



Dominik Mähr, BSc

Entwicklung einer Mess- und Visualisierungssoftware an einem Hochdrehzahlprüfstand für automotive Sensorik

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Elektrotechnik

eingereicht an der

Technischen Universität Graz

Betreuer

Dipl.-Ing. Dr.techn. Jürgen Fabian

Dipl.-Ing. Jens Gächter, BSc

Institut für Fahrzeugtechnik

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift

Danksagung

Mit dem Abschluss dieser Masterarbeit endet nun einer der wohl prägendsten Abschnitte meines Lebens, die Studienzeit. Ich möchte diese Seite nutzen, um mich bei allen die mich auf diesem Weg begleitet und unterstützt haben zu bedanken.

Ich möchte mich für die wirklich tolle, unkomplizierte Betreuung und die fachliche Unterstützung bei Jürgen Fabian und ganz speziell bei Jens Gächter bedanken: „Dr. Love had always time for me.“ Ich möchte mich überhaupt beim gesamten FTG für das angenehme Arbeitsklima und auch für die externen Aktivitäten (vor allem: Bologna) bedanken.

Ein ganz besonderer Dank gebührt natürlich meiner Familie, allen voran meinen Eltern Toni und Hildegard, die mir das Studium überhaupt erst ermöglichten und meinen Geschwistern Alex, Betty und Tobi für die allgegenwärtige und herzliche Unterstützung.

Herberge, Männerhort und Partytempel, dies vereint NUR das *RickJames-Penthouse*. Ich danke meinen drei duften Mitbewohnern Wasi, Ramson und Tschensi für eine legendäre Zeit, von der wir wohl noch unseren Kindern berichten werden. Es war end-laser, ich danke euch!

Einen weiteren zentralen Punkt meiner Studienzeit bildete der *Dynamobauzeichensaal (DZS)*. Neben Platz zum Lernen und Feiern erfolgreicher Prüfungen sind es vor allem die entstandenen Freundschaften die dem *DZS* einen speziellen Platz in meinem Herzen sichern. Speziell hervorheben möchte ich den sogenannten *Schwingkreis* und die Anhänger des *DZS-Weisenrats*, die vor allem während der Zeit des Erfassens dieser Arbeit für eine wichtige Auflockerung des tristen Arbeitsalltags sorgten.

Auch allen Freunden aus meiner Heimat Vorarlberg rund um die Gruppen *Schnifis+*, *bifi*, *Die Käärgemeinschaft* und der *Gemeindemusik Schnifis* möchte ich danken. Trotz monatelanger Abwesenheit hatte ich bei den Besuchen daheim stets einen Platz in diesen großartigen Gemeinschaften.

Zu guter Letzt möchte ich meiner Freundin Theresa danken, die immer sehr verständnisvoll war und mich laufend motiviert hat.

Inhaltsverzeichnis

Danksagung	iii
Kurzfassung.....	vi
Abstract	vii
Abkürzungen.....	viii
1 Einleitung.....	10
2 Grundlagen.....	12
2.1 Rotierende Positionssensoren	12
2.1.1 Sensoren mit rotierbaren Wechselfeldern.....	13
2.1.2 Wirbelstromsensoren	16
2.1.3 Hall-Sensoren.....	16
2.1.4 Magnetoresistive Sensoren	17
2.2 Sensorfehler	17
2.2.1 Amplitudenfehler bei Resolvieren.....	18
2.3.2 Messung des Winkelfehlers.....	21
2.3 Statistische Versuchsplanung	23
2.3.1 Vollfaktorieller Versuchsplan	23
2.3.2 Central-Composite Design.....	24
2.3.3 Box-Behnken-Design	26
3 Prüfstand	28
3.1 Anforderungen.....	28
3.2 Aufbau.....	29
3.3 Komponenten	30
3.4 Konzept Automatisierung	32
3.4.1 Konzept LabVIEW	33
3.4.2 Alternatives Konzept	34
4 Anbindung der Komponenten.....	35
4.1 Elektrischer Antrieb	35
4.1.1 PROFINET	35

4.1.2	PN Driver.....	38
4.1.3	PROFIdrive	39
4.2	Positioniersystem.....	42
4.2.1	RS-232	42
4.2.2	Universal Serial Bus	43
4.3	Temperatursystem.....	44
4.3.1	RS-485.....	44
4.3.2	Kommunikationsprotokoll.....	45
4.4	Spannungsversorgung.....	46
4.5	Messdatenerfassung.....	46
4.5.1	Null-Modem Emulator.....	47
4.5.2	AK-Protokoll.....	47
5	Implementierung in LabVIEW	49
5.1	LabVIEW.....	49
5.2	POSE_Manual.....	50
5.3	POSE_Auto	52
6	Zusammenfassung.....	58
	Abbildungsverzeichnis	59
	Tabellenverzeichnis	61
	Literaturverzeichnis.....	62
A	Anhang	64
A.1	Geräteverzeichnis	65
A.2	Netzwerk- und Schnittstellenkonfiguration.....	65
A.3	Implementierung in Matlab — Erstellung der Versuchspläne	68

Kurzfassung

Die Erfassung der Rotorlage ist ein entscheidendes Element im elektrifizierten Antriebsstrang. Am Institut für Fahrzeugtechnik wird in Kooperation mit MAGNA Powertrain GmbH & Co KG ein Prüfstand für die Evaluierung rotierender Positionssensoren aufgebaut. Diese werden bezüglich ihrer Einsetzbarkeit unter gewissen Parametern, wie Fehlpositionierung und Variation von Drehzahl, Temperatur und Versorgungsspannung getestet und bewertet. Der Prüfstand besteht aus mehreren Komponenten, für die im Zuge dieser Masterarbeit eine einheitliche Bedienoberfläche zur Kontrolle und Steuerung realisiert wurde. Außerdem wurde ein automatisierter Prüfablauf implementiert, um verschiedene Testszenarien zu erstellen und durchzuführen.

Abstract

The sensing of the position angle is a key component in automotive traction drive applications. The Institute of Automotive Engineering in cooperation with MAGNA Powertrain GmbH & Co KG is building up a test bench for the evaluation of angular sensor systems for rotor position sensing. These sensors get tested in regard to mechanical displacement, speed variations, influence of temperature and varying supply voltage. The test bench consists of several components to access these parameters. During this master thesis a consistent software for controlling and visualization of the test bench components was made. Additionally, an automatic test cycle was implemented to create and evaluate different experimental designs.

Abkürzungen

θ	Rotorlage
U_{ref}	Referenz- bzw. Erregerspannung
U_{sin}	Sinus-Ausgangsspannung Winkelsensor
U_{cos}	Kosinus-Ausgangsspannung Winkelsensor
E	Übersetzungsverhältnis Primär- zu Sekundärwicklung
U_H	Hall-Spannung
R_H	Hallkoeffizient
B	magnetische Flussdichte
I	elektrische Stromstärke
d	Dicke des Hallplättchens
U_{err}	Fehlerspannung
θ_{err}	Rotorlagefehler
Φ	Rotorlage Schätzwert
α	Amplitudendifferenz Resolver-Ausgangsspannung
φ_{Ref}	Winkel Referenzsensor
φ_{DUT}	Winkel zu prüfender Sensor
\bar{E}	Qualitativer Winkelfehler über Periode gemittelt
E_{pp}	Qualitativer Winkelfehler Spitze-Spitze Wert
y	Versuchsplanung Qualitätsmerkmal
x_i	Versuchsplanung Eingangsgrößen
c	Versuchsplanung Modellkonstante
n_f	Versuchsplanung Anzahl der Parameter
n_l	Versuchsplanung Anzahl der Stufen
n_r	Versuchsplanung Anzahl der Versuche
n	Drehzahl
T	Umgebungstemperatur DUT (Box Temperatur)
x	Fehlposition DUT lateral vom Mittelpunkt
y	Fehlposition DUT lateral vom Mittelpunkt
z	Fehlposition DUT axial vom Mittelpunkt
$tilt$	Fehlposition DUT Neigung
U	Versorgungsspannung DUT
n_{Soll}	Drehzahl Sollwert
n_{Ist}	Drehzahl Istwert
n_{Nenn}	Nenn Drehzahl

AMR	Anisotropic Magnetoresistive Effect
ASCII	American Standard Code for Information Interchange
BBD	Box Behnken Design
CBA	Component Based Automation
CC	Conformance Class
CCD	Central Composite Design
COM	Communication Equipment
DLL	Dynamic Link Library
DOE	Design of Experiments
DUT	Device Under Test
EOS	End of Shaft
ERP	Enterprise Resource Planning
FPGA	Field Programmable Gate Array
GMR	Giant Magnetoresistance
ID	Identification
IEC	International Electrotechnical Commission
IO	Input/Output
IRT	Isochronous Real Time
IT	Informationstechnik
KFZ	Kraftfahrzeug
MES	Manufacturing Execution System
PA	Prozessautomatisierung
PC	Personal Computer
PCI	Peripheral Component Interconnect
PLC	Programmable Logic Controller
PN	PROFINET
POSE	Position Sensors
PZD	Prozessdaten
R/D	Resolver to Digital
RT	Real Time
SCADA	Supervisory Control and Data Acquisition
SPS	Speicherprogrammierbare Steuerung
STW	Steuerwort
TCP/IP	Transmission Control Protocol/Internet Protocol
USB	Universal Serial Bus
VI	Virtual Instrument
VR	Variable Reluktanz
ZSW	Zustandswort

1 Einleitung

Elektrische Traktionsantriebe gewinnen in der Automobilindustrie derzeit enorm an Bedeutung [1]. Abb. 1-1 zeigt den schematischen Aufbau eines typischen elektrischen Antriebsstrangs mit dreiphasiger elektrischer Maschine. Für die meisten elektrischen Maschinen hat die genaue Kenntnis der Rotorlage erheblichen Einfluss auf die Qualität und Zuverlässigkeit der Regelung. Der Winkellagesensor ist daher ein wichtiger Bestandteil im elektrischen Antriebsstrang und muss für die jeweilige Anwendung passend gewählt werden.

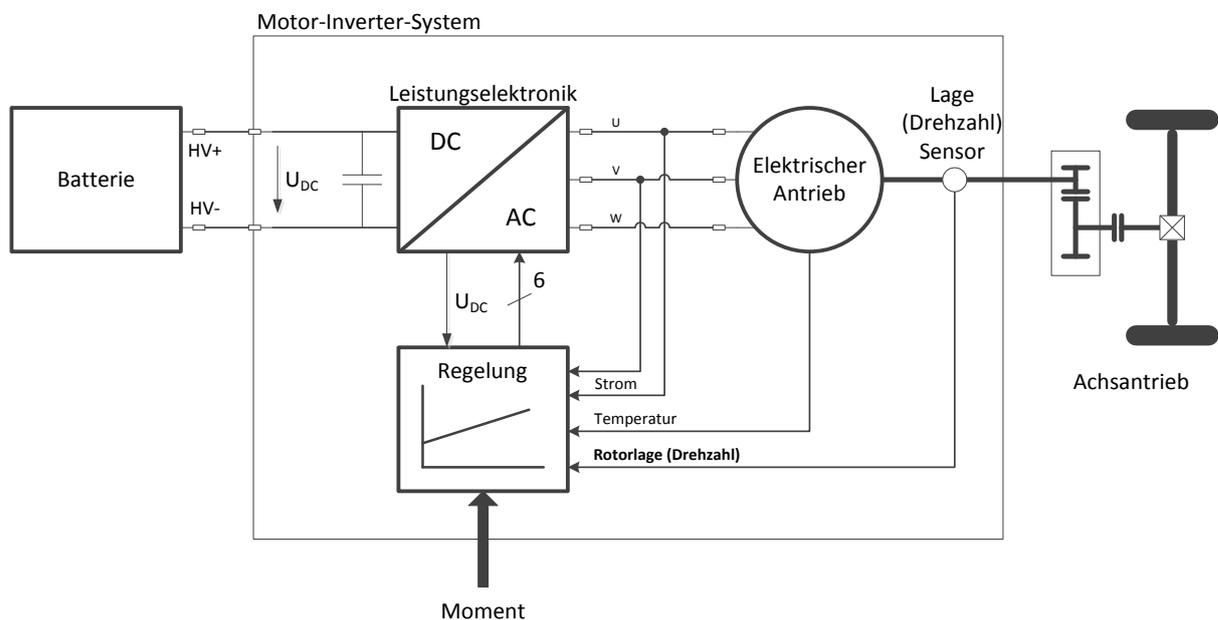


Abb. 1-1: Antriebsstrang eines elektrischen Fahrzeugs [2]

Aufgrund der hohen Leistungsdichte und hohem maximalen Wirkungsgrad werden von der Automobilindustrie momentan bevorzugt permanent-erregte Synchronmaschinen eingesetzt. Diese werden aus praktischen Gründen wie Drehzahl- und Positioniergenauigkeit sowie Betrieb bei optimalem Drehmoment meist feldorientiert geregelt. Dabei wird die exakte Rotorlage neben der Bestimmung der Drehzahl vor allem für die Transformation der Regelgrößen zwischen statorfestem- und rotorflussfestem Koordinatensystem benötigt.

Doch auch andere elektrische Antriebssysteme, wie geschaltete Reluktanzmaschinen oder fremderregte Synchronmaschinen benötigen robuste Positionssensoren mit hoher Genauigkeit sowie hoher Auflösung. Auftretende Sensorfehler bewirken Drehmomentwelligkeit, Effizienzeinbußen, Abnahme des maximalen Drehmoments oder im Extremfall den kompletten Ausfall der Regelung.

Neben der Erfüllung der technischen Anforderungen soll der Sensor aber auch kosteneffektiv sein. Um bereits in frühen Phasen der Entwicklung Entscheidungen treffen zu können, ist es wichtig die in Frage kommenden Sensoren in deren speziellen Anwendungsfeld zu testen. Ein universeller Prüfstand bietet dabei eine optimale Möglichkeit zur Bewertung unterschiedlicher rotierender Positionssensoren anhand reproduzierbarer Testergebnisse [2].

2 Grundlagen

2.1 Rotierende Positionssensoren

Rotierende Positionssensoren oder Winkellagegeber erfassen Drehwinkel oder deren Änderung an einer Welle (z.B. Rotor einer elektrischen Maschine) relativ zu einem feststehenden Teil und setzen dies in ein elektrisches Signal um. Es gibt eine Vielzahl verschiedener Technologien zur Erfassung der Winkellage und auch verschiedene Einteilungsmöglichkeiten.

Eine Möglichkeit ist die grobe Einteilung nach der physikalischen Methode in mechanische, optische, induktive/magnetische und kapazitive Verfahren. Eine weitere Klassifizierung ist anhand der Messung der absoluten Winkellage und deren Zuwachs (inkrementelle Messung) möglich. Bei Sensoren mit absoluter Messmethode steht die Winkellage jederzeit, auch nach Wiedereinschalten der Versorgungsspannung, zur Verfügung. Bei inkrementellen Sensoren wird bei Änderung der Winkellage ein Zähler entsprechend erhöht bzw. verringert. Auch wenn der Zählerstand durch ein nichtflüchtiges Abspeichern gesichert wird, geht die Absolutlage des Winkels nach Abschalten der Versorgungsspannung verloren, da ein mechanisches Ändern der Winkellage im Normalfall auch im abgeschalteten Zustand möglich ist.

Auch nach der Position des Sensors kann unterschieden werden zwischen Montage auf der Welle („through shaft“), teilweiser Montage auf der Welle („partial through shaft“) oder Montage am Ende der Welle („end of shaft EOS“). Der feststehende, meist ablesende bzw. auswertende, Teil des Sensors kann radial oder auch axial zur Welle ausgerichtet sein [2], [3], [4].

Für den Einsatz in einem Fahrzeug kommt aufgrund der Umgebungseinflüsse jedoch nur ein Teil der verfügbaren Sensortechnologien in Frage. Das eingesetzte Sensorsystem soll möglichst robust gegenüber Temperatur, Erschütterungen, mechanischen Toleranzen und auch externen Magnetfeldern sein.

Optische Winkelsensoren sind sehr empfindlich gegenüber Verschmutzungen, im verwendbaren Temperaturbereich limitiert und daher in Fahrzeugen eher ungeeignet. Potentiometersensoren werden zwar oft im automotiven Bereich zur Winkelbestimmung eingesetzt, sind am Rotor einer elektrischen Maschine aufgrund der mechanischen Beanspruchung der Schleifkontakte aber nicht einsetzbar.

Man ist daher bestrebt, kontaktfreie Sensoren zu verwenden, die keinem mechanischen Verschleiß unterliegen und auch den vorig genannten Umgebungseinflüssen trotzen. Diese

Anforderungen erfüllen die induktiven/magnetischen Sensoren am besten [2], [3], [4]. Abb. 2-1 zeigt dazu eine Übersicht verschiedener Messmethoden.

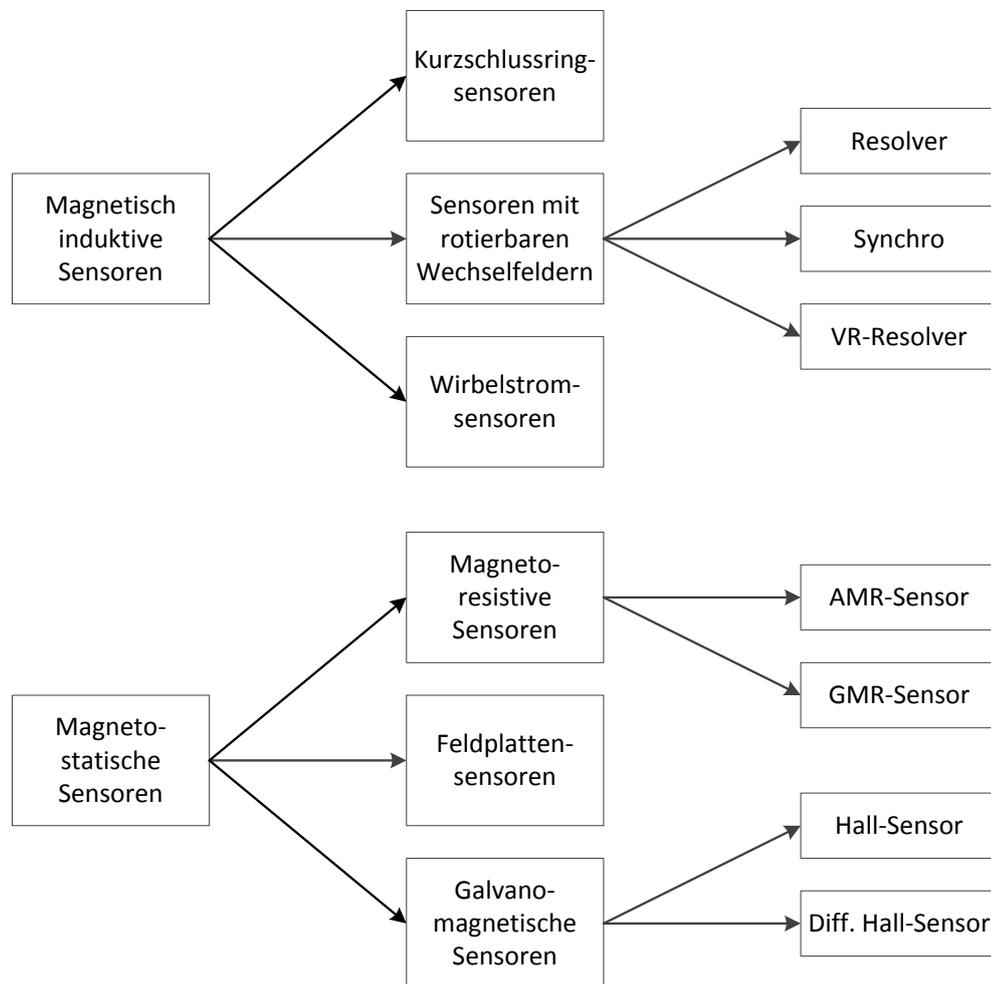


Abb. 2-1: Übersicht induktiv/magnetische Sensoren (VR ... Variable Reluktanz, AMR ... Anisotropic Magnetoresistance, GMR ... Giant Magnetoresistance)

2.1.1 Sensoren mit rotierbaren Wechselfeldern

Resolver

Der Resolver ist ein elektromechanischer Messumformer und kann prinzipiell als rotierender Transformator betrachtet werden. Abb. 2-2 zeigt den grundlegenden Aufbau eines Resolvers. Die Primärspule befindet sich auf dem Rotor und wird über Schleifringe, Bürsten oder berührungslos mittels Drehtransformator nach außen geführt. Die beiden Sekundärwicklungen befinden sich in einem zylindrischen Gehäuse (Stator) und sind um 90° versetzt angeordnet.

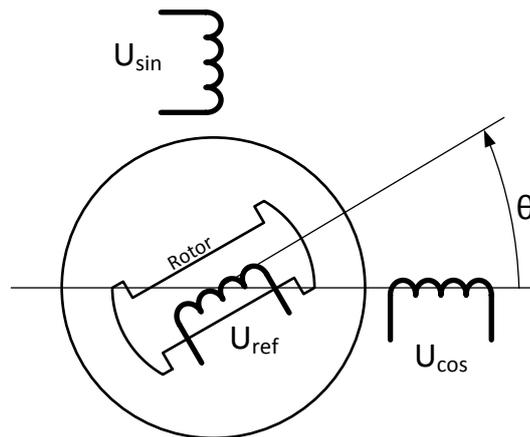


Abb. 2-2: Schematischer Aufbau eines Resolvers [5]

Die Primärwicklung wird mit einer sinusförmigen Referenzspannung gespeist, deren Frequenz einige kHz beträgt. Als Ergebnis erhält man an den Sekundärwicklungen zwei modulierte Ausgangssignale. Diese Ausgangssignale entsprechen der Amplitudenmodulation der Referenzspannung mit Sinus bzw. Kosinus der momentanen Rotorlage θ . Mit der Referenzspannung U_{ref} können die Ausgangssignale U_{sin} und U_{cos} folgendermaßen beschrieben werden [3], [5]:

$$U_{ref} = E \sin(\omega t) \quad (2-1)$$

$$U_{sin} = K E \sin(\omega t) \sin \theta \quad (2-2)$$

$$U_{cos} = K E \sin(\omega t) \cos \theta \quad (2-3)$$

Dabei entspricht E der Amplitude der Referenz- oder auch Erregerspannung und K dem Übersetzungsverhältnis von Primär- zu Sekundärwicklung. Dividiert man nun U_{sin} durch U_{cos} erhält man nach kürzen der Referenzspannung den Tangens der Rotorlage $\tan \theta$ und kann mit dem Arkustangens die Rotorlage bestimmen:

$$\theta = \arctan\left(\frac{U_{sin}}{U_{cos}}\right) = \arctan\left(\frac{\sin \theta}{\cos \theta}\right) \quad (2-4)$$

Für die Auswertung der Resolver-Signale werden sogenannte Resolver-to-Digital Wandler (R/D-Wandler) eingesetzt. Die technische Umsetzung von (2-4) ist jedoch schwierig, da wenn der $\cos \theta$ Null ist, eine Division durch Null stattfindet und auch die Arkustangens-Berechnung ist in einem Embedded System anspruchsvoll. Deshalb verwendet man in der Praxis zur R/D-Wandlung häufig eine sogenannte Tracking Schleife („tracking loop“) [5].

Synchro

Der Synchro ist eine Variante des klassischen Resolvers. Der grundlegende Aufbau und die Funktionsweise sind gleich, anstatt zwei besitzt der Synchro jedoch drei um 120° versetzte Sekundärwicklungen und damit auch drei Ausgangssignale. Mit dieser Konfiguration lässt sich eine höhere Auflösung erzielen [4]. Mit der Referenzspannung U_{ref} (2-1) lassen sich die drei Phasenspannungen U_1 , U_2 und U_3 folgendermaßen beschreiben [3]:

$$U_1 = KE \sin(\omega t) \sin \theta \quad (2-5)$$

$$U_2 = KE \sin(\omega t) \sin(\theta - 120^\circ) \quad (2-6)$$

$$U_3 = KE \sin(\omega t) \sin(\theta - 240^\circ) \quad (2-7)$$

Variable Reluktanz (VR) Resolver

Eine robuste Variante des Resolvers, die sich besonders für den automotiven Einsatz empfiehlt, ist der VR-Resolver. Die von der Erregerspannung gespeiste Primärwicklung befindet sich wie die Sekundärwicklungen am Stator. Der Rotor besteht aus einer elliptisch geformten Scheibe und variiert bei Rotation die Länge des Luftspaltes. Die sinusförmige Änderung des Luftspaltes und dessen Reluktanz im magnetischen Kreis führen zu ebenfalls sinusförmigen Ausgangssignalen. Abb. 2-3 zeigt das Prinzip eines VR-Resolvers mit elliptisch geformter Rotorscheibe. Die Erzeugung der Erregerspannung und die Konditionierung, Demodulation sowie Auswertung der Resolversignale erfolgt üblicherweise in einem integrierten R/D-Chip [2], [4].

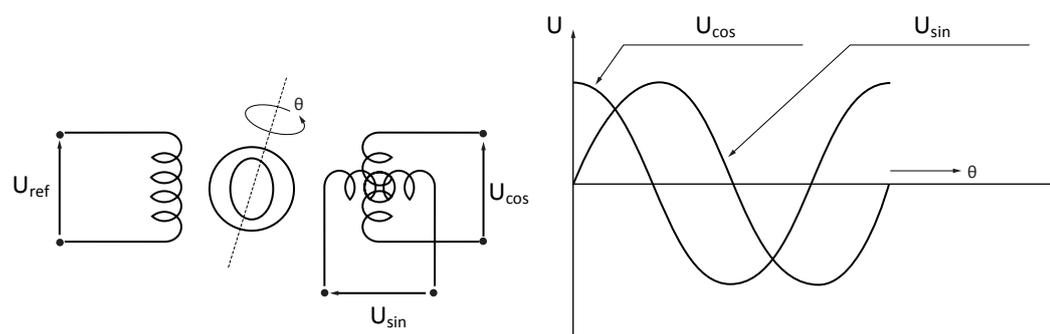


Abb. 2-3: Schematischer Aufbau und Ausgangssignale eines VR-Resolvers [2]

2.1.2 Wirbelstromsensoren

Wird eine elektrisch leitfähige Ebene oder gekrümmte Scheibe (Aluminium, Kupfer, usw.) an eine hochfrequent gespeiste Spule angenähert, so ändert sich deren Wirkwiderstand und Induktivität. Ursache dafür sind Wirbelströme, die in der leitfähigen Scheibe ein Gegenfeld erzeugen. Dieser Effekt kann genutzt werden, um durch eine entsprechende Form dieser Scheibe, auf deren Winkellage zu schließen. Für elektrische Antriebe wird eine dünne sinus- bzw. kosinus-förmige elektrisch leitfähige Schicht auf einer Scheibe am Rotor angebracht, die Sensorelektronik bedeckt nur einen Teil dieser Struktur. Abb. 2-4 zeigt eine Scheibe mit 6-facher Ausführung der Sinusform und axialer Montage des Messgebers. Die Speisung der Spule erfolgt im MHz-Bereich um deren Stromaufnahme klein zu halten. Wirbelstromsensoren sind sehr robust gegenüber elektromagnetischen Störungen [2], [3], [4].

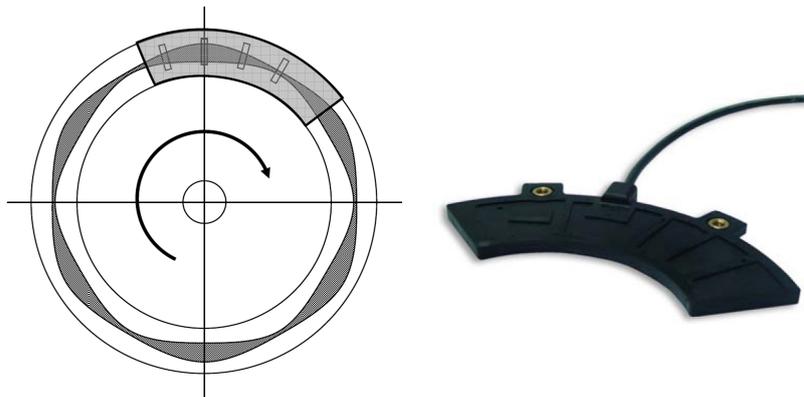


Abb. 2-4: Prinzip Wirbelstromsensor (links) und Sensorelektronik (rechts) [4]

2.1.3 Hall-Sensoren

Hall-Sensoren basieren auf dem gleichnamigen Halleffekt, für dessen Auswertung vor allem dünne Halbleiterplättchen verwendet werden. Wird ein solches Plättchen von einem Strom durchsetzt und wirkt senkrecht zur Stromrichtung ein magnetisches Feld, wird eine Spannungsdifferenz senkrecht zum Strom und zum Feld aufgebaut. Diese Spannung U_H ist proportional zum Strom I und zum Feld B und kann zwischen zwei gegenüberliegenden Punkten quer zur Stromrichtung am Plättchen abgegriffen werden, wobei d die Dicke des Plättchens und R_H den für die Messempfindlichkeit maßgebenden Hallkoeffizient beschreibt [3], [4]:

$$U_H = R_H \cdot I \cdot \frac{B}{d} \quad (2-8)$$

Dieser Effekt kann nun genutzt werden, um die Änderung eines magnetischen Feldes, hervorgerufen beispielsweise durch die Rotation eines Magneten (Abb. 2-5) zu erfassen. Durch die Anordnung von mindestens zwei um 90° versetzter Hallsensoren kann daraus die Rotorlage des Magneten in einem Bereich von 0° bis 360° bestimmt werden [3].

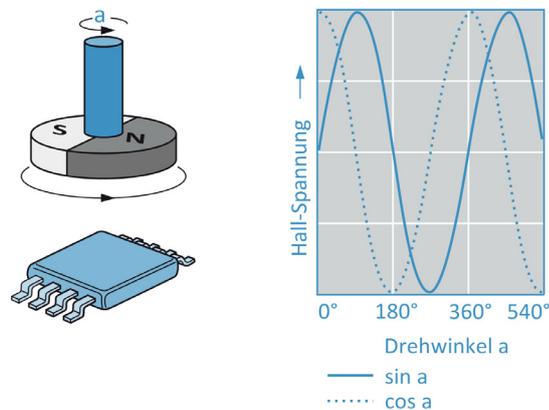


Abb. 2-5: End of Shaft Vierfach-Hallsensor [3]

2.1.4 Magneto-resistive Sensoren

Die grundlegende Eigenschaft der magneto-resistiven Sensoren ist das Detektieren von Amplitude und Richtung eines magnetischen Feldes. Damit kann in derselben Konfiguration wie beim Hallsensor (Abb. 2-5) die Rotorlage einer Welle mit am Ende befindlichen Magneten bestimmt werden. Die beiden meistgenutzten Effekte sind der anisotrope magneto-resistive Effekt (anisotropic magnetoresistance, AMR) und der Riesenmagnetowiderstand (giant magnetoresistance, GMR). Ein Nachteil der AMR-Sensoren ist ihre natürliche Beschränkung des Messbereiches auf 0° bis 180° , GMR-Sensoren dagegen besitzen einen natürlichen Messbereich von 0° bis 360° und auch eine höhere Magnetfeld-Empfindlichkeit bei der Erfassung von Drehzahlen. Diese Vorteile werden durch ein wesentlich komplexeres Schichtsystem von GMR-Sensoren gegenüber den einzelnen Funktionsschichten von AMR-Sensoren ermöglicht. Für beide Technologien besteht der Auswertechip aus zwei Wheatstonebrücken, die die Rotorlage — ähnlich dem Resolver — in Form eines Sinus- und eines Kosinussignals ausgeben [2], [3].

2.2 Sensorfehler

Die zuvor beschriebenen Sensoren unterliegen in ihrem Anwendungsfeld Umwelteinflüssen, wie Temperaturschwankungen, Erschütterungen, magnetischen Feldern aber auch mechanischen sowie elektrischen Fertigungstoleranzen. Diese Einflüsse beeinträchtigen die

Messgenauigkeit und äußern sich je nach eingesetztem Sensortyp sehr unterschiedlich. Stellvertretend für die große Vielfalt an möglichen Sensorfehlern und deren Ursache bei vielen zur Verfügung stehenden Sensortypen, wird exemplarisch der Amplitudenfehler bei Resolvieren etwas genauer betrachtet.

2.2.1 Amplitudenfehler bei Resolvieren

Aufbau und Funktionsweise von Resolvieren sowie die mathematische Beschreibung der Ausgangssignale wurde bereits in Kap. 2.1.1 behandelt. Für die Auswertung mittels R/D-Wandler wird von idealen Resolver-Signalen ausgegangen. In der Praxis bekommt man es allerdings mit nicht idealen Charakteristiken wie Amplitudenfehlern, Quadranten-Abweichungen, induktiven Oberschwingungen und Phasenfehlern zu tun, welche sich natürlich negativ auf das Ausgangssignal, die aktuelle Winkellage, auswirken. Am dominantesten sind dabei die Amplitudenfehler, sie entstehen durch unterschiedliche Wicklungsverhältnisse, unsymmetrischer Referenzspannung oder Fehlern in der Signalkonditionierung mit analogen Komponenten, hervorgerufen von Operationsverstärkern und Filtern [5].

Ideale R/D-Wandlung

Wie bereits in 2.1.1 kurz erwähnt, wird zur Demodulation von Resolver-Signalen oft eine Tracking Schleife eingesetzt. Abb. 2-6 zeigt das Blockschaltbild einer solchen Tracking Schleife wie sie beispielsweise in einem Field Programmable Gate Array (FPGA) implementiert werden kann oder in bereits integrierten R/D-Chips Anwendung findet.

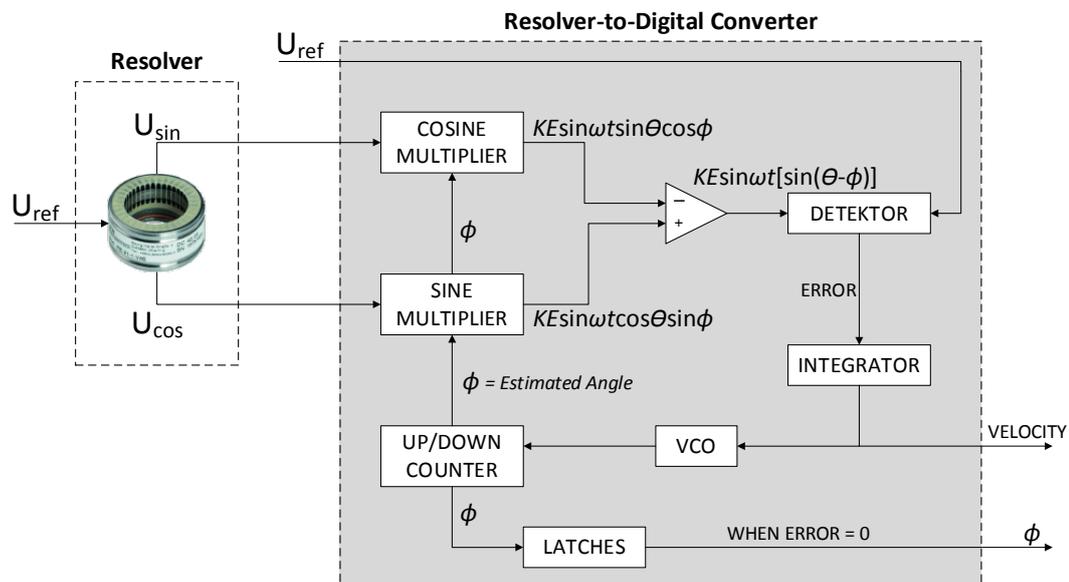


Abb. 2-6: Blockdiagramm Tracking Schleife [5]

Die Tracking-Schleife setzt sich im Wesentlichen aus der Demodulation der Resolver-Signale und aus einer Regelschleife zur Bestimmung der Rotorlage und der Geschwindigkeit zusammen [6].

Die Ausgangssignale des Resolvers U_{sin} (2-2) und U_{cos} (2-3) werden als erster Schritt mit $\cos\Phi$ bzw. $\sin\Phi$ multipliziert. Dabei entspricht Φ der geschätzten Rotorlage. Anschließend wird die Differenz dieser beiden Signale gebildet [5]:

$$U'_{sin-cos} = KE\sin(\omega t)[\sin\theta\cos\Phi - \cos\theta\sin\Phi] \quad (2-9)$$

Mit der folgenden trigonometrischen Beziehung lässt sich (2-9) vereinfachen:

$$\sin\gamma\cos\beta - \cos\gamma\sin\beta = \sin(\gamma - \beta) \quad (2-10)$$

$$U'_{sin-cos} = KE\sin(\omega t)\sin(\theta - \Phi) \quad (2-11)$$

Nach der Demodulation von (2-11) durch den Detektor erhält man die sogenannte Fehlerspannung U_{err} :

$$U_{err} = KE\sin(\theta - \Phi) \quad (2-12)$$

$$\theta_{err} = \theta - \Phi \quad (2-13)$$

Die Tracking-Schleife entspricht im Wesentlichen einem PI-Glied mit zusätzlichem Integrierer, siehe Abb. 2-7. Diese regeln die Fehlerspannung U_{err} zu Null. Mit der Annahme für kleine Winkel (2-14) verschwindet dann ebenfalls der Winkellagefehler θ_{err} und damit folgt der geschätzte Winkel Φ der aktuellen Rotorlage θ .

$$\sin\theta_{err} \approx \theta_{err} \text{ für } |\theta_{err}| \ll 1 \quad (2-14)$$

Zusätzlich liefert die Tracking-Schleife auch noch die Winkelgeschwindigkeit ω und es lassen sich über die beiden Parameter die Bandbreite ω_0 und die Dämpfung δ des PI-Reglers Störungen unterdrücken [6].

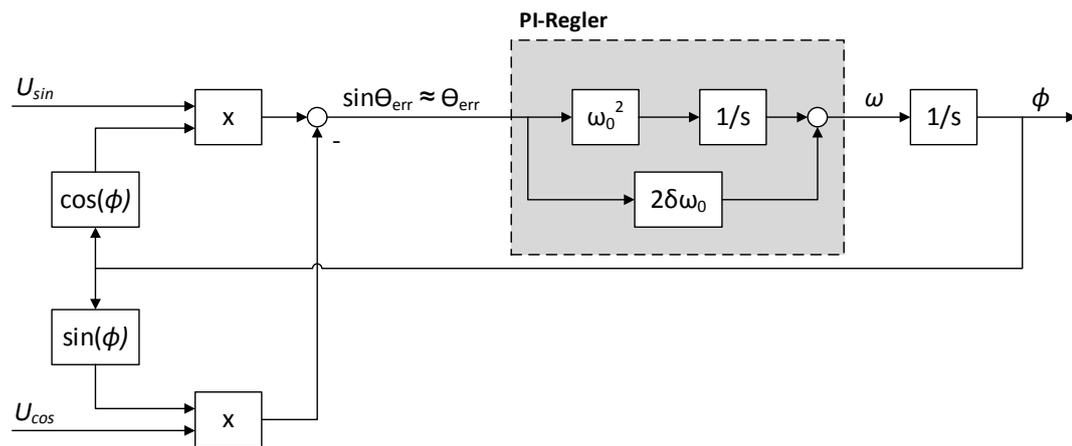


Abb. 2-7: Schematisches Modell des in der Tracking-Schleife enthaltenen Regelkreises [6]

Nicht-Ideale R/D-Wandlung

Es wird nun von einem nicht idealen Resolver mit Amplitudenfehler ausgegangen. Das heißt, dass die Amplituden der beiden Ausgangssignale U_{sin} und U_{cos} ungleich groß sind, dabei entspricht α der Differenz der beiden Amplituden:

$$U_{sin} = KE \sin(\omega t) \sin \theta \quad (2-15)$$

$$U_{cos} = KE(1 + \alpha) \sin(\omega t) \cos \theta \quad (2-16)$$

Die Ausgangssignale des Resolvers U_{sin} (2-15) und U_{cos} (2-16) werden wieder mit $\cos \Phi$ bzw. $\sin \Phi$ multipliziert und daraus die Differenz gebildet:

$$U'_{sin-cos} = KE \sin(\omega t) [\sin \theta \cos \Phi - (1 + \alpha) \cos \theta \sin \Phi] \quad (2-17)$$

Mit der trigonometrische Beziehung (2-10) lässt sich (2-17) folgendermaßen vereinfachen:

$$U'_{sin-cos} = KE \sin(\omega t) [\sin(\theta - \Phi) - \alpha \cos \theta \sin \Phi] \quad (2-18)$$

Der Detektor entfernt durch Demodulation wiederum die Referenzspannung (2-1) aus voriger Gleichung und man erhält die neue Fehlerspannung U_{err} :

$$U_{err} = KE [\sin(\theta - \Phi) - \alpha \cos \theta \sin \Phi] \quad (2-19)$$

Die Tracking-Schleife wird nun wieder U_{err} auf Null regeln, dies führt nun allerdings nicht mehr zum gewünschten Ergebnis $\theta = \Phi$. Setzt man obere Gleichung (2-19) Null lässt sich trotzdem der Winkellagefehler θ_{err} bestimmen:

$$\sin(\theta - \Phi) = \alpha \cos \theta \sin \Phi \quad (2-20)$$

Geht man davon aus, dass man es in der Realität mit einer kleinen Differenz der Resolver-Amplituden α zu tun hat, kann daraus abgeleitet werden, dass der Winkellagefehler θ_{err} ebenfalls klein sein wird und damit $\theta \approx \Phi$ gilt:

$$\sin(\theta_{err}) = \alpha \cos\theta \sin\theta \quad (2-21)$$

$$\cos\theta \sin\theta = \frac{1}{2} \sin(2\theta) \quad (2-22)$$

Mithilfe der Vereinfachung durch Abbruch der Taylor-Reihe (2-14) und der trigonometrischen Beziehung (2-22) erhält man eine mathematische Formulierung für den gesuchten Winkellagefehler:

$$\theta_{err} = \frac{\alpha}{2} \sin(2\theta) \quad (2-23)$$

Wie oben gezeigt wurde, führt eine Differenz der Amplituden von Resolver-Signalen im Winkel-Signal zu einer sinusförmigen Welligkeit, deren Argument der doppelten Rotorlage entspricht [5]. Abb. 2-8 zeigt den ausgewerteten Winkelverlauf eines idealen und eines mit Amplitudenfehler behafteten Resolvers.

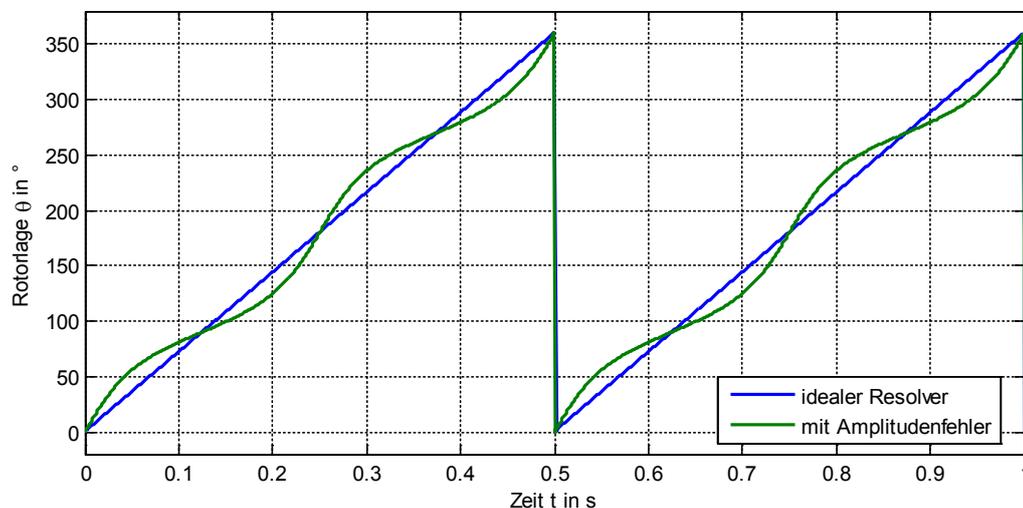


Abb. 2-8: Winkelverlauf Resolver

2.3.2 Messung des Winkelfehlers

Durch den Vergleich der Messergebnisse des zu prüfenden Sensors (Device Under Test DUT) mit einem Referenzsensor, kann der Winkelfehler bestimmt und evaluiert werden. Da die Bewertung anhand der typischen Rampensignale von Winkel als Funktion der Zeit schwierig

ist, wird ein qualitativer Faktor \bar{E} des Winkelfehlers berechnet. Dieser stellt das Integral der absoluten Winkeldifferenz aus Referenzsensor φ_{Ref} und dem zu prüfenden Sensor φ_{DUT} über eine Periode dar [2]:

$$\bar{E} = \int_t^{t+T} |\varphi_{Ref} - \varphi_{DUT}| dt \quad (2-24)$$

Diese Vorgangsweise bringt den Vorteil, dass die Qualität eines Sensorsignals mit einem einzelnen Wert angegeben werden kann. Eine weitere Möglichkeit zur Bewertung des zu prüfenden Sensors, ist die Berechnung des Spitze-Spitze-Werts E_{pp} (peak-peak) aus Maximum und Minimum des Winkelfehlers:

$$E_{pp} = \left| \max(\varphi_{Ref} - \varphi_{DUT}) - \min(\varphi_{Ref} - \varphi_{DUT}) \right| \quad (2-25)$$

Abb. 2-9 zeigt die Messung eines stark fehlpositionierten AMR-Sensors. Die Sinus-Kosinussignale weichen deutlich von einer idealen Sinusform ab und auch die Abweichung des gemessenen Winkels vom idealen rampenförmigen Winkelverlauf ist zu erkennen.

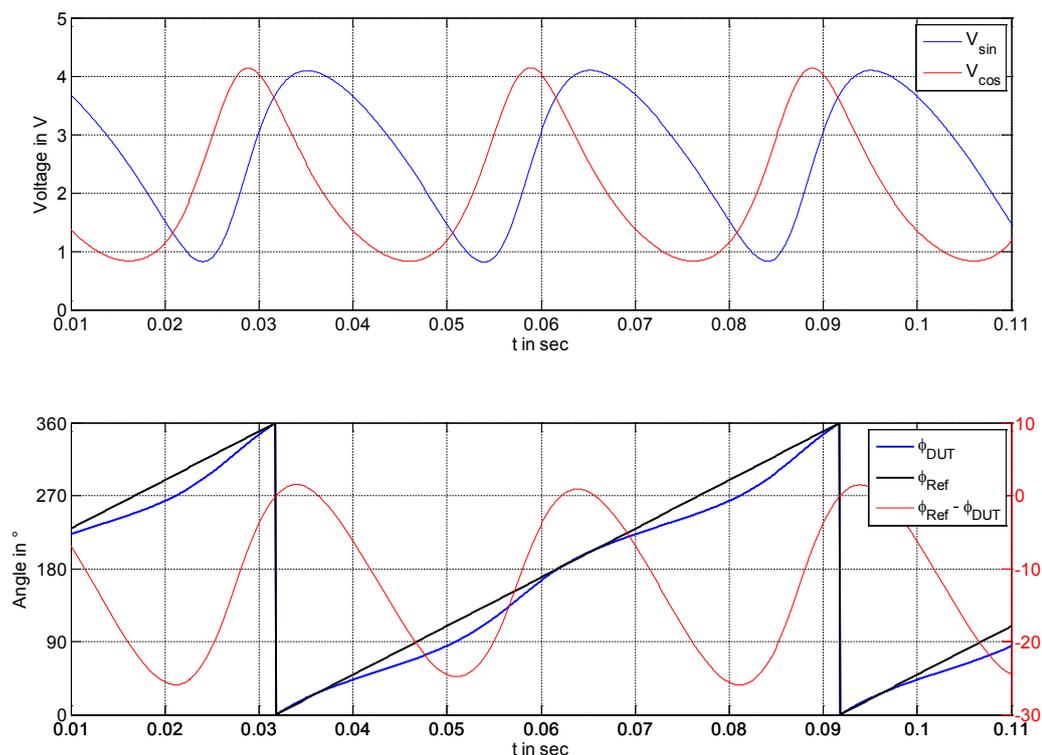


Abb. 2-9: Fehlpositionierter AMR-Sensor: Resolver-Signale (oben) und berechneter Winkelverlauf (unten) [2]

2.3 Statistische Versuchsplanung

Die statistische Versuchsplanung (Design of Experiments DOE) ist eine Methode zur effizienten Planung und Auswertung von Versuchsreihen. Ziel ist die Beschreibung sowie Bewertung des Fehlers rotierender Positionssensoren anhand eines Modells. Das Erstellen eines analytischen Modells ist in der Praxis oft sehr schwierig, in vielen Fällen ist es aber ausreichend die internen Zusammenhänge eines Systems zu quantifizieren. Dies ist etwa mit dem linearen Beschreibungsmodell möglich (2-26). Dieses liefert für jede Kombination der Eingangsgrößen x_i einen Näherungswert des Qualitätsmerkmals y . c_i sind die Modellkonstanten für die Haupteffekte und c_{ij} die Modellkonstanten für die Wechselwirkungen zwischen den Eingangsgrößen. Eine weitere Konstante c_0 beschreibt den Mittelwert. Der Modellfehler ε beschreibt die Qualität des erstellten Modells und ist bei einem passenden Modell klein im Vergleich zur Variation des Qualitätsmerkmals y [7]:

$$y = c_0 + \sum_{i=1}^{n_f} c_i x_i + \sum_{i=1}^{n_f-1} \sum_{j=i+1}^{n_f} c_{ij} x_i x_j + \varepsilon \quad (2-26)$$

Bei nichtlinearen Zusammenhängen zwischen den Eingangsgrößen und des Qualitätsmerkmals arbeitet das lineare Modell ungenau. Das quadratische Beschreibungsmodell erweitert das lineare um einen Term für quadratische Effekte und erzielt dadurch bessere Ergebnisse:

$$y = c_0 + \sum_{i=1}^{n_f} c_i x_i + \sum_{i=1}^{n_f} c_{ii} x_i^2 + \sum_{i=1}^{n_f-1} \sum_{j=i+1}^{n_f} c_{ij} x_i x_j + \varepsilon \quad (2-27)$$

Für die Bestimmung der Modellparameter sind entsprechende Versuche notwendig. Dazu stehen verschiedene Eingangsgrößen (Fehlpositionierung, Drehzahl, usw.) — die sogenannten Parameter — zur Verfügung. n_f bezeichnet die Anzahl der verschiedenen Parameter eines Versuchs. Jeder Parameter kann, abhängig vom zu testenden Sensor, in gewissen Bereichen beliebig variiert werden. Für die Versuchsplanung erfolgt die Einstellung der Parameter in sogenannten Stufen (Levels), wobei jeder Parameter auf mindestens 3 unterschiedlichen Stufen getestet wird [7].

2.3.1 Vollfaktorieller Versuchsplan

Bei einem vollfaktoriellen Versuchsplan (full factor) werden alle möglichen Kombinationen getestet. Für einen Versuchsplan mit n_f Parametern und n_l Stufen erhält man mit folgender Bedingung die Anzahl der Versuche n_r :

$$n_r = n_l^{n_f} \quad (2-28)$$

Für ein System mit drei verschiedenen Parametern (A, B, C) und jeweils zwei unterschiedlichen Stufen (gekennzeichnet mit + und –) ergibt sich folgender vollfaktorieller Versuchsplan mit acht Versuchen (y_1 bis y_8):

Tabelle 2-1: Vollfaktorieller Versuchsplan für 3 Parameter [7]

A	B	C	y
–	–	–	y_1
+	–	–	y_2
–	+	–	y_3
+	+	–	y_4
–	–	+	y_5
+	–	+	y_6
–	+	+	y_7
+	+	+	y_8

Bei einem System mit 6 Parametern und 3 Stufen beträgt die Größe des vollfaktoriellen Versuchsplans bereits 729 Versuche. Oft ist der Zeit- und Ressourcenaufwand für solch große Versuchspläne nicht gegeben. Die statistische Versuchsplanung bietet effektive Möglichkeiten zur Verringerung des Versuchsaufwands, damit viele Faktoren sowie nichtlineare Zusammenhänge untersucht werden können. Für ein quadratisches Beschreibungsmodell, welches für nichtlineare Zusammenhänge gut geeignet ist, stehen für den Entwurf von effektiven Versuchsplänen beispielsweise das Central-Composite-Design oder das Box-Behnken-Design zur Verfügung [7].

2.3.2 Central-Composite Design

Das Central-Composite-Design (CCD) ist ein zweiteiliger Versuchsplan, der auf einem zweistufigen Versuchsplan aufbaut. Der erste, zweistufige Teil des Versuchsplans entspricht einem „Würfel“, der zweite Teil einem „Stern“. Abb. 2-10 zeigt die graphische Darstellung eines CCD mit drei Parametern. Bei vier oder mehr Parametern erweitert sich die Form des Plans um die entsprechende Anzahl Dimensionen, der Würfel wird zu einem sogenannten Hyperwürfel. Die durchzuführenden Versuche entsprechen den Ecken des Würfels.

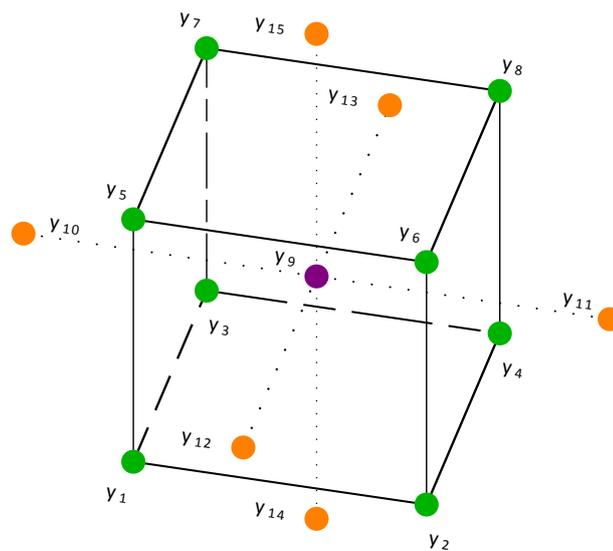


Abb. 2-10: Central-Composite-Design für 3 verschiedene Parameter [7]

Der „Stern“ entsteht durch die Variation der einzelnen Faktoren, ausgehend von der Mittelstellung (Center). Die Abstände vom Mittelpunkt überragen die Seiten des Würfels, man erhält fünf verschiedene Stufen für jeden Parameter. Tabelle 2-2 zeigt den Versuchsplan eines Central-Composite-Designs. Für 3 verschiedene Parameter ergibt sich ein Feld mit 15 Versuchen [7].

Tabelle 2-2: Versuchsplan Central-Composite-Design für 3 verschiedene Parameter [7]

A	B	C	y
—	—	—	y_1
+	—	—	y_2
—	+	—	y_3
+	+	—	y_4
—	—	+	y_5
+	—	+	y_6
—	+	+	y_7
+	+	+	y_8
0	0	0	y_9
— —	0	0	y_{10}
++	0	0	y_{11}
0	— —	0	y_{12}
0	++	0	y_{13}
0	0	— —	y_{14}
0	0	++	y_{15}

Das Hinausragen des Sterns über den Würfel kann in der Praxis ein Problem darstellen, wenn die vom Versuchsplan geforderten Einstellungen nicht realisierbar sind. In diesem Fall kann

das Face-Centered CCD (Abb. 2-11) verwendet werden, bei dem der Stufenabstand des Sterns auf den des Würfels reduziert wird. Die Eigenschaften des Versuchsplans werden allerdings im Vergleich zum regulären CCD verschlechtert [7].

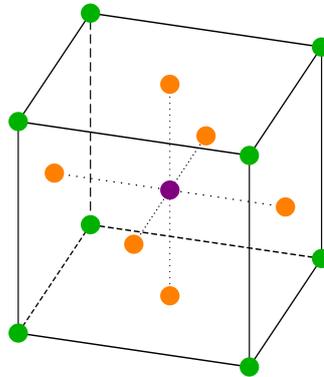


Abb. 2-11: Face-Centered CCD für 3 verschiedene Parameter [7]

2.3.3 Box-Behnken-Design

Der Faktorraum des Box-Behnken-Design (BBD) ergibt ebenfalls einen Würfel (Abb. 2-12), hier liegen die Versuche jedoch, im Gegensatz zum CCD, auf den Kanten. Dieses Entwurfsverfahren für Versuchspläne bietet sich an, wenn die Ecken des Parameterraumes kritisch sind. Es muss allerdings beachtet werden, dass die Beschreibungsfunktion dann auch nur im mittleren Bereich ihr Optimum erreicht und für die Eckbereiche nicht geeignet ist [7].

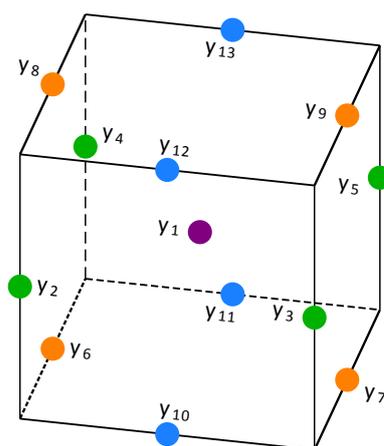


Abb. 2-12: Box-Behnken-Design für 3 verschiedene Parameter [7]

Der zugehörige Versuchsplan (Tabelle 2-3) zeigt den Aufbau des Box-Behnken-Designs für 3 verschiedene Parameter und jeweils 3 Stufen. Es werden jeweils zwei Faktoren in allen

Kombinationen auf zwei Stufen variiert, die übrigen Faktoren bleiben auf der mittleren Stufe („0“). Es ergibt sich ein ringförmiger Aufbau. Für 3 Parameter genügt ein Testumfang von 13 Versuchen.

Tabelle 2-3: Versuchsplan Box-Behnken-Design für 3 verschiedene Parameter [7]

<i>A</i>	<i>B</i>	<i>C</i>	<i>y</i>
0	0	0	y_1
–	–	0	y_2
+	–	0	y_3
–	+	0	y_4
+	+	0	y_5
–	0	–	y_6
+	0	–	y_7
–	0	+	y_8
+	0	+	y_9
0	–	–	y_{10}
0	+	–	y_{11}
0	–	+	y_{12}
0	+	+	y_{13}

3 Prüfstand

3.1 Anforderungen

Der Drehzahlbereich des Sensors ist abhängig von der automotiven Anwendung und dem jeweiligen Typ des Antriebsstrangs. Die maximale Drehzahl orientiert sich, um das gesamte Spektrum abzudecken, an den Anwendungen mit der höchsten vorkommenden Drehzahl für elektrische sowie auch hybride Antriebssysteme von etwa 24000 rpm. Die Umgebungstemperatur ist neben Vibrationen und Schwingungen der wesentliche Umwelteinfluss auf den Sensor und kann je nach Position und Bauform des Sensors zeitweise bis zu 180 °C erreichen. Ein sehr wichtiges Testkriterium ist die Fehlpositionierung des Sensors, dieser soll dreidimensional positionierbar und in eine Drehrichtung schwenkbar (tilt) sein. Die Robustheit des Sensors gegenüber Abweichungen der Versorgungsspannung vom Sollwert soll ebenfalls bewertet werden. Tabelle 3-1 zeigt die Spezifikationen des Prüfstands und die Realisierung des jeweiligen Testparameters [2].

Tabelle 3-1: Spezifikationen des Prüfstands [2]

Parameter	Min.	Max.	Toleranz	Einheit	Realisierung
Drehzahl (n)	-24000	+24000	< 1%	rpm	Elektrisches Antriebssystem
Umgebungstemperatur DUT (T)	-40	+180	< 1	°C	Temperatursystem
Fehlpositionen, lateral (x, y vom Mittelpunkt)	-15	+15	0,01	mm	Positioniersystem
Fehlposition, axial (z)	0	+15	0,01	mm	Positioniersystem
Neigung (tilt)	-45	+45	< 10 ⁻⁵	° mech	Positioniersystem
Versorgungsspannung DUT (U)	0	+40	0,01	V	Spannungsversorgung

Eine zentrale Mess- und Visualisierungssoftware soll einen gemeinsamen Zugriffspunkt für alle Testparameter bilden. Diese sollen vom Benutzer in ihrem spezifizierten Bereich beliebig vorgegeben und auch überwacht werden können. Zudem soll es möglich sein verschiedene Prüfzenarien anhand eines automatisierten Prüfablaufs durchzuführen. Die Modellierung des Sensorfehlers erfolgt mit Hilfe der statistischen Versuchsplanung, anhand dieser sollen die Testparameter in vorgegebenen Bereichen automatisch variiert und die Testergebnisse aufgezeichnet werden. In der weiteren Arbeit wird daher zusammenfassend von einer Automatisierung des Prüfstandes gesprochen.

3.2 Aufbau

Abb. 3-1 zeigt den schematischen Aufbau des Prüfstands und seiner Komponenten. Der mechanische Aufbau (Abb. 3-2) besteht im Wesentlichen aus einer drehbar gelagerten Welle. Auf ihr befinden sich der Referenzsensor und der rotierende Teil des zu prüfenden Sensors (DUT). Angetrieben wird die Welle über eine Wellenbalgkupplung von einem elektrischen Motor. Das Messobjekt befindet sich in einer Thermobox, die mit entsprechend isolierten Zuleitungen mit dem Temperaturschrank verbunden ist. Der feststehende Teil des Sensors ist über eine Halterung mit dem Positioniersystem verbunden, welches sich unter der Thermobox befindet.

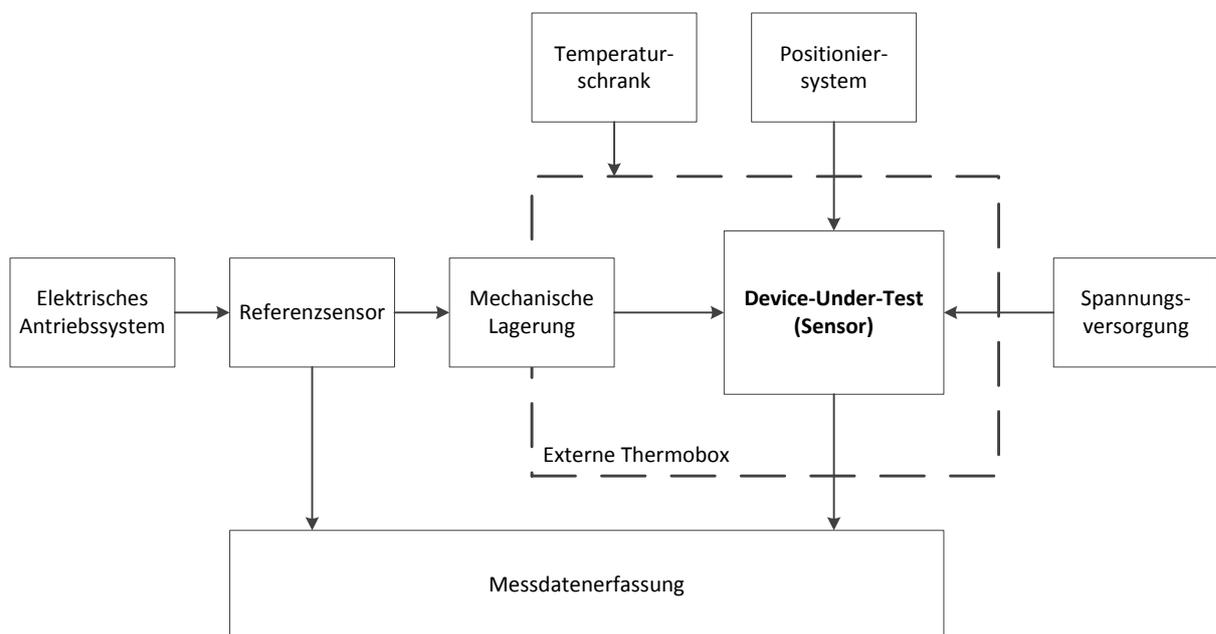


Abb. 3-1: Prüfstand Übersicht [2]

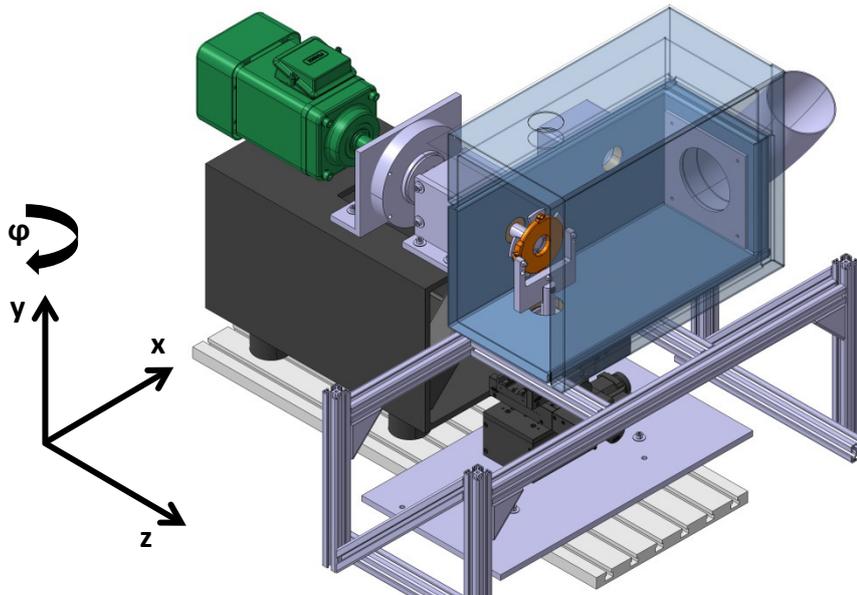


Abb. 3-2: CAD-Modell des Prüfstands mit Koordinatensystem [8]

3.3 Komponenten

Auf die mechanischen Komponenten des Prüfstands (siehe [8]) wird in dieser Arbeit nicht weiter eingegangen. Für diese Arbeit relevant sind lediglich die Komponenten, die zur Einstellung der Testparameter (Tabelle 3-1) verwendet werden bzw. in die Mess- und Visualisierungssoftware eingebunden werden. Dazu zählen der elektrische Antrieb, das Positioniersystem, das Temperatursystem, die Spannungsversorgung des Sensors und die Messdatenerfassung.

Elektrischer Antrieb

Der elektrische Antrieb besteht im Wesentlichen aus einem Elektromotor und einem Antriebssystem, welches aus Umrichter und einer Regel- und Kommunikationseinheit besteht (Abb. 3-3).

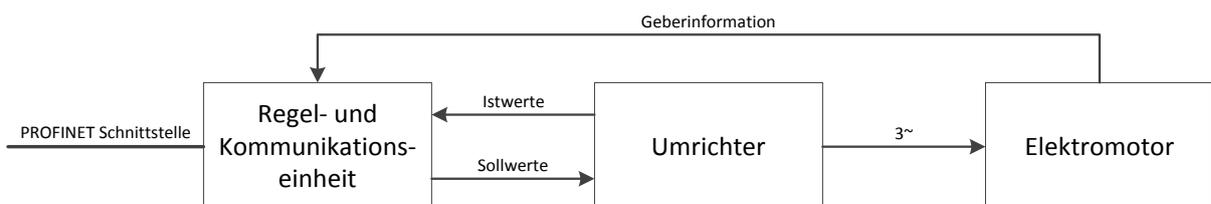


Abb. 3-3: Übersicht elektrischer Antrieb

Die Regel- und Kommunikationseinheit besitzt eine Ethernet-Schnittstelle für die Inbetriebnahme und Konfiguration des gesamten Antriebssystems. Diese wurde mithilfe der Software STARTER von Siemens durchgeführt. Für den weiteren Betrieb bzw. für Steuer- und Regelungsaufgaben steht zur Kommunikation eine PROFINET Schnittstelle zur Verfügung.

Positioniersystem

Für die Positionierung des Sensors sowie auch für die Evaluierung von Fehlpositionierungen wurde eine Kombination aus vier mechanischen Verstelltischen — zwei Lineartische, ein Höhenverstelltisch sowie ein Drehtisch — der Firma OWIS verwendet. Für die Automatisierung des Prüfstandes wurden diese durch die jeweils äquivalente, motorisierte Variante ersetzt. Bei deren Motorisierung kann zwischen 2-Phasen-Schrittmotor und DC-Servomotor gewählt werden. Da für uns beim Verfahren der Positioniersysteme deren Dynamik keine Rolle spielt, wurde die wesentlich günstigere Variante mit 2-Phasen-Schrittmotor gewählt. Zur Ansteuerung wird pro Achse je eine Positioniersteuerung verwendet, zur Kommunikation steht eine USB-Schnittstelle zur Verfügung.



Abb. 3-4: Übersicht Positioniersystem

Temperatursystem

Die Umgebungstemperatur des Sensors wird mit dem Temperaturprüfschrank Espec ARG-0220 der Firma ThermoTEC und einer externen Thermobox variiert. Steuerung und Kommunikation mit dem Temperaturschrank erfolgt durch ein zusätzliches Modul mit RS-485-Schnittstelle.

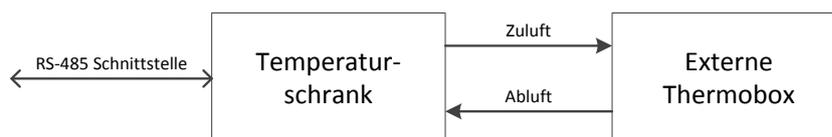


Abb. 3-5: Übersicht Temperatursystem

Spannungsversorgung

Die Versorgungsspannung des Sensors ist ebenfalls ein zu bewertender Parameter. Für diese Aufgabe wird das programmierbare Labornetzteil Voltcraft PSP 1405 verwendet. Dieses besitzt einen Ausgangsspannungsbereich von 0 bis 40 V Gleichspannung bei einer maximalen Restwelligkeit von 10 mV. Der maximale Ausgangsstrom beträgt 5 A und für die Kommunikation besitzt das Netzteil eine serielle Schnittstelle vom Typ RS-232.

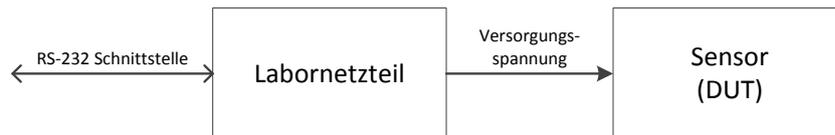


Abb. 3-6: Übersicht Spannungsversorgung

Messdatenerfassung

Die Aufzeichnung der Messdaten erfolgt mit der Messkarte Orion 0816-5M0 + Anschlussbox der Firma Dewetron (Abb. 3-7). Die 8 analogen Eingänge der Karte besitzen eine Auflösung von 16 Bit und zeichnen simultan Daten mit einer Abtastrate von 5 MS/s auf. Die Messkarte befindet sich via PCI-Slot im Prüfstandsrechner und wird mit der Software DEWESoft bedient. Zur externen Bedienung steht das TestBed Plugin zur Verfügung, mit dessen Hilfe man über eine RS-232- oder Ethernet-Schnittstelle Zugriff auf die grundlegenden Funktionen von DEWESoft erhält.

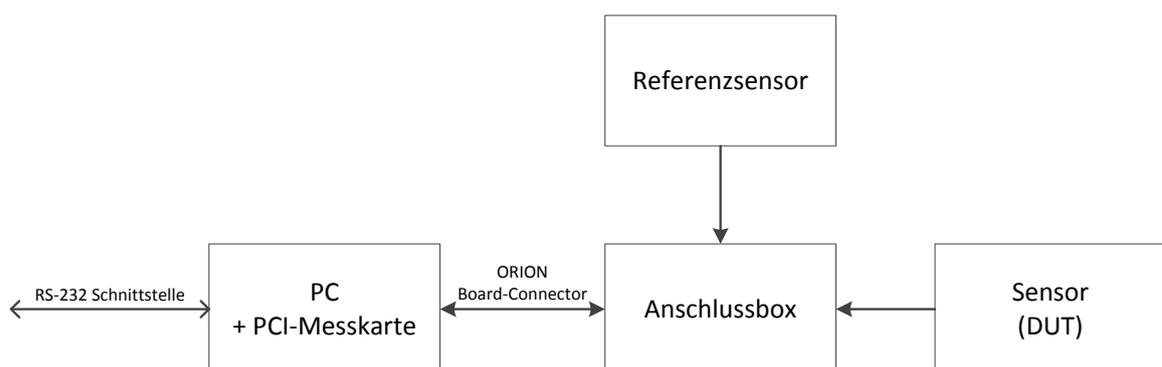


Abb. 3-7: Übersicht Messdatenerfassung

3.4 Konzept Automatisierung

Die Arbeit begann damit, ein geeignetes Konzept für die gemeinsame Steuerung und Visualisierung der einzelnen Komponenten zu finden. Der Prüfstand war in seiner Grundform

vorhanden, es konnten bereits Messungen durchgeführt werden. Die Bedienung der Komponenten geschah jedoch händisch, direkt am Bedienterminal bzw. mit separater Hersteller-Software und war damit unkomfortabel, zeitaufwendig und auch fehleranfällig.

3.4.1 Konzept LabVIEW

Den Mittelpunkt dieses Konzepts bildet die Software LabVIEW, die zum Steuern und auch zur Visualisierung der Prüfstandskomponenten eingesetzt wird. Für die Programmierung und Ausführung von LabVIEW wird der PC der Messdatenerfassung verwendet. Dieser muss ebenfalls direkt mit den Komponenten verbunden werden.

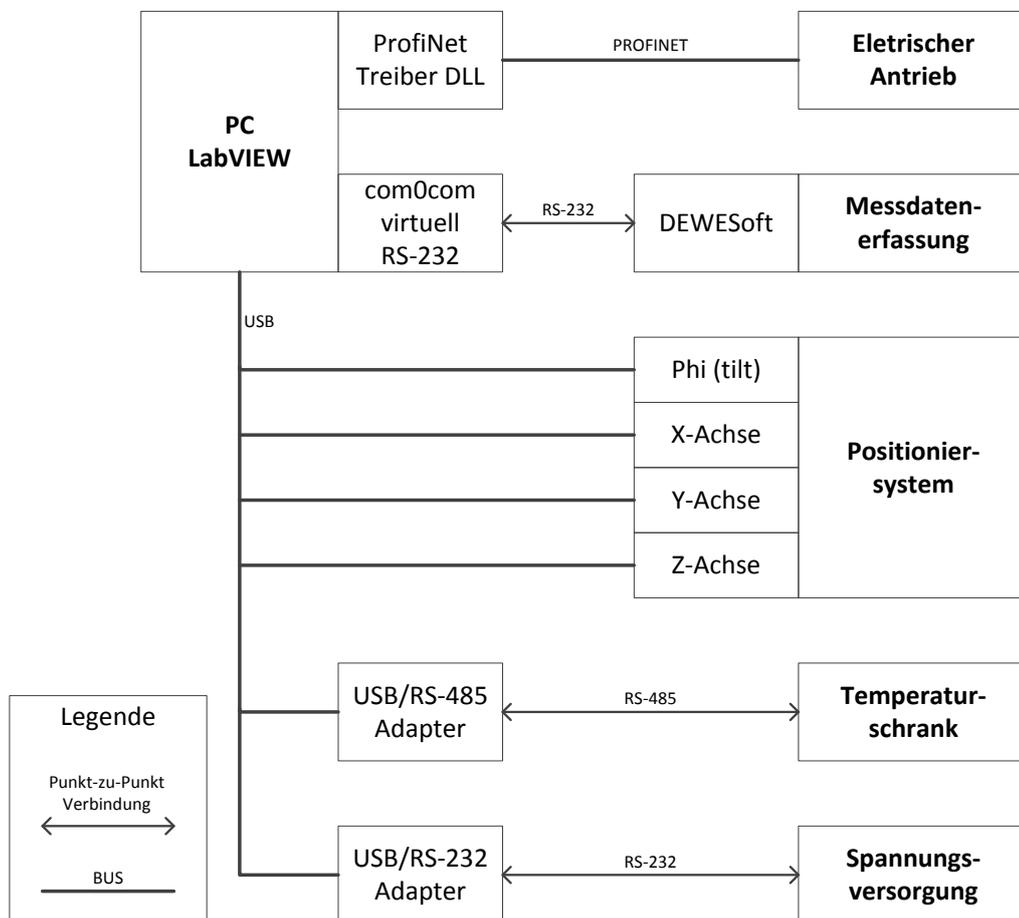


Abb. 3-8: Übersicht Konzept LabVIEW

Die Ansteuerung des elektrischen Antriebs geschieht mit einem PROFINET Treiber. Dieser wird als DLL in LabVIEW eingebunden und stellt Funktionen zur Kommunikation ohne Echtzeitanforderungen über eine Standard Ethernet-Schnittstelle zur Verfügung. Die Verbindung mit den Steuerungen des Positioniersystems erfolgt via USB-Schnittstelle. Auch hier steht zur Kommunikation mit der Hardware eine Programmbibliothek (DLL) des

Herstellers zur Verfügung. Temperaturschrank und Spannungsversorgung werden über die seriellen Schnittstellen RS-485 bzw. RS-232 mit entsprechenden USB-Adaptern und der LabVIEW-Bibliothek NI-VISA angebunden. Da sich die Steuerungs-Software LabVIEW und auch die Messdatenerfassungs-Software DEWESoft auf einem PC befinden, reicht es aus die hardwaremäßige Verbindung zu simulieren. Mit der freien Software com0com werden zwei virtuelle, serielle Anschlüsse erstellt und miteinander verbunden. Eine genaue Beschreibung folgt in Kap. 4 Anbindung der Komponenten.

3.4.2 Alternatives Konzept

Eine Alternative zur Verwendung von LabVIEW für die Automatisierung des Prüfstands ist der Einsatz einer speicherprogrammierbaren Steuerung (SPS). Deren Echtzeitfähigkeit bietet in Bezug auf die Dynamik des elektrischen Antriebs wesentlich mehr Möglichkeiten. Für die Anbindung der restlichen Komponenten ist jedoch – mit der Erweiterung der SPS um die jeweiligen Schnittstellenmodule – ein höherer Aufwand zu erwarten. Für das Erstellen einer Mess- und Visualisierungssoftware gibt es beispielsweise mit WinCC ein PC-basiertes Prozessvisualisierungsprogramm für Steuerungen der Firma Siemens. Neben dem zusätzlichen Zeitaufwand für die Einarbeitung in die SPS-Programmierung waren die zusätzlichen Ausgaben für Software, Schnittstellenmodule und die SPS selbst ausschlaggebend für die Wahl des Automatisierungskonzeptes mit LabVIEW.

4 Anbindung der Komponenten

4.1 Elektrischer Antrieb

Basis für die Kommunikation zwischen dem LabVIEW-PC und dem elektrischen Antrieb ist der industrielle Ethernet-Feldbus PROFINET (Kap. 4.1.1). Das Geräteprofil PROFIdrive (Kap. 4.1.3) stellt die benötigten Protokolle zum Datenaustausch zur Verfügung. Abb. 4-1 zeigt den schematischen Aufbau eines Datenpakets zur Kommunikation mit dem elektrischen Antrieb, bestehend aus der Übertragungstechnologie Ethernet und den Protokollen PROFINET und PROFIdrive.



Abb. 4-1: Schematischer Aufbau — Kommunikation mit dem elektrischen Antrieb

4.1.1 PROFINET

PROFINET (Processing Field Network) ist ein offener Industrial Ethernet Standard (IEC 61158 und IEC 61784) und nutzt dabei bereits vorhandene TCP/IP- und IT-Standards. Anwendung findet PROFINET in der Fertigungs- und Prozessautomatisierung. Hauptgründe für die Verwendung von echtzeitfähigem Ethernet für Feldbusse sind die Verwendung bereits vorhandener und günstiger Ethernet Hardware und ein einheitliches Rechnernetz von der Leitebene im Bürobereich bis zur Feldebene der Industrieanlagen [9]. Abb. 4-2 zeigt eine Einteilung der industriellen Automatisierung.

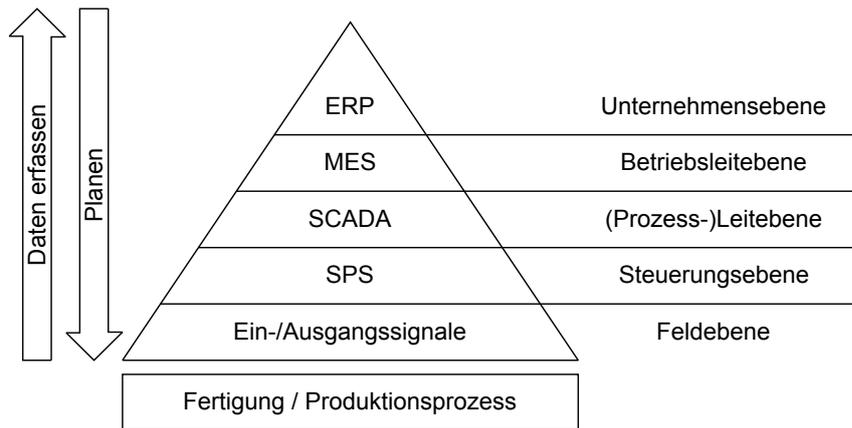


Abb. 4-2: Automatisierungspyramide [10]

PROFINET besitzt ein modulares Konzept und findet Anwendung in verschiedenen Varianten. Während PROFINET CBA (Component Based Automation) für die Kommunikation zwischen mehreren an der Automatisierung beteiligten, verteilten Komponenten verwendbar ist, besitzt PROFINET IO die größte praktische Relevanz. PROFINET IO ist maßgeschneidert für den schnellen Austausch von Daten zwischen dezentralen Feldgeräten und Steuerungen und folgt dabei einem Erzeuger-Verbraucher-System. Das Grundsystem (Abb. 4-3) besteht aus einem sogenannten IO-Controller, typischerweise eine speicherprogrammierbare Steuerung, und mindestens einem dezentral angeordneten Feldgerät (IO-Device). Optional kann das Grundsystem auch noch einen sogenannten IO-Supervisor (Programmiergerät, PC, Bedienpanel) zur Inbetriebnahme, Fehlersuche, etc. beinhalten [9], [11].

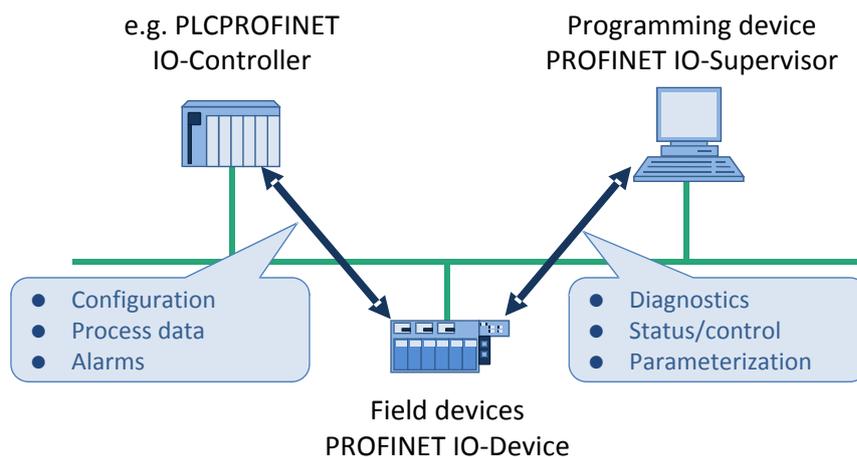


Abb. 4-3: Grundsystem PROFINET IO mit Kommunikationspfaden [9]

Der Funktionsumfang von PROFINET ist kaskadierbar, da je nach Anwendung nicht immer die komplette Funktionalität benötigt wird. Je nach Mindestanforderung stehen drei aufeinander aufbauende Konformitätsklassen (Conformance Class CC) zur Verfügung (Abb.

4-4). Neben der Applikation werden Vorgaben bezüglich der eingesetzten Hardware, Übertragungsmedium und dem Redundanzverhalten getroffen.

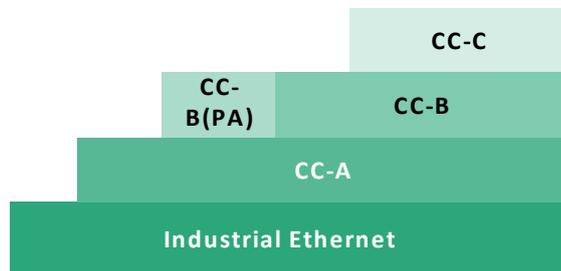


Abb. 4-4: Aufbau der Konformitätsklassen [9]

Die Klasse CC-A ermöglicht die Grundfunktionen für PROFINET IO mit Echtzeitkommunikation (Real-Time RT), es können alle IT-Services uneingeschränkt genutzt werden. Datenübertragung ist leitungsgebunden (100 Base TX/FX) sowie auch drahtlos möglich. Klasse CC-B erweitert dieses Konzept um Netzwerkd Diagnose über IT-Mechanismen sowie Topologieinformationen, wofür aber auch entsprechend kompatible Geräte (Switch, etc.) benötigt werden. Eine Funktion zur Systemredundanz ist in der Klasse CC-B (PA) enthalten. Basis für takt synchrone Applikationen mit isochroner Echtzeit (Isochronous Real-Time IRT) bildet die Klasse CC-C, benötigt dann aber auch Geräte mit hardwareunterstützter Bandbreitenreservierung [9], [11]. Abb. 4-5 zeigt eine Übersicht mit typischen Anwendungen und dem Echtzeitverhalten von PROFINET.

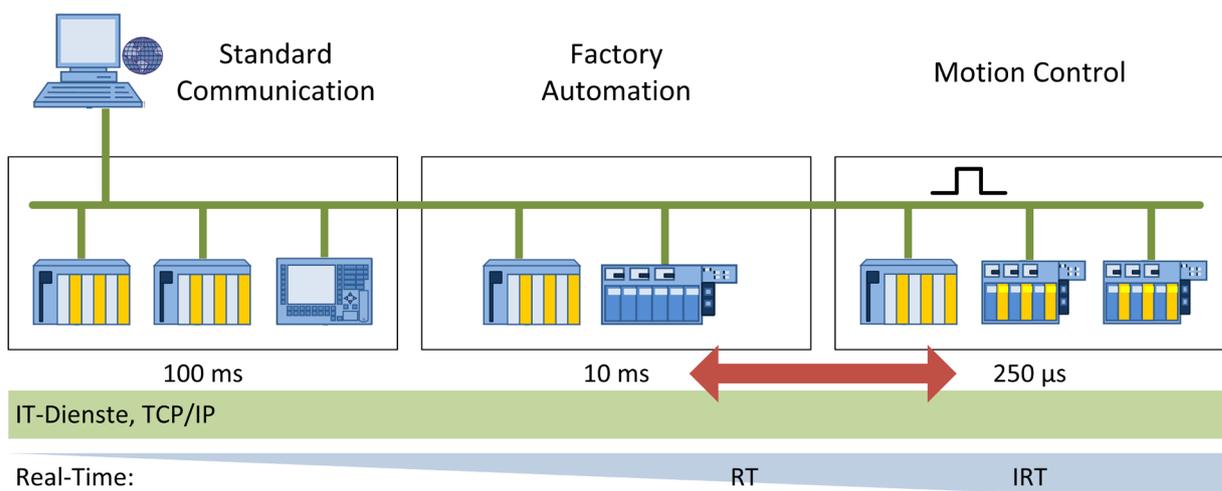


Abb. 4-5: Übersicht Echtzeitverhalten PROFINET [11]

Für den Zweck dieser Arbeit – die Evaluierung von Positionssensoren – ist es ausreichend, konstante Drehzahlen vorzugeben. Da keine zeitkritischen Anforderungen an die Kommunikation mit dem elektrischen Antrieb gestellt werden, ist die Verwendung der

Konformitätsklasse CC-A ausreichend. Ein großer Vorteil, da somit die bereits vorhandene Standard-Ethernet-Hardware verwendet werden kann. Abb. 4-6 zeigt das zugehörige PROFINET Systemmodell. Da das Antriebssystem SINAMICS S120 bereits als PROFINET IO Device – entspricht im Wesentlichen der Form eines Slave – feststeht, übernimmt der LabVIEW-PC die Rolle des PROFINET IO Controllers (Master).

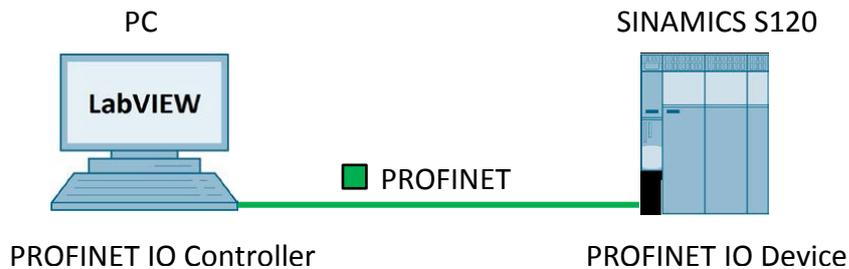


Abb. 4-6: LabVIEW-PC als PROFINET IO Controller

4.1.2 PN Driver

Mit dem PROFINET Kommunikationstreiber (PN Driver) der Firma Siemens können PC-basierte PROFINET-Anwendungen realisiert werden. Solch eine Anwendung ist die Verbindung von LabVIEW mit einem oder mehreren PROFINET IO Devices (siehe [12]). Dabei wird eine auf dem PN Driver basierende Programmbibliothek (DLL) in LabVIEW eingebunden. Diese stellt Funktionen zur Kommunikation mit PROFINET IO Devices direkt aus LabVIEW heraus zur Verfügung. Abb. 4-7 zeigt den schematischen Aufbau der Kommunikation zwischen LabVIEW und dem elektrischen Antrieb mittels PN Driver.

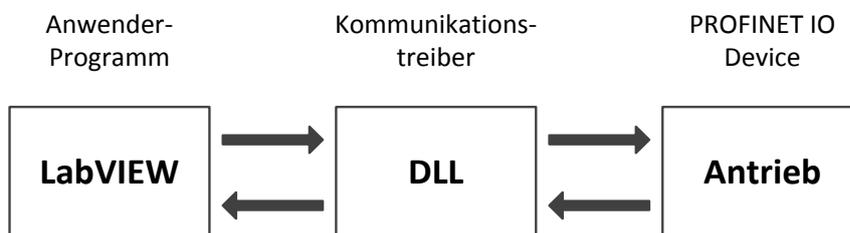


Abb. 4-7: Flussdiagramm PN Driver [12]

Die eingesetzte Programmbibliothek (pno.dll) ermöglicht das Initialisieren und Beenden einer PROFINET Verbindung und den zyklischen Austausch von IO Daten. Die minimal erreichbare Zykluszeit beträgt 32 ms. Die Einhaltung kann jedoch, da das eingesetzte Betriebssystem (Windows) keine Echtzeitfähigkeit besitzt, nicht immer (beispielsweise bei

CPU Überlast) garantiert werden. Der hardwarenahe Zugriff auf die Netzwerkkarte wird durch den Einsatz der freien Programmbibliothek WinPcap erreicht.

4.1.3 PROFIdrive

PROFIdrive ist das Standardprofil für Antriebsgeräte in Verbindung mit den Kommunikationstechnologien PROFIBUS und PROFINET. Die Verwendung offener Antriebsprofile bringt eine einfache und durchgängige Kopplung von Antrieben und Steuerungen unterschiedlicher Hersteller über Kommunikationssysteme [13].

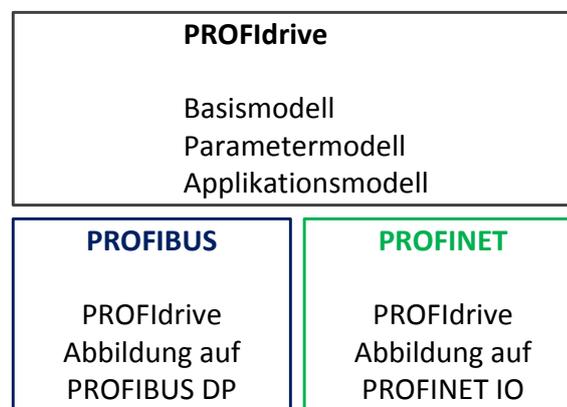


Abb. 4-8: Architektur von PROFIdrive [13]

Die grundlegenden Festlegungen im Profil von PROFIdrive (Abb. 4-8) sind die Definitionen von Basis-, Parameter- und Applikationsmodell. Diese sind vom Kommunikationssystem unabhängig, womit der Betrieb einer Anwendung ohne Änderung sowohl mit PROFIBUS DP – wird in dieser Arbeit nicht verwendet – als auch mit PROFINET IO gewährleistet ist.

Die Art der Einbindung eines Antriebs in eine Automatisierungslösung kann stark variieren. PROFIdrive definiert 6 verschiedene Applikationsklassen mit denen einfache Standardantriebe, Positionierantriebe, zentrale Bewegungsführungen bis hin zu dezentralen Automatisierungen bei getakteten Prozessen abgedeckt werden. Die Regelung eines Antriebs mit Vorgabe eines Drehzahlsollwerts mittels PROFINET wird durch die Applikationsklasse 1 (Abb. 4-9) beschrieben. Die komplette Drehzahlregelung erfolgt dabei im Antrieb.

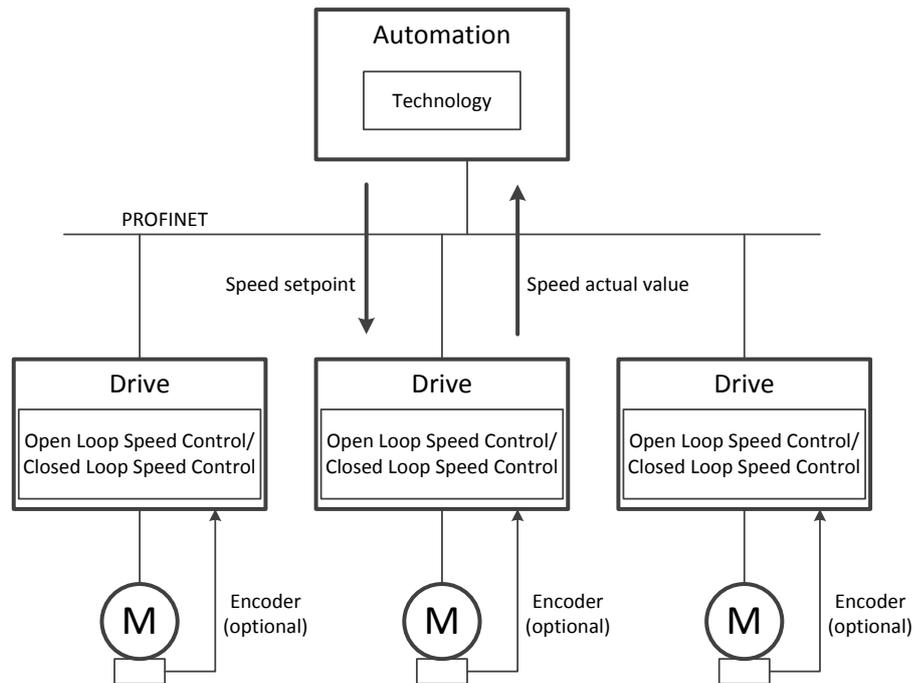


Abb. 4-9: Applikationsklasse 1 (Standardantrieb) [14]

Jede Applikationsklasse stellt verschiedene Telegramme zur Verfügung. Diese dienen zur Definition der Prozessdaten, welche zyklisch übertragen werden. Neben bereits vorgegebener Standardtelegramme und herstellerspezifischen Telegrammen gibt es auch die Möglichkeit der freien Telegrammerstellung. Für die zyklische Übertragung eines Drehzahl Sollwerts stehen verschiedene Standardtelegramme (siehe Abb. A-3) zur Verfügung, gewählt wurde das Standardtelegramm 2 – *Drehzahl Sollwert 32 Bit*. Die zu übertragenden Daten (Prozessdaten PZD) werden wortweise organisiert (1 Wort = 16 Bit), Tabelle 4-1 zeigt den Aufbau des Telegramms. Aus der Sicht des Antriebsobjekts stellen die empfangenen Prozessdaten die Empfangswörter (Steuerwörter oder Sollwerte) und die zu sendenden Prozessdaten die Sendewörter dar. Die Steuerwörter (STW1, STW2) ermöglichen durch das Setzen bzw. Löschen einzelner Bits den Zugriff auf die Grundfunktionen des Antriebs, wie die Freigabe eines Sollwerts, Ein- bzw. Ausschalten des Antriebs, Quittieren einer Störung, etc. Im Gegenzug liefern die Zustandswörter (ZSW1, ZSW2) Informationen zum Zustand des Antriebs, wie unter anderem Einschaltbereitschaft und Störungen, siehe [14].

Tabelle 4-1: Prozessdaten des Standardtelegramms 2 [15]

	Empfangs- Telegramm	Sende- Telegramm
PZD1	STW1	ZSW1
PZD2	NSOLL_B	NIST_B
PZD3		
PZD4	STW2	ZSW2

Der Drehzahlsollwert wird für die Darstellung als 32 Bit Drehzahlsollwert (NSOLL_B) entsprechend normiert. Das höchstwertige Bit bestimmt das Vorzeichen des Sollwerts, die restlichen 31 Bit werden für die Abbildung des Drehzahlbereichs bis zur doppelten Nenn Drehzahl der Maschine genutzt. Somit entspricht die Nenn Drehzahl einem dezimalen Wert von $2^{30} = 4000\ 0000$ hex, siehe Abb. 4-10. Dieselbe Darstellung wird für den 32 Bit Drehzahlwert (NIST_B) verwendet.

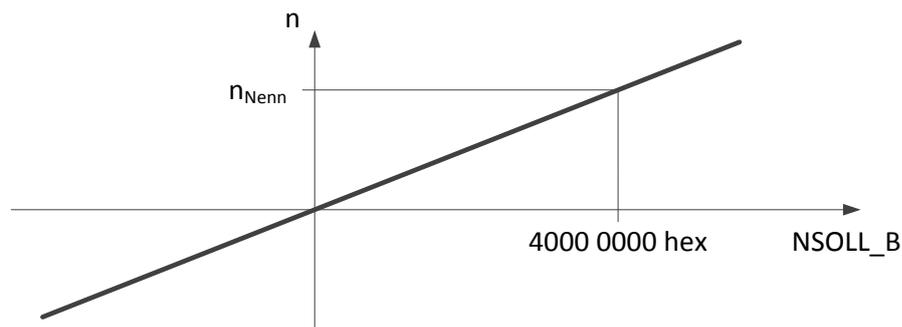


Abb. 4-10: Normierung der Drehzahl [14]

Für die Bestimmung von NSOLL_B aus dem absoluten Sollwert n_{Soll} (4-1) und für die Bestimmung von NIST_B aus dem 32 Bit Drehzahlwert n_{Ist} (4-2) und der Nenn Drehzahl des Antriebs n_{Nenn} gelten die folgenden beiden Beziehungen:

$$NSOLL_B = \frac{2^{30}}{n_{Nenn}} \cdot n_{Soll} \quad (4-1)$$

$$n_{Ist} = \frac{n_{Nenn}}{2^{30}} \cdot NIST_B \quad (4-2)$$

4.2 Positioniersystem

Jede der vier eingesetzten Positioniersteuerungen (PS10) besitzt eine USB-Schnittstelle (Kap. 4.2.2). Die Verbindung zwischen LabVIEW-PC und einer Steuerung ist als sogenannte COM-Brücke realisiert. Der Windows-Gerätetreiber erkennt die Steuerung als USB-Serial-Port und weist ihr eine COM-Port-Nummer zu. Die Verbindung via USB-Schnittstelle wird nach erfolgreicher Installation zu einer virtuellen RS-232-Schnittstelle (Kap. 4.2.1). Die Kommunikation erfolgt mittels Befehlen (siehe [16]). Diese werden in Form von ASCII-Zeichen über die virtuelle RS-232-Schnittstelle übertragen. Vom Hersteller steht eine Programmbibliothek (ps10.dll) zur Einbindung in LabVIEW zur Verfügung. Diese beinhaltet neben dem kompletten Befehlssatz der Positionssteuerung auch die Parametrierung der virtuellen RS-232-Schnittstelle, wie Baudrate, Parität, Anzahl der Datenbits, etc.

Abb. 4-11 zeigt die Vorgangsweise zum Herstellen sowie Beenden des Zugriffs auf eine Achse des Positioniersystems. Beim Verbindungsaufbau wird zuerst die Steuerung anhand ihrer COM-Port-Nummer initialisiert und der Steuerung eine sogenannte Control Unit ID (frei wählbare Zahl) zugewiesen, anhand dieser erfolgt für alle weiteren Befehle die Auswahl der passenden Positioniersteuerung. Anschließend wird die mit der Steuerung verbundene Achse initialisiert und deren Antrieb bestromt. Dieser Vorgang wird ebenfalls beim Verbindungsabbau, jedoch in umgekehrter Reihenfolge, durchgeführt.

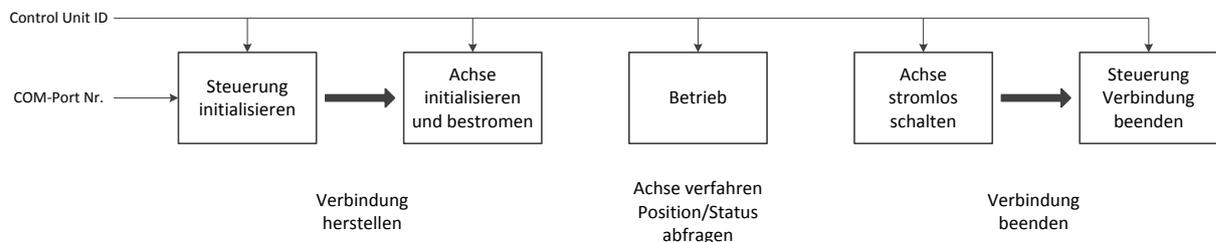


Abb. 4-11: Flussdiagramm Positionssteuerung

4.2.1 RS-232

RS-232 (auch „serielle Schnittstelle“ oder COM-Port genannt) beschreibt einen Standard für die serielle Datenübertragung zwischen PCs und Peripheriegeräten, die als Punkt-zu-Punkt Verbindung aufgebaut ist. Die Datenbits werden dabei nacheinander (seriell) übertragen. Die Übertragung ist zeichengebunden (oft ASCII-codiert) und asynchron. Jedes Zeichen wird daher zeitlich unabhängig von den anderen gesendet. Der Empfänger synchronisiert sich auf jedes eintreffende Zeichen neu, dieses ist durch Start- und Stoppbits gerahmt. Optional wird zur Erkennung von Übertragungsfehlern der Übertragungsrahmen um ein Paritätsbit

erweitert. Vor Beginn einer Datenübertragung müssen sich Sender und Empfänger auf die gleichen Protokollparameter (Tabelle 4-2) einstellen [17], [18].

Tabelle 4-2: Protokollparameter RS-232 [17]

Parameter	Wert
Codelänge	5-8 Bit
Paritätsbit	even, odd, none
Stoppbit(s)	1 oder 2 Bit(s)
Übertragungsgeschwindigkeit	19200 Bit/s
	9600 Bit/s
	4800 Bit/s
	2400 Bit/s
	1200 Bit/s
	600 Bit/s
	300 Bit/s
	150 Bit/s
	110 Bit/s

RS-232-Schnittstellen arbeiten bipolar mit +12 V/-12 V Spannungspegeln. Tabelle 4-3 zeigt die Zuordnung zu den logischen Werten.

Tabelle 4-3: Elektrische Eigenschaften RS-232 [17]

Spannungspegel	Logische Zuordnung
-12V ... -3V	Eins
-3V ... +3V	undefiniert
+3V ... +12V	Null

4.2.2 Universal Serial Bus

Der Universal Serial Bus (USB) ist ein serielles Bussystem und wurde zur Vereinheitlichung der verschiedenen Standardschnittstellen zwischen PCs und Peripheriegeräten, wie RS-232-, Drucker-, Tastatur-, Mausschnittstelle, etc. eingeführt. USB unterstützt das Erweitern und Verschalten bei Laufzeit und wird als Baumstruktur aufgebaut. Verwaltet wird das USB-Netz durch einen Masterknoten (meist ein PC), der Identifizierung und Adressierung der Geräte durchführt und die Punkt-zu-Punkt Verbindungen zwischen Ports teilnehmender Knoten erstellt [18].

4.3 Temperatursystem

Der Temperaturschrank besitzt zur Kommunikation eine RS-485-Schnittstelle. Diese wird mit Hilfe eines Schnittstellenwandlers (USB-RS485-Adapter) mit einer USB-Schnittstelle des LabVIEW-PCs verbunden. Nach Installation des Adapters wird dieser unter Windows als weiterer COM-Port wahrgenommen.

4.3.1 RS-485

Die RS-485-Schnittstelle stellt ebenso wie RS-232 (Kap. 4.2.1) eine asynchrone serielle Schnittstelle dar, unterstützt aber zusätzlich die gleichberechtigte Anschaltung mehrerer paralleler Knoten innerhalb eines Segments in einer Busstruktur. Durch den Übergang auf eine symmetrische Signalübertragung — ein Bit wird als Differenz zweier Spannungen abgebildet — werden Gleichtakt-Störungen unterdrückt und es sind höhere Datenraten möglich. Mit in sich verdrehten Adernpaaren wird die Symmetrierung weiter verbessert. Neben dem Vollduplex-Betrieb mit je einem Adernpaar zum Senden bzw. Empfangen von Daten ist auch ein Halbduplex-Betrieb mit nur einem Adernpaar möglich [18].

The screenshot shows a terminal window titled "STOP" with a status bar indicating "5/ 5 (SAT)" and "19:07:10". The main content is the "RS-485 Communication Setup" menu. The "Address" field is a 16-digit hexadecimal display with digits 1 through 16. The "Delimiter" field has radio buttons for "CR" and "LF". The "Data bits" field has buttons for "8bit" and "7bit". The "Stop bits" field has buttons for "1bit" and "2bit". The "Transmission Rate (bps)" field has buttons for "4800", "9600", and "19200". The "Protocol" field has buttons for "STND" and "E-BUS". The "Parity" field has buttons for "none", "odd", and "even". The "Echo Back" field has a button for "ON/OFF". At the bottom, there is a message: "Specify delimiter and transmission rate to enable communications." and a "Prev. Screen" button.

Abb. 4-12: Protokollparameter RS-485 Thermoschrank [19]

Die Protokollparameter bestehen, sehr ähnlich zur RS-232-Schnittstelle (Tabelle 4-2), aus Übertragungsgeschwindigkeit, Anzahl Daten- bzw. Stoppbits und einem optionalen Paritätsbit. Da RS-485 aber auch die Anordnung mehrerer Geräte als Busstruktur ermöglicht, wird pro Gerät eine Adresse vergeben. Zusätzlich ist neben Protokollwahl auch noch die

Auswahl eines Abschlusszeichens (Delimiter, siehe Kap. 4.3.2) möglich. Abb. 4-12 zeigt die Konfiguration der RS-485 Protokollparameter des Temperoschranks.

4.3.2 Kommunikationsprotokoll

Die Kommunikation zwischen LabVIEW-PC und Temperoschrank folgt einem Master/Slave-Prinzip. Dabei sendet der Slave (Temperoschrank) immer nur Daten, wenn er vom Master (PC) dazu aufgefordert wird. Aus Sicht des PCs werden daher Abfragen und Befehle an den Temperoschrank gesendet, dieser antwortet mit den gewünschten Informationen bzw. mit dem aktuellen Empfangsstatus. Eine Abfrage wird benutzt, um etwa die aktuelle Temperatur in der Temperaturbox oder den Heiz-/Kühl-Zustand des Temperoschranks abzufragen. Ein Befehl hingegen dient beispielsweise zur Einstellung eines bestimmten Betriebsmodus oder zur Vorgabe einer Solltemperatur. Abb. 4-13 zeigt schematisch den Kommunikationsablauf zwischen LabVIEW-PC und Temperoschrank für korrekte und ausführbare Befehle und Abfragen.

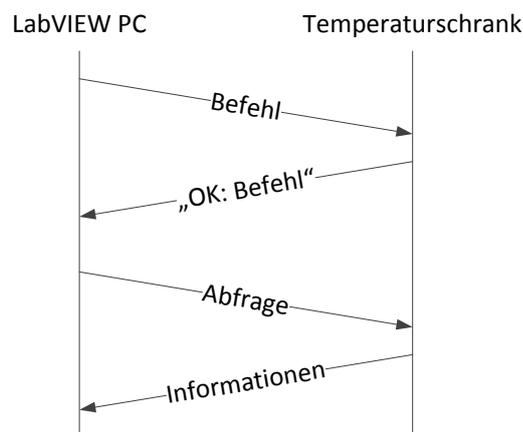


Abb. 4-13: Ablaufschema Kommunikation Temperoschrank [19]

Im Falle von nicht ausführbaren oder nicht verstandenen Kommandos antwortet der Temperoschrank mit einer Fehlermeldung (siehe [19]). Zwischen Erhalt einer Antwort und Senden eines neuen Kommandos schreibt der Hersteller eine Wartezeit von 0,3 s für Abfragen und 0,5 s für Befehle vor. Abb. 4-14 zeigt den Aufbau bestehend aus den Protokollparametern Adresse und Abschlusszeichen, dem eigentlichen Kommando bzw. eventuell erforderlichen Parametern, siehe [19]. Das Abschlusszeichen besteht wahlweise aus dem ASCII-Zeichen CR (carriage return: 0x0D), LF (line feed: 0x0A) oder der Hintereinanderreihung beider Zeichen.

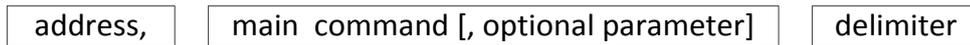


Abb. 4-14: Befehlsformat Temperaturschrank [19]

4.4 Spannungsversorgung

Das für die Spannungsversorgung eingesetzte Labornetzteil besitzt zur Kommunikation eine RS-232-Schnittstelle (4.2.1). Die Protokollparameter sind mit einer Übertragungsgeschwindigkeit von 2400 Bit/s und einer Datenlänge von 8 Bits mit einem Stoppbit und ohne Paritätsbit eindeutig vorgegeben. Das Labornetzteil besitzt, wie auch der Temperaturschrank (4.3.2), die Funktionalität eines Slave-Gerätes und sendet Daten ausschließlich auf Anfrage des LabVIEW-PCs. Es stehen 9 verschiedene Befehle, wie die Vorgabe der Ausgangsspannung und Begrenzungen, Aktivieren oder Deaktivieren des Ausgangs, Auslesen des Ausgangsstroms, usw. zur Verfügung. Jeder Befehl besteht aus 3 Bytes, siehe Abb. 4-15.



Abb. 4-15: Befehlsformat Labornetzteil [20]

Das erste Byte dient zur Identifikation des Befehls. Byte 2 und 3 beinhalten den Wert des entsprechenden Parameters, wie den Spannungswert bei Vorgabe der Ausgangsspannung, einen binären Wert für Aktivieren oder Deaktivieren des Ausgangs, usw. Bei Befehlen zur Abfrage von Statusinformationen des Netzteils besitzen Byte 2 und 3 keine Funktion und werden Null gesetzt. Die Antwortdaten des Labornetzteils besitzen das gleiche Befehlsformat wie die zu sendenden Befehle (Abb. 4-15). Für eine genauere Beschreibung der zur Verfügung stehenden Befehle und deren Format wird auf [20] verwiesen.

4.5 Messdatenerfassung

Für die automatisierte Durchführung von Versuchsplänen ist es notwendig, die Aufzeichnung der Messdaten mit LabVIEW zu steuern. Das direkte Einbinden der verwendeten Messkarte in LabVIEW ist nicht ohne weiteres möglich, zudem wird die Funktionalität der zur Messkarte zugehörigen Software DEWESoft benötigt. Das sogenannte TestBed-Plugin bietet die Möglichkeit, via RS-232-Schnittstelle und AK-Protokoll auf die grundlegenden Funktionen

von DEWESoft zuzugreifen. Da sich LabVIEW und DEWESoft auf einem Rechner befinden, reicht es aus die RS-232-Verbindung mit Hilfe eines Null-Modem Emulators zu simulieren.

4.5.1 Null-Modem Emulator

Die Verbindung von zwei Computern zur Datenübertragung über die serielle Schnittstelle (RS-232) wird als Null-Modem-Verbindung bezeichnet. Wird diese Verbindung zwischen zwei Anwendungen erstellt, die sich beide auf demselben Rechner befinden, kann anstatt der hardware-mäßigen Verbindung zweier COM-Ports ein Null-Modem Emulator verwendet werden. Der Null-Modem Emulator erstellt zwei virtuelle COM-Ports und verbindet diese miteinander, der Eingang des ersten virtuellen COM-Ports ist somit der Ausgang des zweiten virtuellen COM-Ports und umgekehrt, siehe Abb. 4-16.

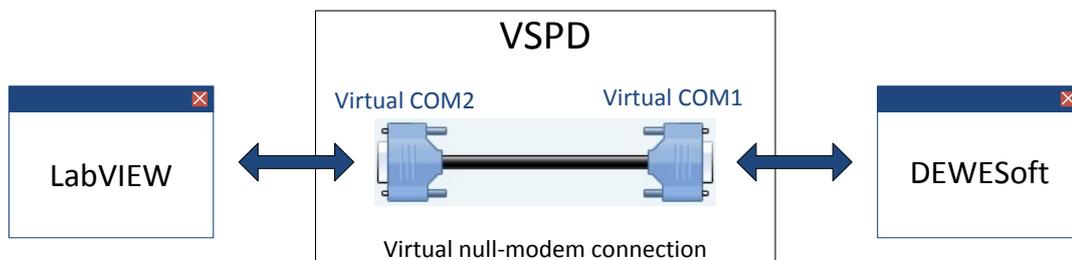


Abb. 4-16: Null-Modem-Verbindung [21]

Die Anwendungen LabVIEW sowie DEWESoft können nun mit je einem virtuellen COM-Port verknüpft werden. Es steht eine Reihe an Null-Modem-Emulatoren zur Verfügung, in dieser Arbeit wurde die freie Software com0com verwendet.

4.5.2 AK-Protokoll

Das AK-Protokoll beschreibt einen Standard (ISO 16016) für die Kommunikation an Abgasprüfständen zwischen Hauptcomputer und dezentraler Abgasmesstechnik. Als Übertragungsmedium für das AK-Protokoll kann entweder die RS-232- oder die Ethernet-Schnittstelle verwendet werden. Die hierarchische Verwaltung der Kommunikation wird, wie bereits bei fast allen der eingesetzten Prüfstandskomponenten in dieser Arbeit, gegliedert in einen Master (LabVIEW) und einen Slave (DEWESoft). Das AK-Protokoll umfasst zwei verschiedene Formate, D2T- sowie Tornado-Befehlssatz, wobei sich diese im Syntax, kaum aber in der Funktionalität unterscheiden. Für diese Arbeit relevante Funktionen des D2T-Befehlssatzes sind das Starten bzw. Stoppen einer Messung und das Abspeichern der Messdaten unter einem vorgegebenen Dateinamen. Abb. 4-17 zeigt den Aufbau der

ASCII-codierten Befehle. Diese beginnen mit dem ASCII-Zeichen 0x02 (start of text) und enden mit 0x03 (end of text). Die eigentlichen Befehle bestehen jeweils aus einem Unterstrich und vier Zeichen gefolgt von je nach Art des Befehls abhängigen optionalen Parametern. Für eine Auflistung und vollständige Beschreibung aller Befehle wird auf [22] verwiesen.



Abb. 4-17: Befehlsformat D2T [22]

5 Implementierung in LabVIEW

5.1 LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) ist die Entwicklungsumgebung der zugehörigen grafischen Programmiersprache G. Diese wurde entwickelt, um eine leistungsstarke Programmiersprache für die Kommunikation mit Geräten, zur Datenanalyse und zur Visualisierung der Ergebnisse zur Verfügung zu stellen [23]. Abb. 5-1 zeigt exemplarisch ein einfaches LabVIEW-Programm, welche als virtuelle Instrumente (VI) bezeichnet werden und aus zwei Komponenten bestehen. Der graphische Programmcode wird im sogenannten Blockdiagramm erstellt, die Bedienoberfläche (Frontpanel) stellt die Schnittstelle zum Benutzer dar.

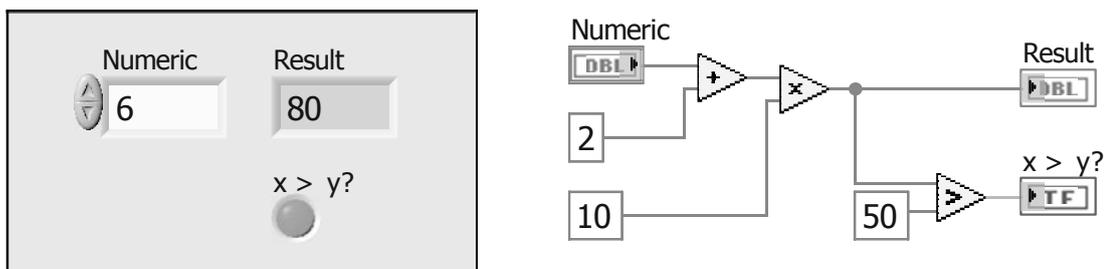


Abb. 5-1: Frontpanel (links) und Blockdiagramm (rechts) eines LabVIEW-Programms [23]

Das Beispielprogramm hat folgende Funktion. Die einzugebende Zahl „Numeric“ wird nach Addition der Zahl 2 mit 10 multipliziert und in der Anzeige „Result“ dargestellt. Der Indikator „x > y“ zeigt an, ob das berechnete Ergebnis größer als 50 ist. LabVIEW arbeitet nach dem Datenflusskonzept, das heißt der Additionsblock wird erst ausgeführt, wenn die Eingabe vorhanden ist, der Multiplikationsblock, wenn das Ergebnis der Addition zur Verfügung steht, usw. Auf diese Weise lassen sich Programme sehr schnell und intuitiv entwickeln und sind, wie etwa ein Schaltplan oder Flussdiagramm sehr gut lesbar. Zudem stellt LabVIEW eine umfangreiche Bibliothek mit mathematischen Funktionen, zur Kommunikation mit Geräten, zur Datenerfassung mit Messkarten oder zur Einbindung von Programmcode anderer Sprachen (z.B. Matlab) zur Verfügung. Typische Anwendungsgebiete sind beispielsweise die industrielle Bildverarbeitung, Mathematik und Statistik, Regelungstechnik, diverse Echtzeitanwendungen, Web-basierte Automatisierung, Dokumentation, usw.

5.2 POSE_Manual

Das LabVIEW VI POSE_Manual (PO Positions-, SE Sensoren) dient zur manuellen Bedienung der Prüfstandskomponenten. Es dient für Aufgaben, wie beispielsweise die mechanische Justierung eines Sensors und der Inbetriebnahme. Abb. 5-2 zeigt die Benutzeroberfläche des Programms, strukturiert nach den Komponenten Positioniersystem, elektrischer Antrieb, Temperatursystem und Spannungsversorgung.

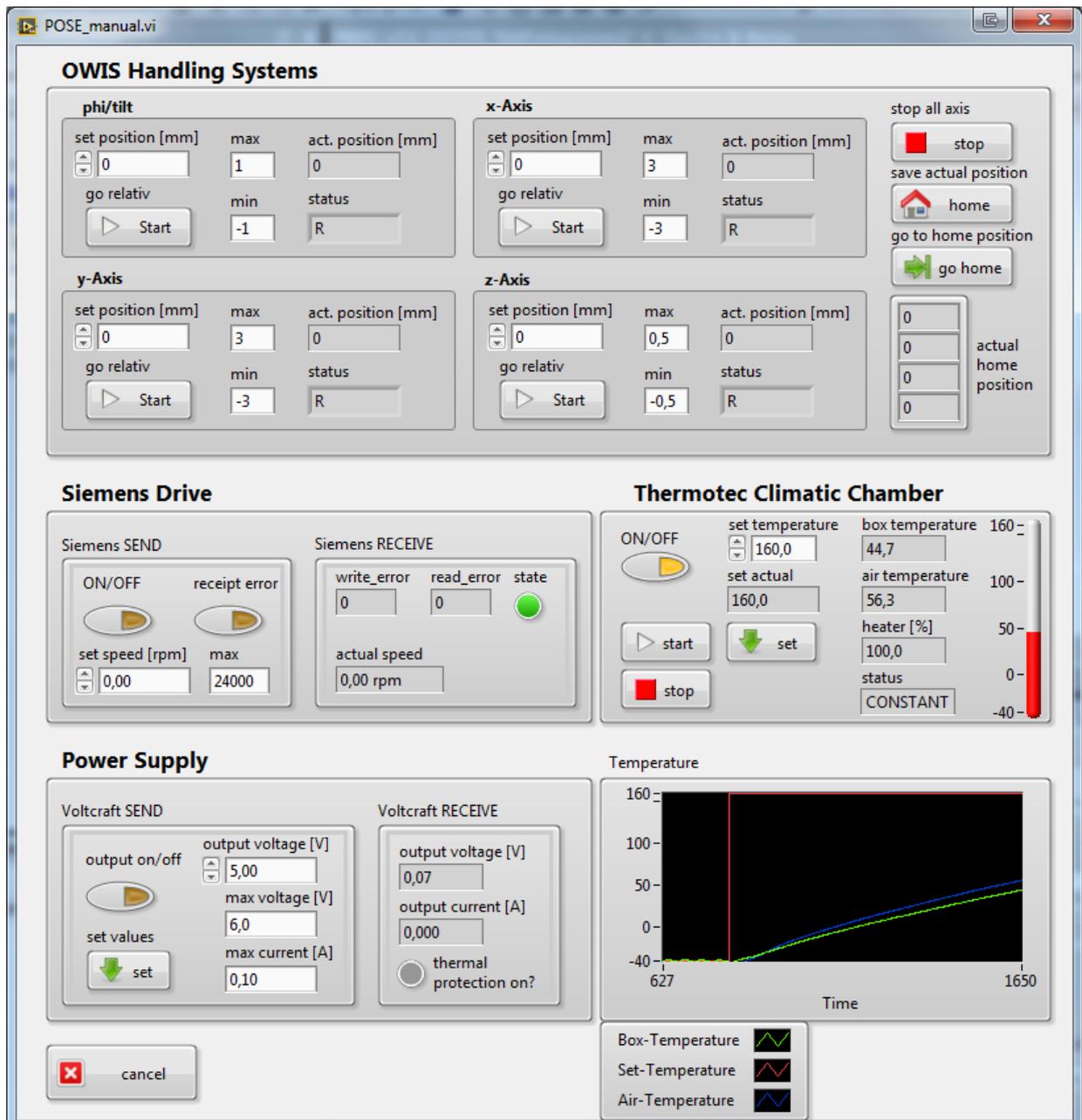


Abb. 5-2: Benutzeroberfläche POSE_Manual.vi

Die Bedienoberfläche des Positioniersystems besitzt für alle vier Achsen je ein Panel zum relativen Verfahren der Position, sowie der Anzeige der aktuellen Position und des Betriebsstatus. Zudem kann via Min/Max-Eingabe der Bereich für gültige Positionen, relativ zur sogenannten Home-Position, eingegrenzt werden. Die Home-Position ist im Normalfall die zentrale Ausrichtung des Sensors bzw. die Position mit minimalem Fehler (sweet-spot) und kann per Button („save actual position“) abgespeichert — die Anzeigen der aktuellen Position werden damit auf Null gesetzt — und auch direkt angefahren werden („go to home position“). Für einen Abbruch der Positionierung sorgt der Stopp-Button, dieser stoppt unmittelbar das Verfahren aller vier Achsen.

Für den elektrischen Antrieb sind Eingabemöglichkeiten für die Vorgabe der Solldrehzahl und der maximal zulässigen Drehzahl (beide Drehrichtungen), sowie zum Ein- und Ausschalten des Antriebs und Quittieren allfälliger, antriebsspezifischer Fehler vorhanden. Angezeigt werden die aktuelle Istdrehzahl, Statusanzeige des Antriebs und etwaige Kommunikationsprobleme in Form von Fehlercodes (siehe [12]).

Der Eingabebereich des Temperatursystems besteht neben Ein-/Ausschaltknopf im Wesentlichen aus der Vorgabe der Solltemperatur. Angezeigt werden die aktuelle Temperatur in der Thermobox und im Klimaschrank, der aktuelle Betriebsmodus und die aktuelle Heizleistung in Prozent. Zusätzlich wird der zeitliche Verlauf der aktuellen Temperaturwerte in einem Diagramm dargestellt.

Die Spannungsversorgung ermöglicht in ihrem Teil der Bedienoberfläche die Vorgabe des Sollwerts der Ausgangsspannung, die Vorgabe der Werte für Spannungs- sowie Strombegrenzung und die Möglichkeit die vorgewählte Spannung am Ausgang des Netzteils an- und auszuschalten. Dargestellt werden die aktuelle Ausgangsspannung, der Ausgangsstrom und der Status der thermischen Begrenzung.

Abb. 5-3 zeigt schematisch den Ablauf des Programms POSE_Manual. Im ersten Schritt werden die Komponenten initialisiert. Beim Temperatursystem besteht die Initialisierung lediglich aus dem Einschalten des Temperaturschranks, dies wird erst dann durchgeführt, wenn das Temperatursystem — dieses besitzt einen hohen Energieverbrauch — auch wirklich benötigt wird. Ist die Initialisierung erfolgreich, läuft für jede Komponente innerhalb einer Schleife der eigentliche Betriebsvorgang ab. Um den Prozessor nicht unnötig zu belasten bzw. auch um Timings während der Kommunikation einzuhalten, sind innerhalb der Schleifen entsprechende Wartezeiten vorhanden. Tritt während der Kommunikation mit einer Prüfstandskomponente ein Fehler auf, wird die jeweilige Schleife verlassen und der Betrieb der zugehörigen Komponente eingestellt. Mit dem Cancel-Button kann der Benutzer die initialisierten Verbindungen zu den Komponenten schließen und das Programm beenden.

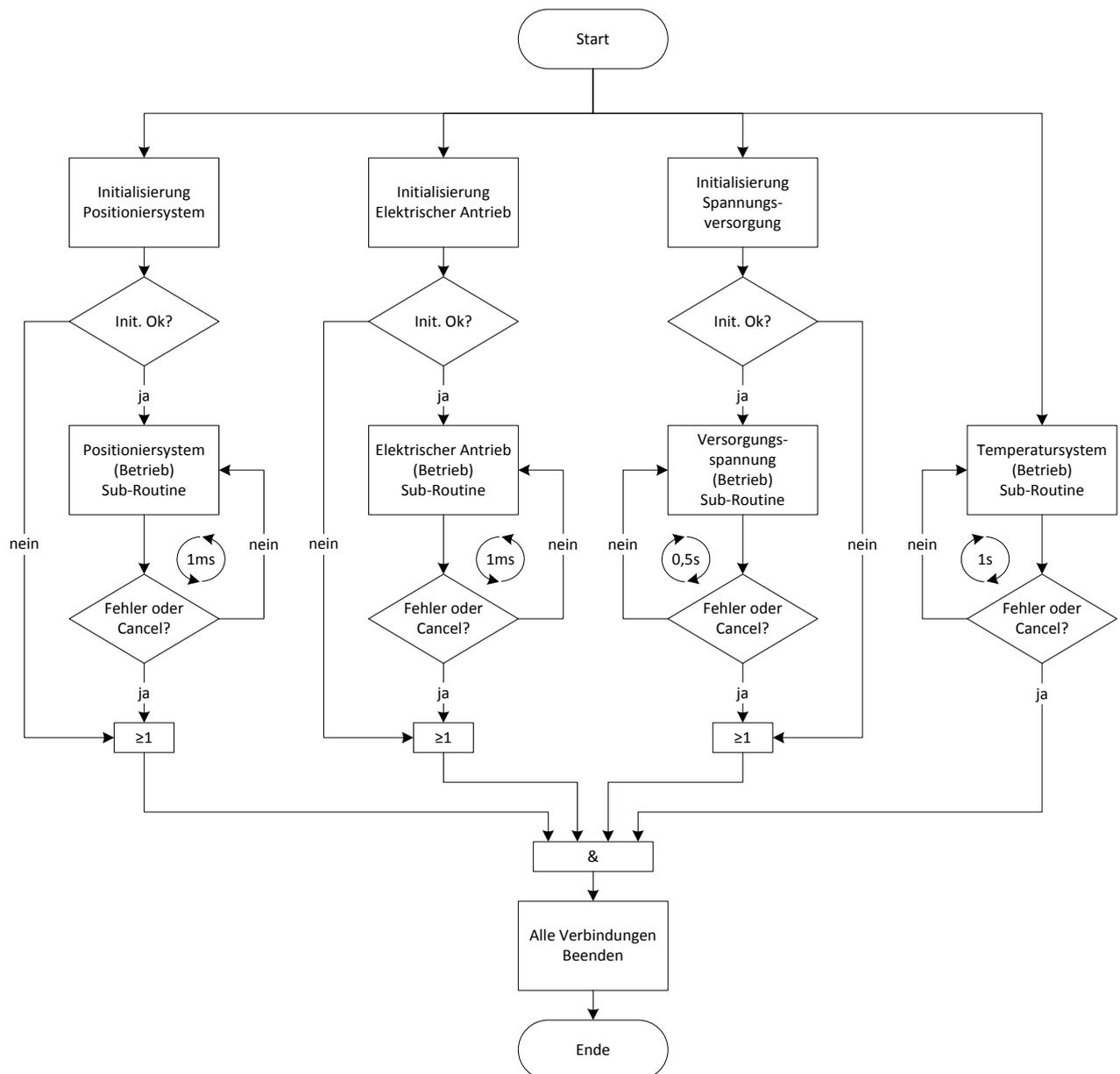


Abb. 5-3: Flussdiagramm POSE_Manual

5.3 POSE_Auto

Das Programm POSE_Auto dient zur automatisierten Messung verschiedener Prüfzenarien die in Form von Versuchsplänen vorgegeben werden. Diese können direkt in einem Unterprogramm oder extern erstellt und anschließend geladen werden. Abb. 5-4 zeigt die Benutzeroberfläche des Programms. Die Tabelle „test table“ beinhaltet die Werte des aktuell gewählten Versuchsplans. Diese können editiert und der ganze Versuchsplan anschließend gespeichert werden. Beim Laden eines bereits erstellten Versuchsplans, kann dieser für die Messung mehrerer Durchläufe dupliziert werden.

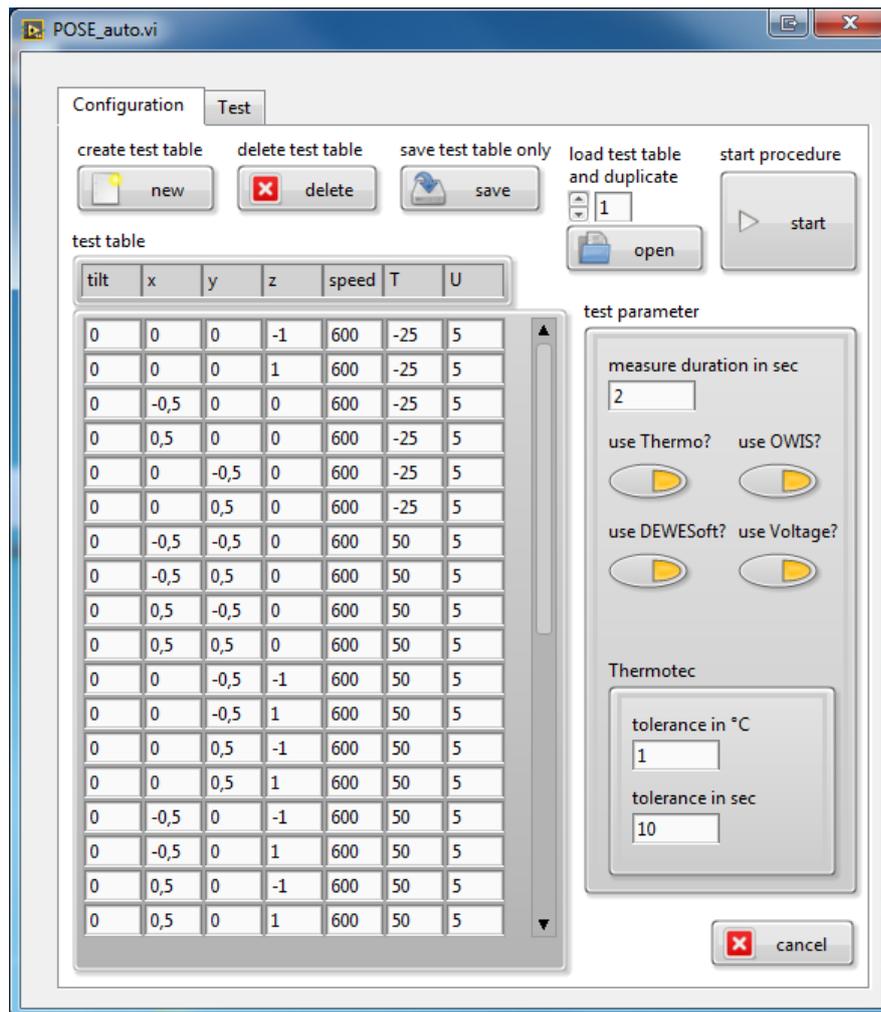


Abb. 5-4: Benutzeroberfläche POSE_Auto.vi - Konfiguration

Das Panel „test parameter“ dient zur Auswahl der Prüfstandskomponenten, die bei der Durchführung eines automatisierten Prüfablaufs benötigt werden. Das Eingabefeld „measure duration“ bestimmt, wie lange die Messdatenerfassung für den jeweiligen Messpunkt Werte aufzeichnet. Soll der Versuchsplan ohne Erfassung von Messwerten durchgeführt werden — „use DEWESoft?“ wird deaktiviert — kann ausgewählt werden, ob die Messpunkte automatisch mit Abwarten der angegebenen Testdauer oder manuell durch Abwarten einer Eingabe abgearbeitet werden.

Da die Regelung der Temperatur aufgrund der Trägheit des Systems „Temperaturschrank mit externer Temperaturbox“ zum Überschwingen neigt und die Dauer für das Abklingen der Schwingung mitunter sehr hoch ist, kann ein Toleranzband für das Erreichen des Temperatursollwerts angegeben werden. Dazu wird ein Toleranzwert für die Temperatur und eine Zeitdauer angegeben. Erst wenn der aktuelle Temperaturwert für die angegebene Zeitdauer innerhalb der Toleranz bleibt, wird die Messung durchgeführt.

The interface includes the following controls and data:

- OWIS tilt:** constant, 0
- dphi [deg]:** 0
- OWIS x:** variable, x min [mm]: -0,5, x max [mm]: 0,5
- OWIS y:** variable, y min [mm]: -0,5, y max [mm]: 0,5
- OWIS z:** variable, z min [mm]: -1, z max [mm]: 1
- Siemens:** constant, 600
- speed [rpm]:** 600
- Thermotec:** variable, T min [°C]: -25, T max [°C]: 125
- Voltage U:** constant, 5
- design method:** Box Behnken Design
- number of runs:** 1
- number of center points:** 10
- test table:**

tilt	x	y	z	speed	T	U
0	0	0	-1	600	-25	5
0	0	0	1	600	-25	5
0	-0,5	0	0	600	-25	5
0	0,5	0	0	600	-25	5
0	0	-0,5	0	600	-25	5
0	0	0,5	0	600	-25	5
0	-0,5	-0,5	0	600	50	5
0	-0,5	0,5	0	600	50	5
0	0,5	-0,5	0	600	50	5
0	0,5	0,5	0	600	50	5
0	0	-0,5	-1	600	50	5
0	0	-0,5	1	600	50	5
0	0	0,5	-1	600	50	5
0	0	0,5	1	600	50	5
0	-0,5	0	-1	600	50	5
0	-0,5	0	1	600	50	5
0	0,5	0	-1	600	50	5
0	0,5	0	1	600	50	5
0	0	0	0	600	50	5
0	0	0	0	600	50	5
- Buttons:** generate new testtable (refresh), use testtable (ok)
- save:** DUT (text field), file path control (X:\Work\Matlab\), save button (Save)

Abb. 5-5: Benutzeroberfläche zur Erstellung von Versuchsplänen

Für die Erstellung eines Versuchsplanes (Abb. 5-5) sind die Designmethoden Box-Behnken-Design, Central-Composite Faced, Central-Composite Inscribed, Full Factor sowie die zur Evaluierung benötigten Zufallspläne Latin Hypercube und Pseudo Random verfügbar. Für die zufällig erstellten Versuchspläne ist die Anzahl der Versuche wählbar, bei den Designmethoden ist diese meist schon vorgegeben, hier kann jedoch die Anzahl der Messungen im Mittelpunkt (Center) gewählt werden.

Die zu prüfenden Parameter werden entweder als konstant, mit der Angabe eines fixen Wertes, oder als variabel mit der Angabe eines Wertebereichs festgelegt. Dieser Bereich besteht aus einem Minimal- und einem Maximalwert, der für die Designmethoden notwendige Mittelpunkt wird aus dem Mittelwert bestimmt. Das Erstellen der Versuchspläne geschieht, wie auch das Speichern als MAT-File, mit dem MATLAB-Skript für LabVIEW. Mit diesem können in MATLAB-Syntax erstellte Programme (siehe Kap. A.3) in LabVIEW eingebettet werden. Ist ein Versuchsplan vorhanden und die Konfiguration der

Testparameter abgeschlossen, kann der Prüfablauf gestartet werden. Die Benutzeroberfläche des Messbetriebs (Abb. 5-6) zeigt den Fortschritt der bereits durchgeführten Versuche und den aktuellen Arbeitsvorgang. Die Statusinformationen der Prüfstandskomponenten werden ebenfalls angezeigt bzw. farblich ausgegraut, falls die jeweilige Komponente nicht für den Prüfablauf eingesetzt wird.

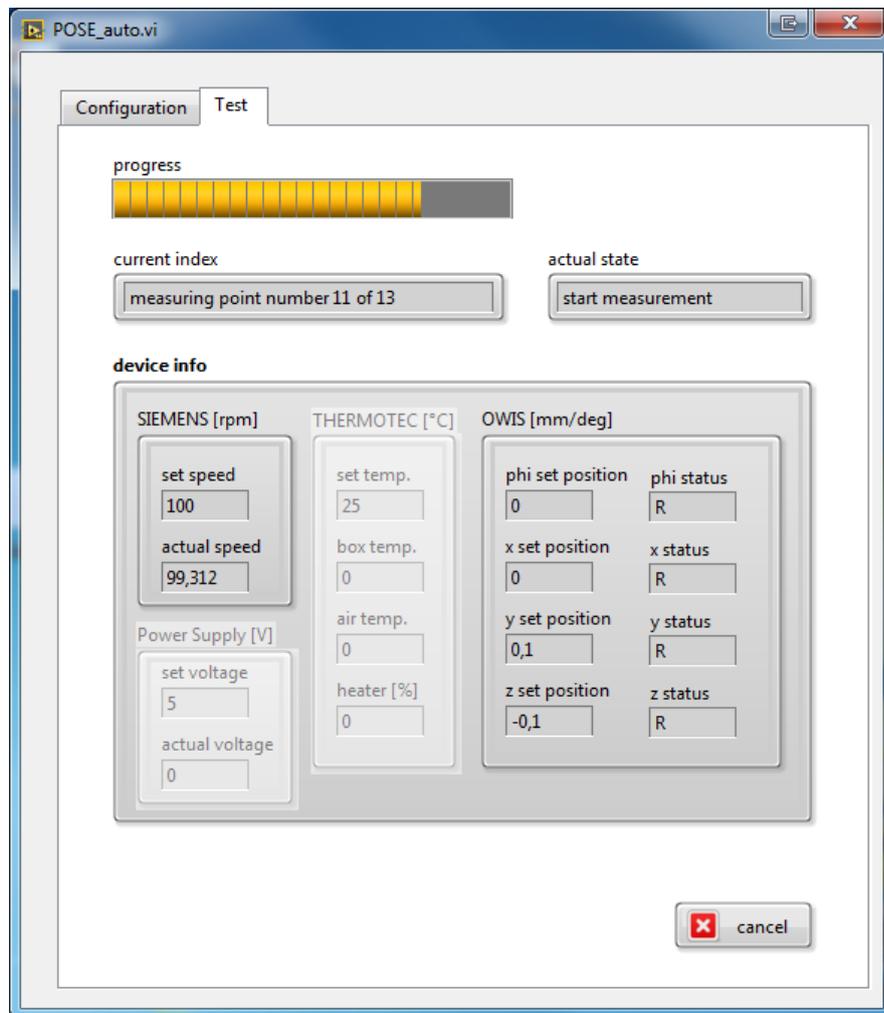


Abb. 5-6: Benutzeroberfläche POSE_Auto — Messbetrieb

Architektur und Funktionsweise des Programms POSE_Auto ist in Abb. 5-7 zu sehen. Nach der bereits beschriebenen Konfiguration durch den Benutzer werden je nach Wahl der Parameter die Komponenten elektrischer Antrieb, Positioniersystem, Temperatursystem und Spannungsversorgung initialisiert und anschließend die Sollwerte des ersten Messpunkts vorgegeben. Sind diese erreicht, wird die Messung gestartet und die Messdaten aufgezeichnet. Anschließend wiederholen sich Sollwert vorgeben, abwarten, Messdaten aufzeichnen für jeden Messpunkt im Versuchsplan.

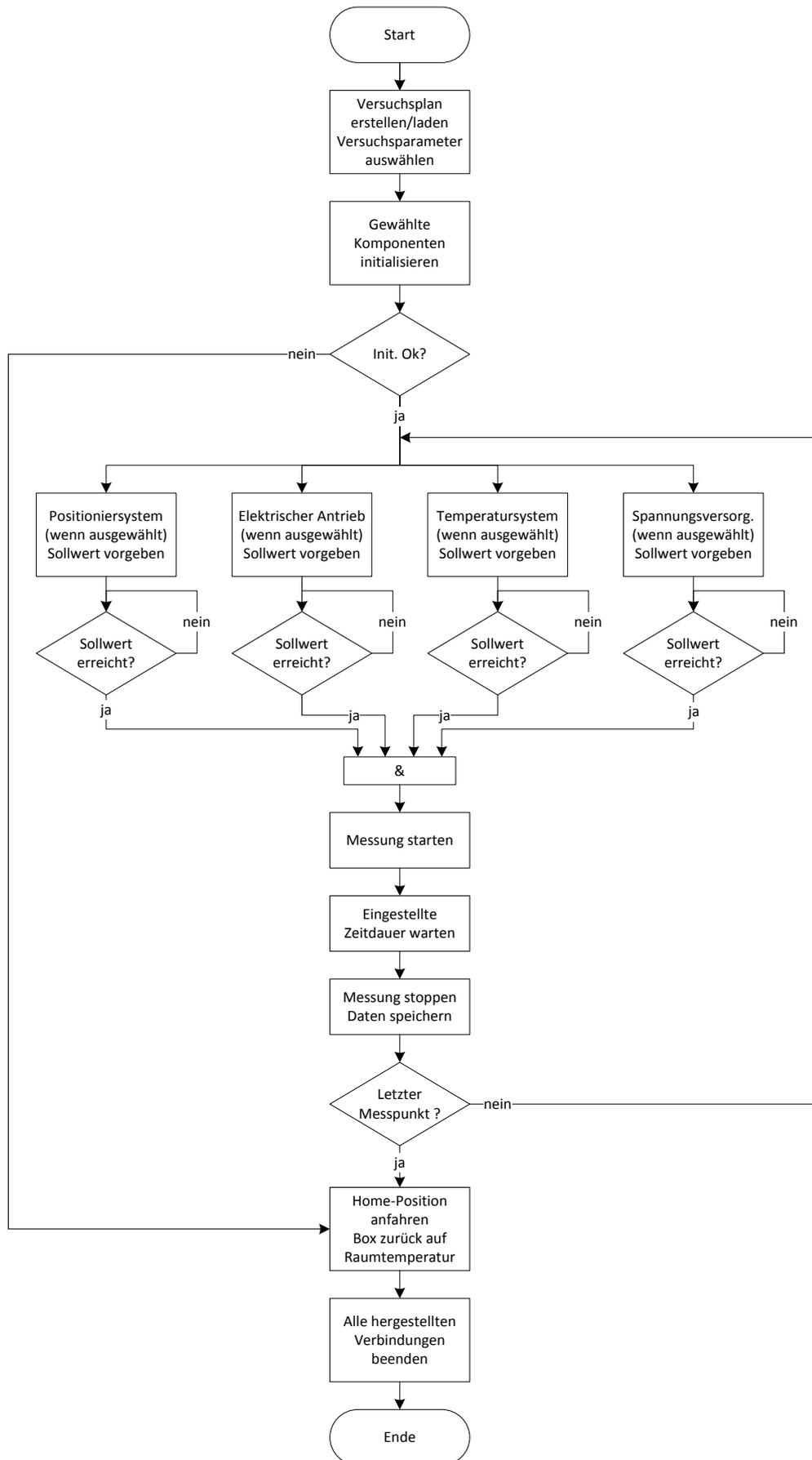


Abb. 5-7: Flussdiagramm POSE_Auto

Ist das Ende des Versuchsplans erreicht oder wird die Messung zwischenzeitlich vom Benutzer abgebrochen, wird das Positioniersystem wieder in seine ursprüngliche Home-Position bewegt, die Temperatur innerhalb der externen Temperaturbox wieder in den Bereich der Raumtemperatur gebracht und die Verbindungen zu den Komponenten beendet.

6 Zusammenfassung

Es wurde in dieser Arbeit ein Konzept für die Anbindung der Prüfstandskomponenten an eine zentrale Steuereinheit (LabVIEW-PC) erstellt und somit die Automatisierung des Prüfstands umgesetzt. Für die Inbetriebnahme oder für einzelne Messungen wurde eine einheitliche Bedienoberfläche zur Steuerung und Visualisierung der Komponenten geschaffen. Ein zusätzliches Programm ermöglicht die Erstellung und die autonome Durchführung verschiedener Versuchspläne.

Die automatisierte Abarbeitung der Versuchspläne bringt viele Vorteile und neue Möglichkeiten zur Evaluierung rotierender Positionssensoren. Die schnellere Durchführung der Versuchspläne bringt eine massive Zeitersparnis, zudem kann die Zeit während der automatisierten Messung für andere Tätigkeiten genutzt werden. Dank dieser effektiven Ausführung bietet es sich an, verschiedene Versuchspläne (Box-Behnken-Design, Central-Composite-Design, vollfaktorieller Versuchsplan, Kombinationen oder eigene Versuchspläne) zu evaluieren und zu vergleichen. Fehlerhafte Messungen durch Bedienfehler der Prüfstandskomponenten können praktisch ausgeschlossen und zudem eine hohe Reproduzierbarkeit der Messergebnisse gewährleistet werden. Ein weiterer Nutzen ist das Erzielen eines höheren Sicherheitslevels am Prüfstand, da dessen Bedienung durch den Anwender auch von außerhalb mittels Fernzugriff (Remote-Desktop) des abgeschlossenen Prüfstandsraumes möglich ist.

Abbildungsverzeichnis

Abb. 1-1: Antriebsstrang eines elektrischen Fahrzeugs [2].....	10
Abb. 2-1: Übersicht induktiv/magnetische Sensoren (VR ... Variable Reluktanz, AMR ... Anisotropic Magnetoresistance, GMR ... Giant Magnetoresistance)	13
Abb. 2-2: Schematischer Aufbau eines Resolvers [5].....	14
Abb. 2-3: Schematischer Aufbau und Ausgangssignale eines VR-Resolvers [2]	15
Abb. 2-4: Prinzip Wirbelstromsensor (links) und Sensorelektronik (rechts) [4]	16
Abb. 2-5: End of Shaft Vierfach-Hallsensor [3].....	17
Abb. 2-6: Blockdiagramm Tracking Schleife [5].....	18
Abb. 2-7: Schematisches Modell des in der Tracking-Schleife enthaltenen Regelkreises [6]..	20
Abb. 2-8: Winkelverlauf Resolver	21
Abb. 2-9: Fehlpositionierter AMR-Sensor: Resolver-Signale (oben) und berechneter Winkelverlauf (unten) [2]	22
Abb. 2-10: Central-Composite-Design für 3 verschiedene Parameter [7]	25
Abb. 2-11: Face-Centered CCD für 3 verschiedene Parameter [7].....	26
Abb. 2-12: Box-Behnken-Design für 3 verschiedene Parameter [7]	26
Abb. 3-1: Prüfstand Übersicht [2].....	29
Abb. 3-2: CAD-Modell des Prüfstands mit Koordinatensystem [8].....	30
Abb. 3-3: Übersicht elektrischer Antrieb.....	30
Abb. 3-4: Übersicht Positioniersystem	31
Abb. 3-5: Übersicht Temperatursystem	31
Abb. 3-6: Übersicht Spannungsversorgung	32
Abb. 3-7: Übersicht Messdatenerfassung	32
Abb. 3-8: Übersicht Konzept LabVIEW	33
Abb. 4-1: Schematischer Aufbau — Kommunikation mit dem elektrischen Antrieb	35
Abb. 4-2: Automatisierungspyramide [10].....	36
Abb. 4-3: Grundsystem PROFINET IO mit Kommunikationspfaden [9].....	36
Abb. 4-4: Aufbau der Konformitätsklassen [9].....	37
Abb. 4-5: Übersicht Echtzeitverhalten PROFINET [11]	37
Abb. 4-6: LabVIEW-PC als PROFINET IO Controller	38
Abb. 4-7: Flussdiagramm PN Driver [12]	38
Abb. 4-8: Architektur von PROFIdrive [13].....	39
Abb. 4-9: Applikationsklasse 1 (Standardantrieb) [14]	40
Abb. 4-10: Normierung der Drehzahl [14].....	41
Abb. 4-11: Flussdiagramm Positionssteuerung.....	42
Abb. 4-12: Protokollparameter RS-485 Thermoschrank [19].....	44

Abb. 4-13: Ablaufschema Kommunikation Temperaturschrank [19]	45
Abb. 4-14: Befehlsformat Temperaturschrank [19]	46
Abb. 4-15: Befehlsformat Labornetzteil [20].....	46
Abb. 4-16: Null-Modem-Verbindung [21]	47
Abb. 4-17: Befehlsformat D2T [22].....	48
Abb. 5-1: Frontpanel (links) und Blockdiagramm (rechts) eines LabVIEW-Programms [23] ...	49
Abb. 5-2: Benutzeroberfläche POSE_Manual.vi.....	50
Abb. 5-3: Flussdiagramm POSE_Manual	52
Abb. 5-4: Benutzeroberfläche POSE_Auto.vi - Konfiguration	53
Abb. 5-5: Benutzeroberfläche zur Erstellung von Versuchsplänen.....	54
Abb. 5-6: Benutzeroberfläche POSE_Auto — Messbetrieb	55
Abb. 5-7: Flussdiagramm POSE_Auto.....	56
Abb. A-1: Prüfstand zur Evaluierung rotierender Positionssensoren	64
Abb. A-2: Konfiguration des Null-Modem Emulators.....	66
Abb. A-3: PROFIdrive Standardtelegramme und Prozessdaten 1 [15].....	67

Tabellenverzeichnis

Tabelle 2-1: Vollfaktorieller Versuchsplan für 3 Parameter [7]	24
Tabelle 2-2: Versuchsplan Central-Composite-Design für 3 verschiedene Parameter [7]	25
Tabelle 2-3: Versuchsplan Box-Behnken-Design für 3 verschiedene Parameter [7]	27
Tabelle 3-1: Spezifikationen des Prüfstands [2]	28
Tabelle 4-1: Prozessdaten des Standardtelegramms 2 [15].....	41
Tabelle 4-2: Protokollparameter RS-232 [17].....	43
Tabelle 4-3: Elektrische Eigenschaften RS-232 [17]	43
Tabelle A-1: Komponenten des elektrischen Antriebs.....	65
Tabelle A-2: Komponenten des Positioniersystems.....	65
Tabelle A-3: Netzwerkkonfiguration des elektrischen Antriebs.....	65
Tabelle A-4: COM-Schnittstellenbelegung	65

Literaturverzeichnis

- [1] A. Kampker, D. Vallée und A. Schnettler, Elektromobilität - Grundlagen einer Zukunftstechnologie, Springer Vieweg, 2013, ISBN 978-3-642-31985-3.
- [2] J. Gächter, J. Fabian, M. Hirz, A. Schmidhofer und H. Lanzenberger, „Evaluation of Angular Sensor Systems for Rotor Position Sensing of Automotive Electric Drives,“ 18th International Forum on Advanced Microsystems for Automotive Applications (AMAA), Berlin, Germany, 06/2014.
- [3] K. Reif, Sensoren im Kraftfahrzeug, Springer Vieweg, 2012, ISBN 978-3-8348-1778-5.
- [4] C. Ebbesson, „Rotary Position Sensors - Comparative study of different rotary position sensors for electrical machines used in an hybrid electric vehicle application,“ Master thesis, Lund University, Department of Industrial Electrical Engineering and Automation, Sweden, 08/2011.
- [5] H. Seon-Hwan, K. Young-Hwa, K. Jang-Mok und O. Jin-Seok, „Compensation of Position Error due to Amplitude Imbalance in Resolver Signals,“ Journal of Power Electronics, Pusan, 09/2009.
- [6] J. O. Krah und M. Albers, „Resolver Auswertung über FPGA mit Delta Sigma Technologie,“ KOLLMORGEN Europe GmbH, [Online]. Available: www.wiki-kollmorgen.eu. [Zugriff am 02.09.2015].
- [7] K. Siebertz, D. van Bebber und T. Hochkirchen, Statistische Versuchsplanung, Springer-Verlag Berlin, 2010, ISBN 978-3-642-05492-1.
- [8] S. Walch, „Entwicklung eines Prüfstandes für die Evaluierung von Hochdrehzahlsensorik von elektrischen Traktionsantrieben,“ Masterarbeit, Institut für Fahrzeugtechnik, Technische Universität Graz, Österreich, 2015.
- [9] Profibus Nutzerorganisation e.V. (PNO), „PROFINET Systembeschreibung - Technologie und Anwendung,“ Profibus Nutzerorganisation e.V. (PNO), Karlsruhe, 10/2014.
- [10] B. Heinrich, P. Linke und M. Glöckler, Grundlagen Automatisierung, Springer Vieweg, 2015, ISBN 978-3-658-05960-6.

- [11] HMS Industrial Networks GmbH, „www.feldbusse.de,“ [Online]. Available: <http://www.feldbusse.de/Profinet/profinet.shtml>. [Zugriff am 07.08.2015].
- [12] Siemens AG, „SINAMICS G/S: PROFINET Anbindung an LabVIEW,“ Siemens AG Industry Sector, Erlangen, 01/2015.
- [13] Profibus Nutzerorganisation e.V. (PNO), „PROFIdrive Systembeschreibung - Technologie und Anwendung,“ Profibus Nutzerorganisation e.V. (PNO), Karlsruhe, 02/2011.
- [14] Siemens AG, SINAMICS S120 Funktionshandbuch, Nürnberg: Siemens AG, 01/2013.
- [15] Siemens AG, SINAMICS S120/S150 Listenhandbuch, Nürnberg: Siemens AG, 01/2012.
- [16] OWIS GmbH, „Betriebsanleitung Positioniersteuerung PS10,“ OWIS GmbH, Staufen, 2009.
- [17] G. Küveler und D. Schwach, Informatik für Ingenieure und Naturwissenschaftler 2, Friedr. Vieweg & Sohn Verlag, 2007, ISBN 978-3-8348-0187-6.
- [18] M. Paulweber und K. Lebert, Mess- und Prüfstandstechnik - Antriebsstrangentwicklung, Hybridisierung, Elektrifizierung, Springer Vieweg, 2014, ISBN 978-3-658-04452-7.
- [19] ESPEC CORP., „Environmental Stress Chamber AR Series - Communication Function,“ ESPEC CORP., Osaka, 07/2012.
- [20] Voltcraft, „Communication Protocol for Programmable Power Supply,“ Voltcraft, Hirschau, 2008.
- [21] S. Randazzo, „com0com,“ [Online]. Available: com0com.sourceforge.net. [Zugriff am 03.09.2015].
- [22] DEWETRON Elektronische Messgeräte GmbH, „DEWESoft 7 - AK Command Interface,“ DEWETRON, Trbovlje, Version 3.2.
- [23] B. Mütterlein, Handbuch für die Programmierung mit LabVIEW, Spektrum Akademischer Verlag, 2009, ISBN 978-3-8274-2337-5.

A Anhang



Abb. A-1: Prüfstand zur Evaluierung rotierender Positionssensoren

A.1 Geräteverzeichnis

Tabelle A-1: Komponenten des elektrischen Antriebs

Komponente	Hersteller, Bezeichnung
Elektromotor	Perske, VS 60.11-2
Antriebssystem	Siemens, SINAMICS S120
Umrichter	Siemens, PM 340
Regel- und Kommunikationseinheit	Siemens, CU310/2 PN

Tabelle A-2: Komponenten des Positioniersystems

Achsenbezeichnung	Hersteller, Bezeichnung
φ (tilt)	OWIS, DMT 65-D25-HSM Drehmesstisch mit Drehteller
x-Achse	OWIS, LTM 60-50-HSM Präzisions-Lineartisch
y-Achse	OWIS, HVM 100-30-HSM Höhenverstellisch
z-Achse	OWIS, LTM 60-25-HSM Präzisions-Lineartisch
für jeweils eine Achse	OWIS, PS 10-SM Positioniersteuerung

A.2 Netzwerk- und Schnittstellenkonfiguration

Tabelle A-3: Netzwerkkonfiguration des elektrischen Antriebs

	LabVIEW-PC	Elektrischer Antrieb
IP-Adresse	192.168.0.1	192.168.0.10
Subnetz-Maske	255.255.255.0	255.255.255.0
PROFINET-Name	labview	s120

Tabelle A-4: COM-Schnittstellenbelegung

COM-Port	Anwendung
COM 1	Virtueller COM-Port: DEWESoft (Messdatenerfassung)
COM 2	Virtueller COM-Port: LabVIEW (Messdatenerfassung)
COM 5	RS-485: Klimaschrank (Temperatursystem)
COM 6	USB-Serial-Port: x-Achse (Positioniersystem)
COM 7	USB-Serial-Port: z-Achse (Positioniersystem)
COM 8	USB-Serial-Port: y-Achse (Positioniersystem)
COM 9	USB-Serial-Port: Phi/Tilt (Positioniersystem)
COM 10	RS-232: Labornetzteil (Spannungsversorgung)

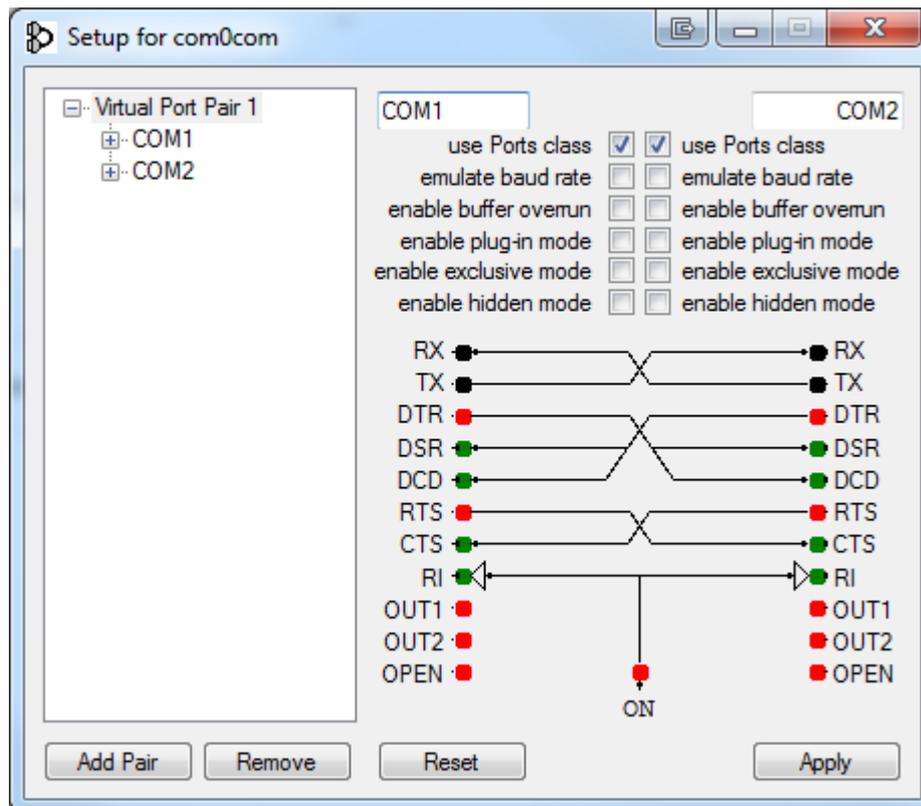


Abb. A-2: Konfiguration des Null-Modem Emulators

		PROFIdrive Ablastzeit siehe [1020.7]							
		Nicht geeignet für geberlose Vektorregelung. <4><4>[2440] [2450] automatisch							
Teleogramm	1	2	3	4	5	6	7	9	
Appl.-Class	1	1	1, 4	1, 4	4 DSC	4 DSC	3	3	
PZD1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1
PZD2	NSOLL_A NIST_A	NSOLL_B NIST_B	NSOLL_B NIST_B	NSOLL_B NIST_B	NSOLL_B NIST_B	NSOLL_B NIST_B	AKTSATZ	AKTSATZ	AKTSATZ
PZD3		STW2	ZSW2	ZSW2	ZSW2	ZSW2	STW2	ZSW2	
PZD4		G1_STW	G1_ZSW	G1_STW	G1_ZSW	G1_STW	G1_ZSW	MDL_TA	XIST_A
PZD5			G1_XIST1	G1_XIST1	G1_XIST1	G1_XIST1		MDL_VELOCITY	
PZD6			G1_XIST2	G1_XIST2	G1_XIST2	G1_XIST2			
PZD7				XERR	XERR	XERR			
PZD8				KPC	KPC	KPC		MDL_ACC	
PZD9								MDL_DEC	
PZD10				GZ_ZSW	GZ_ZSW	GZ_ZSW		MDL_MOD	
PZD11				GZ_XIST1	GZ_XIST1	GZ_XIST1			
PZD12				GZ_XIST2	GZ_XIST2	GZ_XIST2			
PZD13									
PZD14									
PZD15									
PZD16									
PZD17									
PZD18									
PZD19									
PZD20									
PZD21									
PZD22									
PZD23									
PZD24									
PZD25									
PZD26									
PZD27									
PZD28									
PZD29									
PZD30									
PZD31									
PZD32									

DO: ENC, SERVO, VECTOR		1	2	3	4	5	6	7	8
PROFIdrive – Standardtelegramme und Prozessdaten 1		fp_2415_54_deu.vsd		S120/S150/G130/G150		22.11.11V04.05.00		-2415 -	

<1>-Abhängig vom Antriebsobjekt können nur bestimmte Telegramme verwendet werden.
 <2>Beim Ändern von p0922 = 999 auf einen anderen Wert wird die Telegrammbelegung gemäß [2415] bis [2423] automatisch hergestellt.
 Beim Ändern von p0922 ungleich 999 auf p0922 = 999 bleibt die "alte" Telegrammbelegung gemäß [2415] bis [2423] erhalten.
 <3>-Die maximale PZD Anzahl ist abhängig vom Antriebsobjekttyp.
 <4>-Nur bei SINAMICS S120/S150.

☐ = Lagebersignal

Abb. A-3: PROFIdrive Standardtelegramme und Prozessdaten 1 [15]

A.3 Implementierung in Matlab — Erstellung der Versuchspläne

```

%% Create TestTable Samples
%% POSE_Auto, Dominik Mähr 07/2015

param_max = 7; % Gesamtanzahl der wählbaren Parameter
min_fakt = [2 1 1 0 0 0];
anz_fakt = sum(param_selection);

% Überprüfe ob für die jeweilige Entwurfsmethode genug variable Parameter
% gewählt wurden:
if anz_fakt > min_fakt(design_method+1)
    switch design_method %#ok<*ALIGN>
        case 0, % Box Behnken
            Samples_norm = bbdesign(anz_fakt, 'center', N);
        case 1, % Central Composite Faced
            Samples_norm = ccdesign(anz_fakt, 'type', 'faced', 'center', N);
        case 2, % Central Composite Inscribed
            Samples_norm =
ccdesign(anz_fakt, 'type', 'inscribed', 'center', N);
        case 3, % Full Factor
            Samples_norm = [fullfact(3*ones(1,anz_fakt))-2;
zeros(N,anz_fakt)];
        case 4, % Latin Hypercube
            Samples_norm = lhsdesign(N,anz_fakt)*2-1;
        case 5, % Pseudo Random
            Samples_norm = rand(N,anz_fakt)*2-1;
    end

% Sortiere die erzeugten Samples nach der Temperatur
if param_selection(6) == 1
    if param_selection(7) == 1
        Samples_norm = sortrows(Samples_norm,anz_fakt-1);
    else
        Samples_norm = sortrows(Samples_norm,anz_fakt);
    end
end

% Erzeuge die absoluten Samples (aus den normierten)
switch design_method
    case {0,1,2,3},
        Samples_abs = ones(length(Samples_norm),1)*parameter_const;
        i_max = length(Samples_norm);
    case {4,5},
        Samples_abs = ones(N,1)*parameter_const;
        i_max = N;
end

k = 1;
for j = 1:param_max
    if param_selection(j) == 1
        max_value = parameter_var(2,j);
        mean_value = sum(parameter_var(:,j))/2;
        for i = 1:i_max
            Samples_abs(i,j) = Samples_norm(i,k)*(max_value-
mean_value)+mean_value;
        end
        k = k+1;
    end
end
end

```

```
% Falls zu wenig variable Parameter gewählt wurden, wird der konstante
% Parametervektor ausgegeben:
else
    Samples_abs = parameter_const;
end

% Runden der Parameter: OWIS, Drehzahl und Spannung auf 2 Nachkommastelle
%                               Temperatur auf 1 Nachkommastellen
Samples_abs(:, [1:5 7]) = round(Samples_abs(:, [1:5 7]).*100)./100;
Samples_abs(:, 6) = round(Samples_abs(:, 6).*10)./10;

% Für number of runs > 1: versuchsplan entsprechend duplizieren
for l = 1:runs-1
    Samples_abs = [Samples_abs; Samples_abs];
    Samples_norm = [Samples_norm; Samples_norm];
end
```