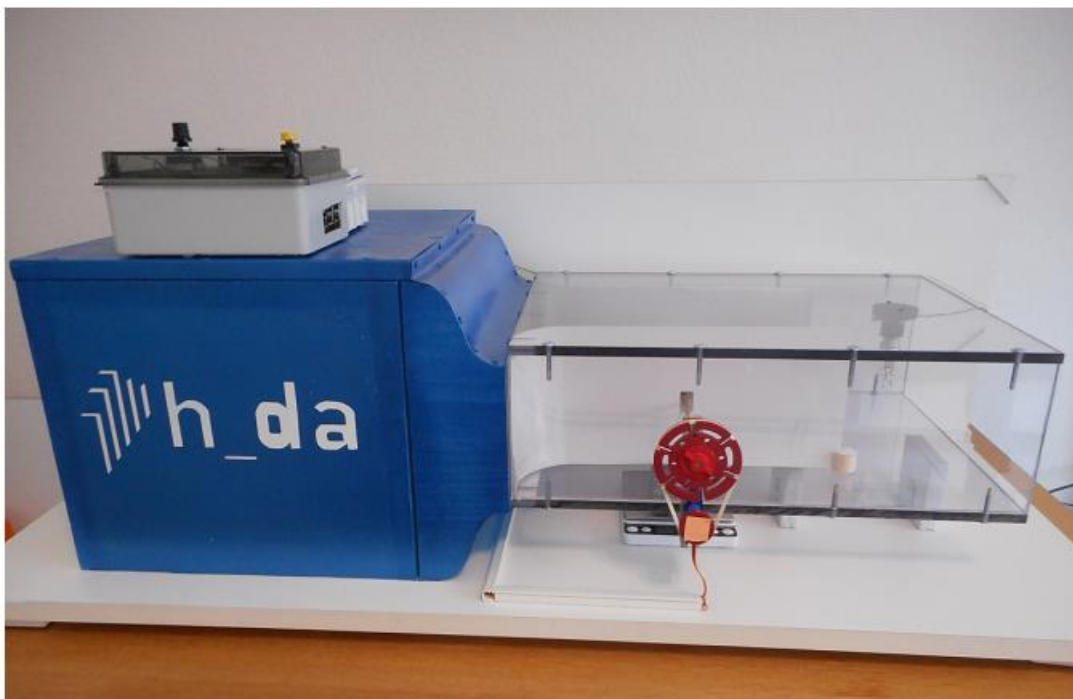


H. Ahuis
T. Höber
P. Ros Leranoz
W. Steinnagel
A. Zörner

Hochschule Darmstadt h_da
Studiengang
Master of Science - Elektrotechnik

Projektarbeit C22

Entwurf und Realisierung eines Rapid Control Prototyping-Systems mittels Matlab PLC Coder für CoDeSys nach der modellbasierten Softwareentwicklungs-Methode, durch Generierung von Structured Text-Codes aus MATLAB-Simulink-Modellen.



Die übergeordnete Problemstellung dieses Projektes beinhaltet die Messung der konstruktiven Auslegung von Flügelprofilen, wie Sie im Flugzeugbau verwendet werden. Je nach Form und Ausmaße des Profils ergeben sich unterschiedliche physikalische Eigenschaften. Die wesentlich wichtigen unbekannteten Eigenschaften einer neuen Flügelprofilform sind Auftriebskräfte, Widerstandskräfte und der Strömungswiderstandskoeffizient. Diese Projektarbeit beinhaltet den Steuerungsentwurf mittels MBSE für einen Strömungskanal zur Messung der Auftriebskräfte und der anschließenden Ermittlung der Auftriebsbeiwerte.

Ziel des Projektes ist die Auslegung einer Steuerung für einen realen Windkanal-Prüfstand unter Verwendung der modellbasierten Software-Entwicklung (MBSE). Mit dem Windkanal-Prüfstand soll in unterschiedlichen Abläufen unter Variation von bestimmten Betriebsparametern charakteristische Eigenschaften eines Flügelprofils ermittelt werden.

Inhaltsverzeichnis

INHALTSVERZEICHNIS	3
1. EINLEITUNG	5
1.1 THEMA DES PROJEKTES	5
1.2 PROBLEMSTELLUNG.....	5
1.3 ZIEL DES PROJEKTES.....	5
2. ANFORDERUNGSANALYSE	7
2.1 PROJEKTAUFTRAG	7
2.2 PROJEKTZEITPLAN	8
2.3 UMFELD-/ STAKEHOLDERANALYSE.....	10
2.4 RISIKOANALYSE.....	11
2.5 KOSTENERMITTLUNG	12
3. THEORETISCHE GRUNDLAGEN	13
3.1 MODELLBASIERTE SOFTWAREENTWICKLUNG	13
3.1.1 Konzept der modellbasierten Softwareentwicklung.....	13
3.1.2 Simulation in MBSE.....	14
3.2 RAPID CONTROL PROTOTYPING	15
3.2.1 Vorgehensweise bei Rapid Control Prototyping	16
3.3 PLC CODER	16
3.3.1 Die Bedeutung von PLC.....	16
3.3.2 Simulink PLC Coder	16
3.3.3 Arbeiten mit dem Simulink PLC Coder	17
3.3.4 Code Optimierung und IDE Integration	18
3.4 FUNKTION VON CODESYS.....	20
3.4.1 Anleitung zur Installation von CODESYS für Raspberry-Pi	21
4. TECHNISCHE REALISIERUNG	24
4.1 SYSTEM- UND SOFTWAREENTWURF	24
4.2 HARDWAREMODELL	26
4.3 SOFTWAREMODELL	30
4.4 TESTEN DER STEUERUNGSSOFTWARE	35
5. ZUSAMMENFASSUNG	40
6. AUSBLICK	41
ABKÜRZUNGSVERZEICHNIS	42
LITERATURVERZEICHNIS	43
ABBILDUNGSVERZEICHNIS	44
TABELLENVERZEICHNIS	46
7. ANHANG	47
AUSLEGUNG DER SENSOREN	47
THERMISCHE ANEMOMETRIE THEORETISCHE GRUNDLAGEN	54
IMPLEMENTIERUNG.....	55
SOFTWARE	56

Zusammenfassung

Ziel des vorliegenden Projektes ist es, das Konzept der modellbasierten Software-Entwicklung zur Funktionsrealisierung eines Strömungskanal-Modells zu verifizieren. Auf Grundlage dieser Vorgehensweise soll für den vorliegenden Testaufbau eine anwendungsspezifische Software entwickelt werden, die es dem Bediener ermöglicht Strömungsgeschwindigkeit des im Kanal fließenden Luftstromes und Anstellwinkel eines in diesem Luftstrom befindlichen Flügelprofils zu variieren um somit ca-Wert und Auftriebskräfte des eingesetzten Profils zu ermitteln. Es zeigt sich, dass das Verfahren der modellbasierten Software-Entwicklung in Matlab/Simulink und die darauf folgende Kompilierung und Implementierung auf die Zielhardware in Form eines Raspberry Pi durchgeführt wird und das Hardwaremodell des Strömungskanals die entworfene Funktion, in diesem Fall die steuerungstechnische Variation der Profilneigung, für den Bediener bereitgestellt. Es lassen sich somit einfache und komplexe Steuerungsabläufe durch Modelle erzeugen, die auf entsprechender Zielhardware, wie speicherprogrammierbare Steuerungen, den vollen Funktionsumfang wie von Programmen abbilden, die durch die gewöhnlichen nach DIN EN 61131 etablierten Programmiersprachen wie Funktionsbausteinsprache, Anweisungsliste oder Kontaktplan erstellt worden sind.

Anzumerken ist, dass in diesem Projekt die Ansteuerung der Lüfter nicht auf dem Verfahren der modellbasierten Softwareentwicklung stattfindet, sondern durch eine Implementierung auf einem Mikrocontroller gelöst worden ist. Der Grund hierfür liegt in der dauerhaften zeitkritischen Erzeugung des PWM-Signal. Auf der Codesys-Runtime des Raspberry Pi würde diese zusätzliche, neben der PWM für die Servomotor-Ansteuerung, Signalerzeugung die Einhaltung der maximalen Pulsbreitenweite gefährden. Es wurde sich darauf geeinigt nur die Ansteuerung des Servomotors durch den modellbasierten Entwurf zu realisieren.

Neben diesem Software-Funktionsumfang lässt sich über den Teststand zusammenfassen, dass die in Software erzeugten Funktionen von dem Hardwaremodell durchgeführt werden. Weiterhin besteht neben der Anzeige der aktuellen Auftriebskraft auch die Möglichkeit, die Strömungsgeschwindigkeit zu erfassen.