU)
- redundante Auslegung im Flugzeug (2 SFCCs pro A/C)



3. High Lift System Development Simulator (HLSDS)

- Aufgaben und Systemeigenschaften:
 - Widerspiegelung des High Lift Systems des A380 durch mathematische Simulationsmodelle
 - Echtzeitfähigkeit der Simulation des funktionalen Verhaltens der Systemkomponenten
 - Schnittstelle Mensch \leftrightarrow Maschine für alle Cockpit Kontroll Funktionen zur Simulation von Piloteneingaben
 - Bereitstellung der Systemsignale, wie sie auch im original System vorhanden sind



3. High Lift System Development Simulator (HLSDS)

- Nutzungsmöglichkeiten des HLSDS:
 - Verifikation und Zertifizierung des A380 High Lift System Controllers (SFCC)
 - Testwiederholungen infolge SFCC Updates oder Modifikationen
 - Fehlersuche
 - Entwicklungstests



3.1 Hardwarekomponenten des HLSDS

- Hardware:
 - 4 Simulation Processing Units (SPUs)
 - Test Unit Adapter (TUA)
 - ARINC600 Gerätestecker für SFCC
 - 12 Anschlussstellen für High Lift System Komponenten
 - Signal conditioning
 - Break Out Panel (BOP)
 - Simulation Host



3.2 Software des HLSDS (Avionics Development System 2nd Generation)

- Aufgaben:
 - Ablaufsteuerung der Testläufe
 - Konfiguration der Hardware- und Softwareumgebung
 - Online Visualisierung von Testdaten
 - Offline Visualisierung von Testdaten
 - Aufzeichnung der Testdaten
 - Bereitstellung interaktiver Schnittstelle für externe Applikationen



4. Testvektorerstellung (Validation)

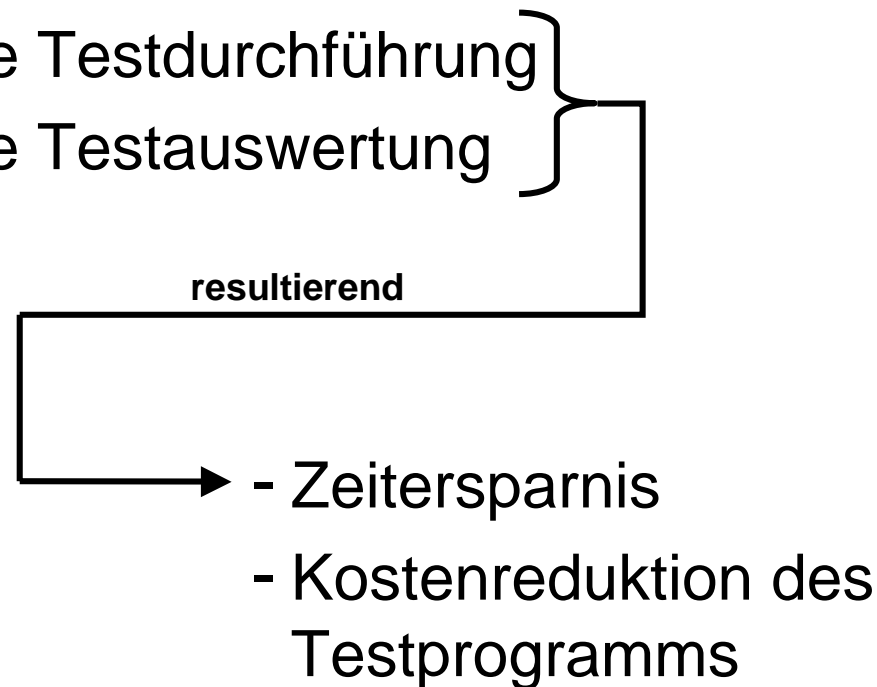
- Definition „Testvektor“:
 - Testparameter in Abhängigkeit der Zeit
- Erstellung:
 - „Virtual Prototyping“ (Matlab/Simulink; Scade; MTest)
 - Archivierung als Matlab- oder Excel Format
- Durchführung:
 - Testdurchführung gegen Systemanforderungen
 - Anhand textuell erstellter Testabläufe



4.1 Testvektorimplementierung

- Aufgabe:
 - Formatierung der Testvektoren in ein weiter zu verarbeitendes Format
 - Echtzeit Einbindung der Testvektordaten in den Testablauf des HLSDS

- > Optimierte Testdurchführung
- > Optimierte Testauswertung



4.2 Umsetzung

- Vorbereitende Maßnahmen:
 - Umwandlung des Testvektor in eine Textdatei
 - Anpassung der „sample rate“ an den Simulator
- Ausführung durch Matlab Programm „Datenwandler.m“
 - Importieren der Testvektordaten in den Matlab Wokspace
 - Öffnen einer beschreibbaren Textdatei
 - Parameterbenennung
 - Einlesen und strukturiertes exportieren der Testvektordaten in die Textdatei



4.1 Umsetzung

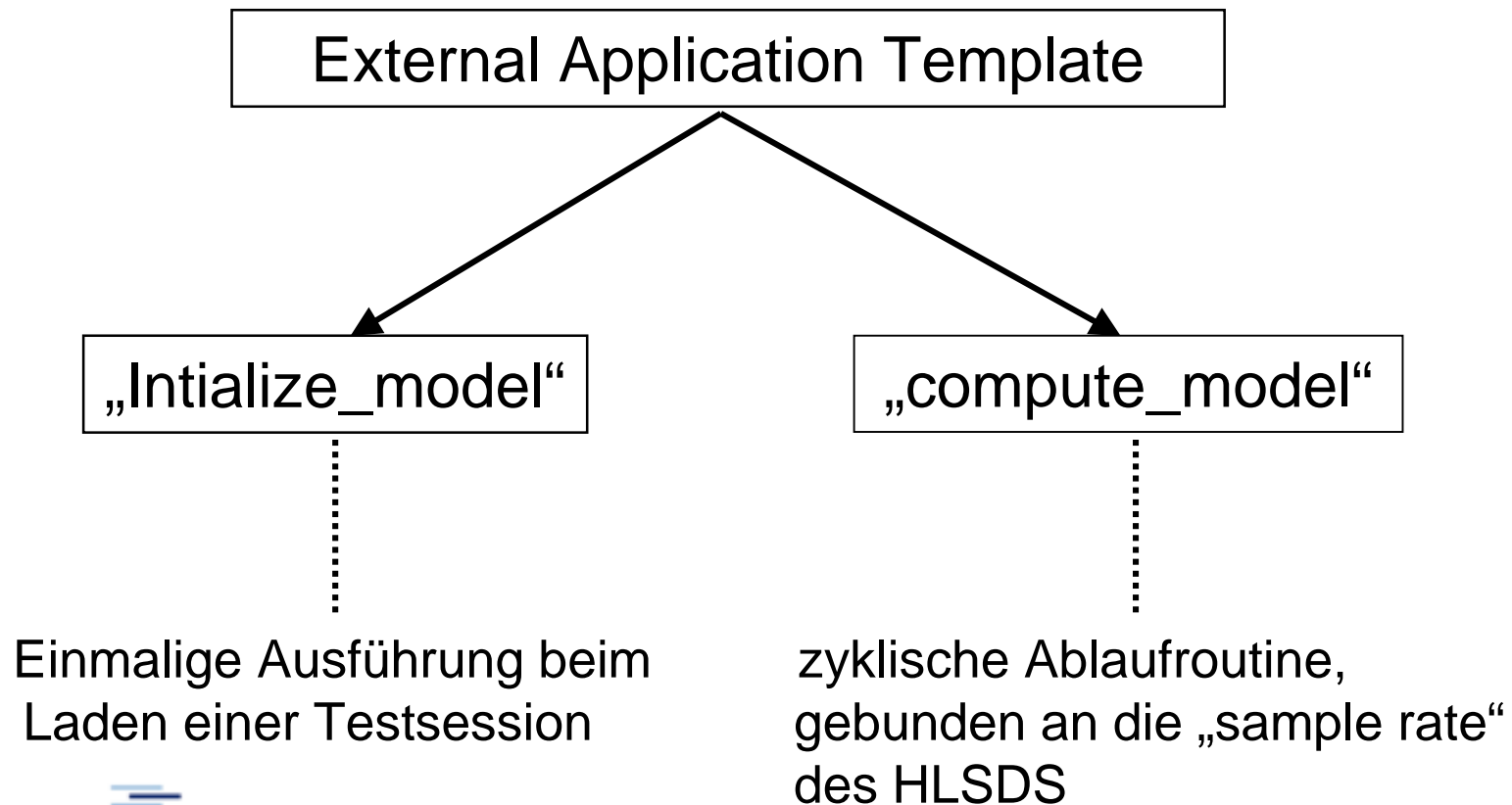
Das Hauptprogramm der Testdatenimplementierung ist in zwei Unterprogramme gegliedert.

- Aufgabe „Read-Funktion“:
 - Strukturiertes, zeilenweises Auslesen der Textdatei (von lokaler Festplatte)
 - Übertragung der Testdaten in den Speicherbereich des Echtzeitrechners (Simulator)
 - Anlegen einer Datenstruktur (pro Zeitschritt ein Datenblock)
- Aufgabe „Write-Funktion“:
 - „Echtzeit-Übergabe“ der Parameterdaten an die HLSDS internen Parameter



5. Testvektorimplementierung

- Einbindung des C-Programms in die Testumgebung über das ADS2-Werkzeug „devExchange“



5.1 Erstellen eines ausführbaren Tests

- Folgende Dateien sind zur Durchführung eines Tests erforderlich:

**A
b
f
o
l
g
e**



1. SID Datei (**S**ystem **I**nterface **D**efinition)
2. Ausführungsdatei (executable)
3. Komponent Datei
4. Panel Datei
5. Session Datei (ausführbarer Test)

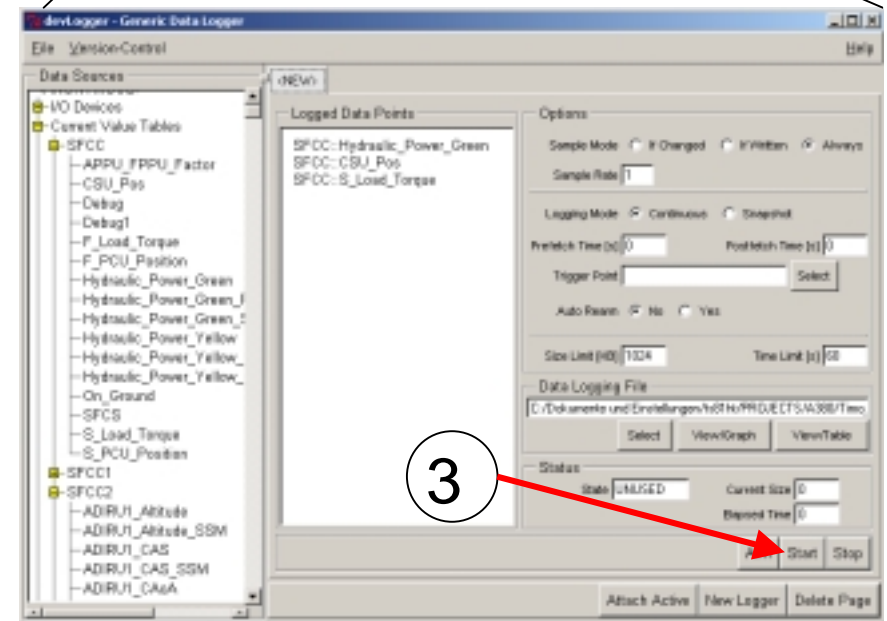


6. Beschreibung von Verifikationstest mit Hilfe des erstellten C Programms

• Ablauf:



1. Laden der Testsession
2. Konfiguration des Daten-Logger
3. Starten des Daten-Logger
4. Starten der Testsession



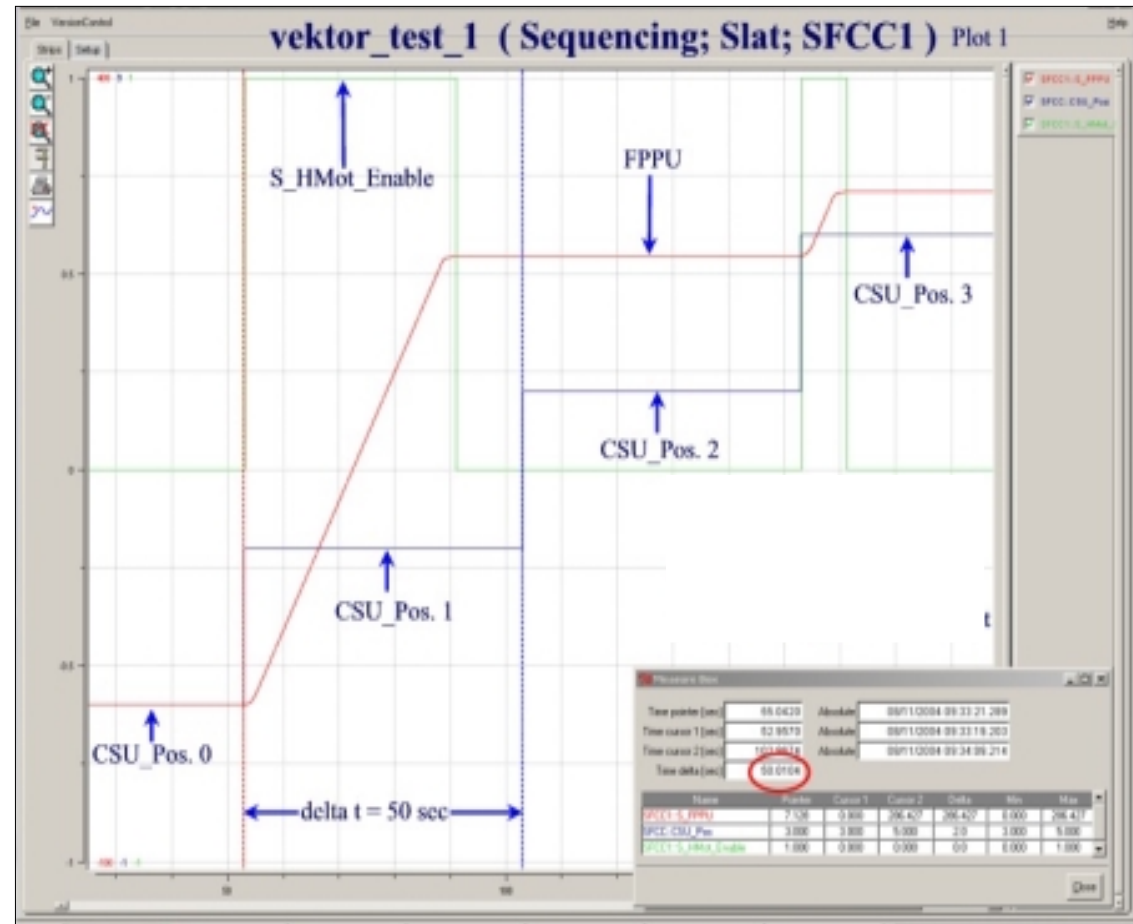
7. Darstellung und Diskussion der Testergebnisse

- 1. Testphase:

- Kontrolle, ob Input Parameter korrekt an das System übergeben werden

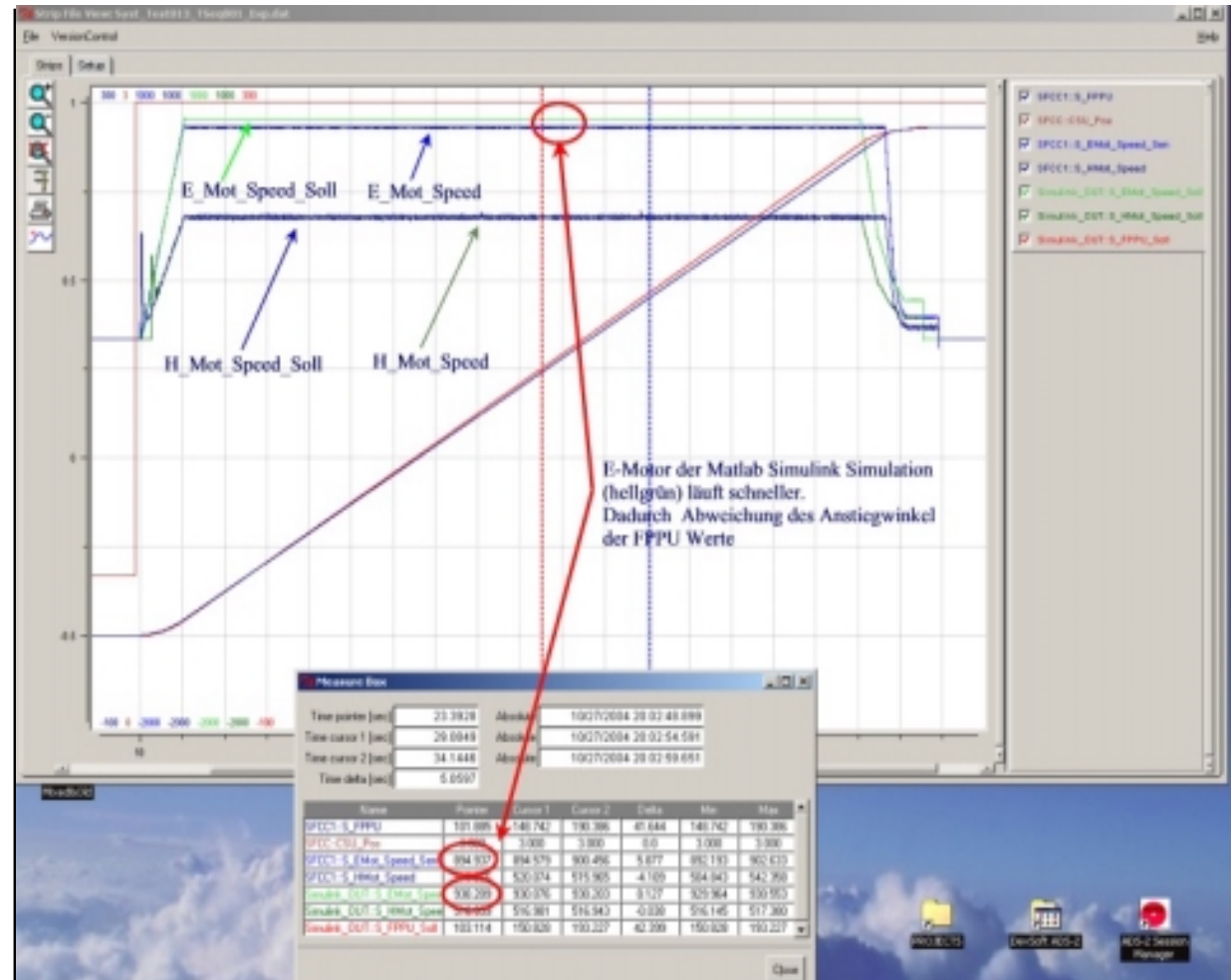


- Abgleich zwischen graphischer Auswertung und textuellem Testvektor



7. Darstellung und Diskussion der Testergebnisse

- 2. Testphase
 - Optimierte Testauswertung
- Methodik**
- ↓
- Implementierung von Input und Soll-Output Parametern in den Testablauf
 - Graphischer Abgleich Soll-/Ist-Parameter



Vielen Dank für Ihre Aufmerksamkeit
Haben Sie Noch Fragen?

