

# Analyse von Verfahren für roboterunterstützte Fügeprozesse

Masterarbeit  
zur Erlangung des akademischen Grades

**Master of Science in Engineering (MSc)**

Fachhochschule Vorarlberg  
Master in Mechatronics

Vorgelegt von  
Georg Malin  
Dornbirn, 06.08.2021

Betreut von  
Prof. (FH) Dipl.-Ing. Robert Amann

Durchgeführt bei  
thyssenkrupp Presta AG, Eschen - Liechtenstein

# Kurzreferat

## **Analyse von Verfahren für roboterunterstützte Fügeprozesse**

Die vorliegende Masterarbeit befasst sich mit der Analyse von Unterstützungsverfahren, um mithilfe eines Roboters einen Fügeprozess kombiniert mit einer Fügeachse durchführen zu können. Durch das Führen der Bauteile durch den Roboter während des Fügevorgangs entsteht eine Überbestimmung zwischen Fügeachse und Roboter. Die auftretenden Kräfte und Momente, die dadurch auf den Roboter wirken, sind zu untersuchen. Mithilfe der Analysen soll ermittelt werden, ob der Roboter während der Fügeprozesse innerhalb seiner Leistungsgrenzen betrieben wird und ob eine ausreichend genaue Kompensation der Kräfte möglich ist. Ein dazu erstellter Versuchsaufbau soll diese Analysen ermöglichen. Nach dem Erarbeiten von Grundlagen der Roboterregelung wird speziell auf die, bei diesem Versuchsaufbau zu Verfügung stehenden, Unterstützungsverfahren des Roboters eingegangen. Die Eigenschaften und Anwendung des jeweiligen Verfahrens werden beleuchtet und dargestellt. Dem folgt eine Übersicht der geplanten Fügeversuche, um das Verhalten des Roboters während der Fügeprozesse zu analysieren. Die Auswertung einer durchgeführten Messsystemanalyse und der Fügeversuche stellt die Verwendbarkeit und die erreichbaren Toleranzen sowie die Stärken und Schwächen der Unterstützungsverfahren dar.

# **Abstract**

## **Analysis of Procedures for Robot Assisted Mating Processes**

The master thesis at hand deals with the analysis of support procedures in order to conduct a mating process – combined with a mating axis – with the aid of a robot. Due to the robot guiding the components during the mating procedure, an over-determined situation occurs. The resulting forces and torques, which are put on the robot, are to be analysed. With the help of that data it is determined whether the robot works within its performance limits and whether it is possible to get an accurate enough compensation of the forces. A specifically created test setup is used to enable the analysis. After working out basics of robot controlling, support techniques of the robot are highlighted; specifically those at hand in that test setup. The features and application of the respective procedure are displayed and presented. This is followed by an overview of the planned mating attempts in order to analyse the behaviour of the robot during mating processes. The evaluation of a carried out measurement system analysis and the mating attempts shows the applicability, attainable tolerances and strengths and weaknesses of the support techniques.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VIII</b>
<b>Abkürzungsverzeichnis</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation der Arbeit . . . . .	1
1.2 Zielsetzung und Abgrenzung . . . . .	2
1.3 Vorgehensweise . . . . .	3
<b>2 Methode</b>	<b>5</b>
2.1 Versuchsaufbau . . . . .	5
2.1.1 Komponenten und ihre Anordnung . . . . .	5
2.1.2 Netzwerkaufbau . . . . .	8
2.1.3 Datengenerierung und Speicherung . . . . .	10
2.2 Grundlegende Elemente der Robotik . . . . .	11
2.2.1 Jacobimatrix . . . . .	11
2.2.2 Singularitäten . . . . .	13
2.2.3 Regelung von Robotern . . . . .	13
2.3 Funktionsprinzipien der Unterstützungsverfahren des Roboters .	17
2.3.1 Softmove . . . . .	17
2.3.2 Force Control . . . . .	19
2.3.3 Sensor Synchronization . . . . .	21
2.3.4 Externally Guided Motion . . . . .	24
2.4 Fügeoperationen zur Beurteilung der Prozessqualität . . . . .	27
2.4.1 Einfache Versuche . . . . .	28
2.4.2 Federpaket . . . . .	32
2.4.3 Lager fügen . . . . .	33
2.4.4 Getriebegehäuse mit Welle auf Deckel fügen . . . . .	34

<b>3</b>	<b>Durchführung der Versuche und Ergebnisse</b>	<b>37</b>
3.1	Verifizierung der Messwerte . . . . .	37
3.1.1	Linearität Kistler-Kraftsensoren in der Fügeachse . . . . .	37
3.1.2	Linearität des oberen Kistler-Kraftsensors in der Fügeachse und des Roboter-Kraftsensors . . . . .	38
3.1.3	Beurteilung der Messwerte . . . . .	39
3.2	Durchführung der Fügeversuche . . . . .	40
3.2.1	Fügeversuche mit dem Federpaket . . . . .	42
3.2.2	Fügeversuche Lager einpressen . . . . .	45
3.2.3	Fügeversuche Getriebegehäuse mit Welle auf Deckel fügen	48
<b>4</b>	<b>Interpretation</b>	<b>52</b>
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>56</b>
	<b>Literaturverzeichnis</b>	<b>58</b>
	<b>Eidesstattliche Erklärung</b>	
	<b>Anhang</b>	

# Abbildungsverzeichnis

1.1	Lenksäule und Lenkwelle CEPS . . . . .	2
1.2	Schema der doppelten Kraftmessung . . . . .	3
2.1	Übersicht Versuchsaufbau . . . . .	6
2.2	Netzwerktopologie des Versuchsaufbaus . . . . .	9
2.3	EtherCAT: Standard-Ethernet-Telegramm . . . . .	9
2.4	Profinet Priorisierung . . . . .	10
2.5	Unterscheidung interne und externe Regelung . . . . .	14
2.6	Hybride Kraft- / Lageregelung . . . . .	15
2.7	Dämpfung und Tiefpassfilter . . . . .	21
2.8	Überblick Sensor Synchronization . . . . .	22
2.9	EGM Regelkreis . . . . .	25
2.10	Konfiguration Linearitätsprüfung Kistler Kraftsensoren . . . . .	29
2.11	Grundlegender Ablauf Fügeprozess . . . . .	30
2.12	Baugruppe Federpaket . . . . .	32
2.13	Fügeablauf Lager einpressen . . . . .	34
2.14	Fügeablauf Getriebegehäuse auf Deckel fügen mit Welle auf Lager schieben . . . . .	35
2.15	Robotergreifer mit Aufnahme für Getriebegehäuse und Welle . . . . .	36
3.1	Mittelwerte der Kraftstufen für die Linearitätsprüfung des oberen Kistler-Kraftsensors in Solara . . . . .	38
3.2	Mittelwerte der Kraftstufen für die Linearitätsprüfung des Roboter-Kraftsensors in Solara . . . . .	39
3.3	Ausschnitt von Messwerten in einer CSV Datei . . . . .	40
3.4	Fügekraftverläufe beim Federpaket mit FC . . . . .	42
3.5	Abgeleitete Kräfte beim Federpaket mit FC . . . . .	43
3.6	Abgeleitete Kräfte beim Federpaket mit SM . . . . .	44
3.7	Abgeleitete Drehmomente beim Federpaket mit SM . . . . .	45
3.8	Fügekraftverläufe beim Lager einpressen mit FC . . . . .	46

3.9	Abgeleitete Kräfte beim Lager einpressen mit FC . . . . .	46
3.10	Abgeleitete Kräfte beim Lager einpressen mit SM . . . . .	47
3.11	Fügekraftverläufe beim Getriebegehäuse mit Welle auf Deckel fügen mit FC . . . . .	48
3.12	Abgeleitete Kräfte beim Getriebegehäuse mit Welle auf Deckel fügen mit FC . . . . .	49
3.13	Fügekraftverläufe beim Getriebegehäuse mit Welle auf Deckel fügen mit SM . . . . .	50
3.14	Abgeleitete Kräfte beim Getriebegehäuse mit Welle auf Deckel fügen mit SM . . . . .	51

# Tabellenverzeichnis

2.1	Reihenfolge Datenspeicherung . . . . .	11
2.2	Roboter Werkzeugdaten . . . . .	27
3.1	Durchgeführte Versuche mit Parameter . . . . .	41

# Abkürzungsverzeichnis

<b>CEPS</b>	Column Electric Power Steering - Lenksäule mit Lenkungsunterstützung direkt in der Lenksäule
<b>MSA</b>	Messsystemanalyse
<b>SPS</b>	Speicher Programmierbare Steuerung
<b>ST</b>	Strukturierter Text
<b>DOF</b>	Degrees of Freedom - Freiheitsgrade
<b>UDP</b>	User Datagram Protocol
<b>TCP</b>	Transmission Control Protocol
<b>CSV</b>	Comma-Separated Values
<b>SM</b>	Softmove
<b>FC</b>	Force Control
<b>EGM</b>	Externally Guided Motion
<b>(Roboter)TCP</b>	Tool Center Point
<b>IE</b>	Industrielles Ethernet

# 1 Einleitung

## 1.1 Motivation der Arbeit

Im modernen Weltmarkt müssen neue Produkte in immer kürzerer Zeit entwickelt und ihre Produktion industrialisiert werden, damit diese sich erfolgreich etablieren können. Ebenso sollen die Produkte rasch an sich verändernde Marktbedingungen angepasst werden können. Für die Produktion bedeutet dies, dass die Produktionsabläufe und die Produktionsmaschinen flexibel sein müssen, um sich diesen Rahmenbedingungen unter Berücksichtigung der Kosten anpassen zu können. Bisher wurden die Montageanlagen bei thyssenkrupp Presta AG meist in Hinblick auf eine kleine Zykluszeit und für ein eingeschränktes Typenspektrum an Produkten konzipiert. Roboter werden dazu hauptsächlich für das Beladen und Entladen von Bauteilen verwendet. Für Fertigungsschritte wie z.B. Schraub- oder Fügeprozesse benötigt es weitere Werkzeuge und Aufnahmen. Dieser Aufbau führt zu schnellen und stabilen Produktionsmaschinen, jedoch bedeutet dies ebenfalls zeit- und kostenintensive Umbauten, sollten sich das Produkt oder Komponenten ändern. Die thyssenkrupp Presta AG in Eschen, Liechtenstein, fertigt von massivumgeformten Komponenten für die Industrie bis zu kompletten Lenksystemen ein breites Produktspektrum. Speziell bei der Fertigung von elektrisch unterstützten Lenkungssystemen nimmt die Komplexität und Vielfalt der unterschiedlichen Produkte hinsichtlich der Abmessungen und Formen zu. Als Beispiel sind dabei Lenksäulen bei denen die elektrische Lenkunterstützung direkt in der Lenksäule integriert ist - Column Electric Power Steering (CEPS) - zu nennen (Abbildung 1.1). Um dieser Vielfalt mit flexiblen Maschinenkonzepten zu begegnen, sollen reine Handlingsroboter in die Prozesse eingebunden werden. Dadurch lassen sich nicht nur Werkzeugkosten für Aufnahmen und Teilehandling in den Prozessmodulen einsparen, sondern auch die Integrationszeiten für neue oder abgeänderte Produkte verkürzen. In weiterer Folge sinken die Rüstzeiten bei den Anlagen, da weniger Werkzeuge zu tauschen sind.



Abbildung 1.1: Lenksäule mit Lenkwelle, Lenkungsunterstützung in der Lenksäule (CEPS)

Quelle: thyssenkrupp Presta AG

## 1.2 Zielsetzung und Abgrenzung

In dieser Arbeit wird die Integration eines Roboters in einen Fügeprozess mit einer elektro-mechanischen Fügeachse untersucht. Die zu fügenden Bauteile werden dabei von dem Roboter gehalten und geführt. Die Bewegung des Roboters soll sich dabei der Fügeachse durch eine geeignete Nachführung anpassen. Diese Bewegung des Roboters stellt speziell während der Fügeoperation, wenn die Bauteile gefügt werden und die Fügekraft durch die Achse eingeleitet wird, eine regelungstechnische Herausforderung dar. Die Bewegung des Roboters ist dadurch erzwungen, wodurch sich eine klassische Bahnsteuerung des Roboters nicht eignet [17, S.161]. Dabei ist zu untersuchen, ob sich diese Nachführung grundsätzlich realisieren lässt. Weiters muss untersucht werden, ob die Genauigkeit der Nachführung ausreicht, um einen stabilen Fügeprozess nachzuweisen und ob die über den Roboter abgeleiteten Kräfte gering genug sind, damit dieser innerhalb seiner Leistungsgrenzen betrieben werden kann. Das Hauptaugenmerk wird dabei auf die Interaktion zwischen Roboter und Fügeachse gerichtet und darauf, welche Technologien zur Nachführung geeignet sind. Die Entwicklung der jeweiligen Fügeprozesse unterliegt nicht dieser Arbeit.

## 1.3 Vorgehensweise

Um die Zielsetzung in Abschnitt 1.2 zu erreichen, wird ein Versuchsaufbau erstellt, um das Verhalten des Roboters zu analysieren. Ein Teilbereich der Arbeit betrachtet Kräfte, die durch das Führen der Bauteile mittels des Roboters während des Fügens über denselben abgeleitet werden. Dazu wird eine Überlegung aus einer vorangegangenen Bachelorarbeit [9] übernommen. Um die abgeleiteten Kräfte messen zu können wird einerseits ein Kraftsensor zwischen Roboterflansch und Werkzeug verwendet, andererseits werden zwei weitere Kraftmess-elemente jeweils an der Ober- und Unterseite der Fügeachse angebracht. Der schematische Aufbau ist in Abbildung 1.2 dargestellt. Damit lässt sich die Genauigkeit der Kraftmessung des Roboters nachweisen und es kann untersucht werden, ob weitere Einflüsse während des Fügevorganges auftreten.

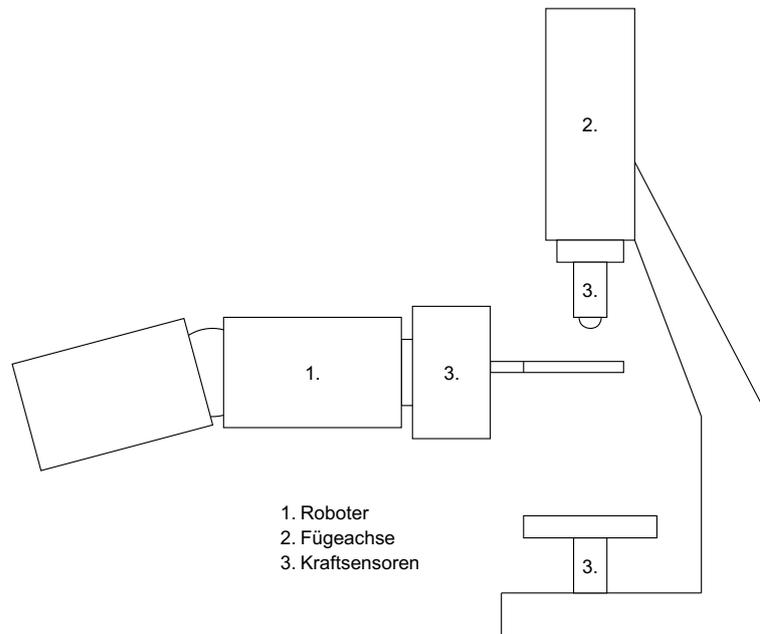


Abbildung 1.2: Schema der doppelten Kraftmessung  
Quelle: Ausarbeitung des Verfassers in Anlehnung an [9]

Um die Qualität und Vergleichbarkeit der beiden Kraftmessungen zu bestimmen, sind zuerst einfache Versuche mit Roboter und Fügeachse vorgesehen, ohne Bauteile zu fügen. In weiterer Folge stehen verschiedene Produktgruppen zur Verfügung, um die Ergebnisse der einfachen Versuche anhand realer Fügebedingungen zu verifizieren.

Um die Qualität eines Fügeprozesses zu beurteilen, wird oft die dazu benötigte Fügekraft herangezogen. Je nach Art der Bauteile ist bei der Fügekraft eine bestimmte Toleranz einzuhalten. Wird aber eines der Bauteile mit einem Roboter während dem Fügen geführt und Kräfte über denselben abgeleitet, verfälscht dies das Ergebnis des Fügeprozesses. Deshalb wird insbesondere betrachtet, ob sich eine voraussagbare und gleichmässige Kompensation realisieren lässt. Die Analyse, ob eine solche Fähigkeit vorliegt, erfolgt anhand der Ergebnisse der Fügeversuche und mithilfe einer Messsystemanalyse (MSA) der Messketten. Bei einer MSA wird überprüft, ob die gemessenen Werte die Realität, den wahren Wert, in ausreichend sicherem Maße wiedergeben [15].

# 2 Methode

## 2.1 Versuchsaufbau

Als Basis für die Erstellung des Versuchsaufbaus dient der Versuchsaufbau der vorangegangenen Bachelorarbeit [9]. Die wesentlichen Änderungen betreffen das Robotersystem, welches von der Marke Fanuc auf ABB umgestellt wird. Der Kraftsensor auf dem Roboter kommt ebenfalls von der Marke ABB. Außerdem kommt ein anderer SPS (Speicherprogrammierbare Steuerung) Rechner zum Einsatz.

### 2.1.1 Komponenten und ihre Anordnung

Übersicht der verwendeten Komponenten:

- Beckhoff SPS C6930 mit Eingangskarten und Schnittstellen
- ABB Roboter IRB 2600-20/1.65
- ABB 6-DOF<sup>1</sup> Kraftsensor 660 am Flansch des Roboters
- Test- und Fügwerkzeuge am Flansch des Kraftsensors
- Promess elektro-mechanische Fügeachse bis 60 kN
- Kistler Quarz Kraftsensoren 9321BU und 9331B
- Kistler Ladungsverstärker 5074A
- Heidenhain Längenmesslineal LC485 ML 320 mm
- Lumberg Sensor/Aktor Ein-/Ausgangsverteiler Box ASBS 8/LED 12-4
- Aventics 5/3 Wege Pneumatikventile

---

<sup>1</sup>DOF - Degrees of Freedom

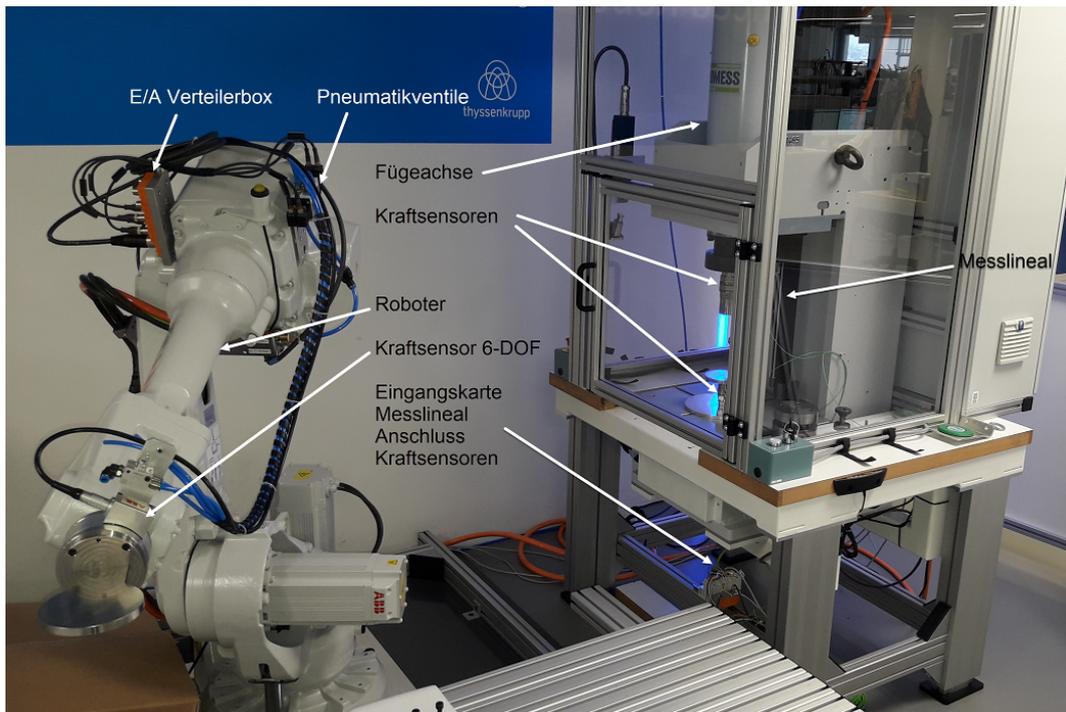


Abbildung 2.1: Übersicht Versuchsaufbau  
Quelle: Ausarbeitung des Verfassers

### Beckhoff SPS C6930

Die verwendete Beckhoff C6930 ist als Industrie-PC eine Kombination aus einem PC welcher mit Windows Betriebssystem ausgestattet ist und einer SPS für Automatisierungsaufgaben. Für die Programmierung wird die Software Twin-CAT 3 ebenfalls von Beckhoff verwendet [6]. Die Programmierung des Ablaufs erfolgt in einer Schrittkette, welche in Strukturiertem Text (ST) geschrieben wird. An die C6930 sind folgende Eingangs- und Ausgangsklemmen angeschlossen:

- EK1100 EtherCAT-Koppler 3x
- EL6631 Profinetklemme für die Verbindung zum ABB Roboter
- EL6631 Profinetklemme für die Verbindung zur Promess Fügeachse
- EL1014 Digitale Eingänge für Bedienelemente am Schaltschrank
- EL2004 Digitale Ausgänge für Visualisierungselemente am Schaltschrank

- EL5032 2-Kanal-EnDat-2.2-Interface für den Anschluss des Heidenhain Messlineals
- EL2521-0024 Encoder Simulationsklemme

### **ABB Roboter IRB 2600-20/1.65**

Der ABB IRB 2600 Roboter ist ein Industrieroboter, welcher eine Handlingskapazität von 20 kg und eine Reichweite von 1,65 m besitzt (siehe Datenblatt in Anhang A.1). Aus dieser Handlingskapazität ergibt sich:

$$F_h = m_h * g = 20 \text{ kg} * 9,81 \text{ m/s} = 196,2 \text{ N} \quad (2.1)$$

Diese Kraft  $F_h$  dient als Grenzwert, welche über den Roboter während des Fügevorganges abgeleitet werden darf, um Beschädigungen am Roboter zu verhindern. Weiters sind im Datenblatt Drehmomente um die vierte und fünfte Achse des Roboters von  $M_{4,5} = 36,3 \text{ N m}$  und um die sechste Achse von  $M_6 = 16 \text{ N m}$  angegeben.

### **ABB 6-DOF Kraftsensor 660**

Der Kraftsensor, welcher am Flansch des Roboters angebracht wird, ist ein 6-DOF Sensor und kann Kräfte in drei Freiheitsgraden der Translation  $F_x, F_y, F_z$  und Momente in drei Freiheitsgraden der Rotation  $T_x, T_y, T_z$  messen. Der Sensor hat ein eigenes Koordinatensystem, welches durch die Robotersteuerung entsprechend in ein abweichendes Koordinatensystem umgerechnet werden kann [3].

### **Test- und Fügwerkzeuge am Flansch des Kraftsensors**

Für die einfachen Versuche wird eine runde Stahlplatte in der Form eines Tischtennisschlägers an dem Kraftsensor geschraubt. Weitere Fügwerkzeuge werden in Abschnitt 2.4 beschrieben.

### **Promess elektro-mechanische Fügeachse**

Als Fügeachse wird eine elektro-mechanische Servofügeachse der Marke Promess verwendet. Mit dieser Achse sind Kraft-Weg überwachte Fügeprozesse umsetzbar und sie kann eine Nennkraft von 60 kN bei einer Nenngeschwindigkeit von 240 mm/s aufbringen. Der verbaute Kugelgewinde-Antrieb ermöglicht eine Positioniergenauigkeit von 0,01 mm [14].

### **Kistler Quarz Kraftsensoren**

Die Kistler Kraftsensoren sind 1-DOF Kraftsensoren und sind mit einem Quarzkristall ausgestattet, welcher elektrische Ladung exakt proportional zur Belastung abgibt. Die Messkette besteht aus Sensor, Kabel und Ladungsverstärker, welcher die Messsignale an die SPS sendet. Die Genauigkeit beträgt dabei etwa 2 % des eingestellten Messbereichs [11].

### **Kistler Ladungsverstärker**

Der Ladungsverstärker wandelt die eingehende Ladung des Kraftsensors für die SPS in einen Gleitkommamesswert um und sendet diesen über ein industrielles Ethernet Netzwerk. Dabei kann zwischen verschiedenen Messbereichen gewählt werden. Damit wird die Präzision im jeweiligen Bereich erhöht (siehe Anhang A.2).

### **Heidenhain Längenmesslineal**

Das Heidenhain Längenmesslineal LC485 ML 320 wird für die Positionsmessung des Fügwerkzeuges verwendet. Es besitzt einen Glasmassstab, welcher eine Auflösung von  $\pm 5 \mu\text{m}$  ermöglicht. Als Datenschnittstelle wird die EnDat2.2 Schnittstelle verwendet (siehe Anhang A.3).

## **2.1.2 Netzwerkaufbau**

Wie in Abbildung 2.2 ersichtlich, stellt der SPS-Rechner mit einem Switch das zentrale Element des Versuchsaufbaus dar. Auf diesem läuft das Programm ab und die angeschlossenen Komponenten werden durch Befehle gesteuert. Als Basisnetzwerk dient EtherCAT. EtherCAT ist eine Echtzeit-Ethernet-Technologie, auch Industrielles Ethernet (IE), welche sich für weiche und harte Echtzeitanforderungen in der Automatisierungs- und Messtechnik eignet. Von weicher Echtzeitbedingung wird gesprochen, wenn Aktualisierungszeiten von ca. 10 ms bei zyklischer Datenübertragung eingehalten werden, von harter Echtzeitbedingung bei Aktualisierungszeiten von ca. 1 ms und einer Taktgenauigkeit bis auf  $1 \mu\text{s}$  [20, S.584]. Von einem EtherCAT-Master, hier die Beckhoff SPS, wird ein Telegramm ausgesendet, welches die angeschlossenen Geräte, auch Slaves genannt, durchläuft. Die Slaves tauschen während dem Durchlaufen die an sie adressierten Daten aus. Dies ermöglicht sehr kurze Zykluszeiten unter  $100 \mu\text{s}$  [8]. Ein EtherCAT Standard-Telegramm ist in Abbildung 2.3 ersichtlich. An das EtherCAT Netzwerk sind Buskoppler angeschlossen, welche die Verbindung

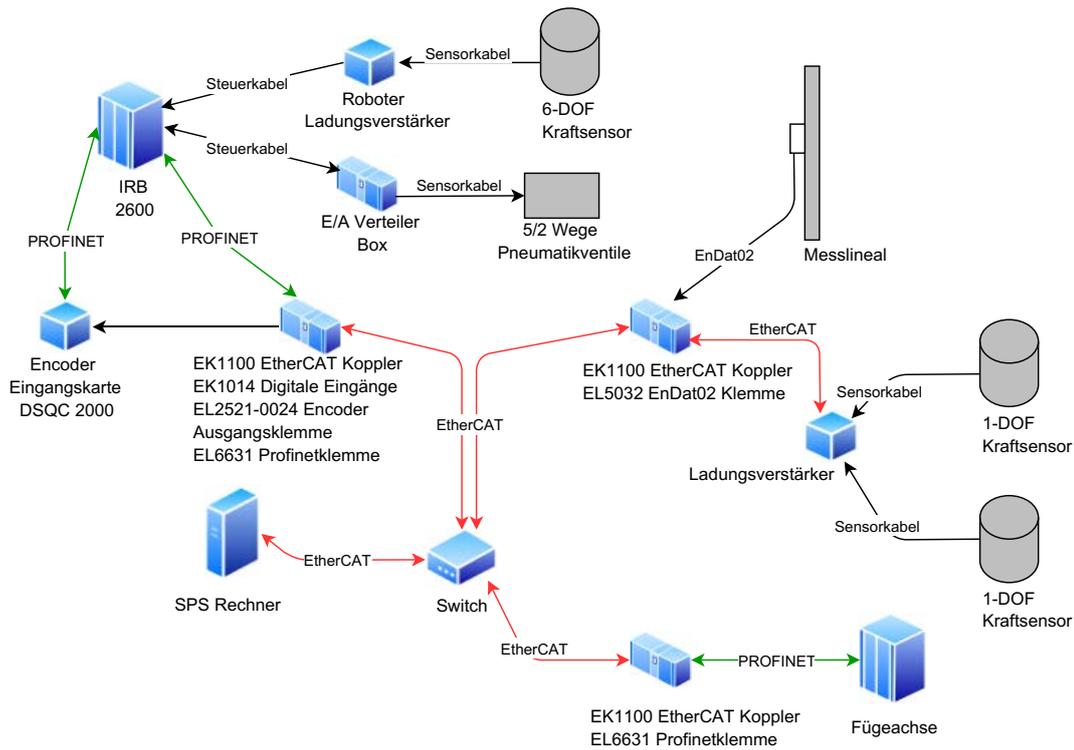


Abbildung 2.2: Netzwerktopologie des Versuchsaufbaus

Quelle: Ausarbeitung des Verfassers

zu PROFINET herstellen. PROFINET ist ebenfalls ein industrieller Ethernet Standard, welcher die weichen und harten Echtzeitanforderungen erfüllt. Ein wesentlicher Unterschied dabei ist die Art der Telegrammübertragung. Bei Profinet wird nicht ein einziges Telegramm durch alle Slaves gesendet, sondern mehrere Pakete. Diese sind dafür mit unterschiedlichen Prioritäten versehen. Priorisierte Pakete können andere Pakete überholen, somit können die Echtzeitbedingungen eingehalten werden. Dies wird mithilfe von Switches umgesetzt. Abbildung 2.4 zeigt eine solche Priorisierung in einem Switch, an dem 3 Geräte

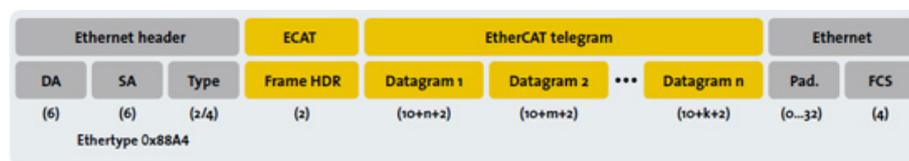


Abbildung 2.3: EtherCAT: Standard-Ethernet-Telegramm

Quelle: [8]

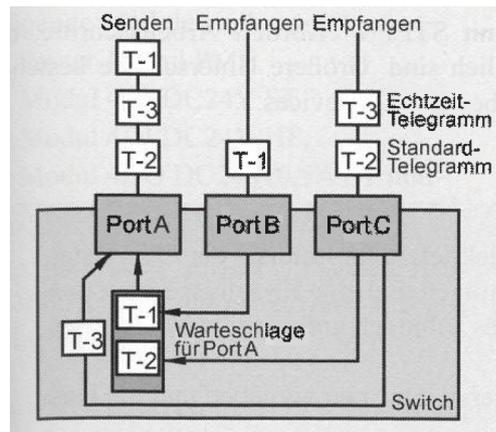


Abbildung 2.4: Profinet Priorisierung  
Quelle: [20, S.585]

angeschlossen sind. Über Profinet sind einerseits der Roboter und andererseits die Fügeachse mit der SPS verbunden. Innerhalb der Schaltschränke des Roboters und der Fügeachse sind Profinet Switche verbaut, um weitere Geräte in das Profinet Netzwerk anzubinden. So werden die Module für digitalen Ein- und Ausgänge und die DSQC 2000 Eingangskarte für Sensor Synchronisation (siehe Unterabschnitt 2.3.3) über Profinet im Roboter angeschlossen.

### 2.1.3 Datengenerierung und Speicherung

Um die Analyse der verschiedenen Fügeversuche zu ermöglichen, müssen Daten dafür gesammelt und gespeichert werden. Die dazu verwendeten Übertragungsmethoden sind, wie oben beschrieben, die industriellen Ethernet Varianten EtherCAT sowie Profinet und das UDP (User Datagram Protocol) Transportprotokoll. UDP ähnelt dem besser bekannten TCP (Transmission Control Protocol) welches einen fehlerfreien Softwarekanal zwischen zwei Anwenderprogrammen auf zwei Geräten herstellt. Das UDP Protokoll ist hingegen einfacher und prüft die fehlerfreie Übertragung nicht, ist dadurch aber schneller und besser geeignet für Echtzeitdaten als TCP [20, S.563-565].

In dieser Umsetzung wird generell IE zur Sammlung der Messwerte verwendet. Die SPS ermittelt die Messwerte aus den verschiedenen Geräten über die IE und speichert diese temporär in einem Array. Abschließend werden diese in eine CSV-Datei (Comma-Separated Values) geschrieben und somit dauerhaft auf der Festplatte des SPS-Rechners gespeichert. Die weitere Verarbeitung und

Auswertung der Daten erfolgt auf einem separaten PC mittels Matlab und Q-DAS Solara. Die Reihenfolge der Speicherung der Messwerte ist in Tabelle 2.1 ersichtlich.

CSV Position	Beschreibung
1	Laufende Nummer Messwert
2	Position Heidenhainlineal [mm]
3	Kraftwert Kistler 1 oben [N]
4	Kraftwert Kistler 2 unten [N]
5	Kraftwert in X ABB Sensor [N]
6	Kraftwert in Y ABB Sensor [N]
7	Kraftwert in Z ABB Sensor [N]
8	Drehmoment um X ABB Sensor [N]
9	Drehmoment um Y ABB Sensor [N]
10	Drehmoment um Z ABB Sensor [N]
11	Position in X ABB Roboter [mm]
12	Position in Y ABB Roboter [mm]
13	Position in Z ABB Roboter [mm]
14	Winkel um X ABB Roboter [deg]
15	Winkel um Y ABB Roboter [deg]
16	Winkel um Z ABB Roboter [deg]

Tabelle 2.1: Reihenfolge Datenspeicherung  
Quelle: Ausarbeitung des Verfassers

## 2.2 Grundlegende Elemente der Robotik

Um das Führen der Bauteile durch den Roboter während des Fügevorgangs zu ermöglichen, sind verschiedene Lösungsansätze vorhanden. Der erste Ansatz verwendet eine Nachführung auf Basis einer Kraftregelung. Der zweite Ansatz baut auf eine Nachführung auf Basis externen Positionsvorgaben, welche im Roboter für die Lageregelung verwendet werden. Dazu müssen jedoch erst einige grundlegende Elemente der Robotik näher betrachtet werden.

### 2.2.1 Jacobimatrix

Die Jacobimatrix generell besteht aus partiellen Ableitungen und ist mehrdimensional. Dabei stellt die Anzahl der Zeilen die Menge der Funktionen und

die Spalten die Anzahl der Variablen dar. In der Robotik werden dazu die Gelenkwinkel als die Variablen und die Gleichungen aus der Vorwärtskinematik als Funktionen, was die Anzahl der Freiheitsgrade darstellt, verwendet. Der für diese Arbeit verwendete Roboter besitzt 6 Achsen und 6 Freiheitsgrade, was zu einer quadratischen 6 x 6 Matrix führt. Die Jacobimatrix ermöglicht, dass aus den Gelenkwinkel und Gelenksgeschwindigkeiten die kartesischen Geschwindigkeiten errechnet werden können [7, S.149-150].

$$v = J(q)\dot{q} \quad (2.2)$$

Wobei  $q$  die Gelenkwinkel in Form eines 6 x 1 Vektors,  $\dot{q}$  die Gelenksgeschwindigkeiten ebenfalls als 6 x 1,  ${}^0J$  die Jacobimatrix und  ${}^0v$  den Vektor der jeweils drei kartesischen Linear- und Rotationsgeschwindigkeiten als 6 x 1 darstellen.  ${}^0J$  und  ${}^0v$  sind jeweils im selben Koordinatensystem hier in 0 für das Endeffektor Koordinatensystem. Dies bedeutet, dass die Gelenksgeschwindigkeiten und kartesischen Geschwindigkeiten durch eine lineare Transformation verknüpft sind. Allerdings hat diese Aussage nur infinitesimale Zeit Gültigkeit, da sich die Gelenkwinkel ändern und die Jacobimatrix dadurch neu berechnet werden muss. Analog dazu kann diese Beziehung auch in der Kraftdomäne beschrieben werden [7, S.156-157]. Es gilt:

$$\tau = J(q)^T F \quad (2.3)$$

Dabei stellt  $\tau$  ein 6 x 1 Vektor mit den Gelenkdrehmomenten,  $J(q)$  die Jacobimatrix und  $F$  ein 6 x 1 Vektor, der jeweils drei kartesischen Kräfte und Momente, welche am Endeffektor wirken, dar. In Fällen in denen die Jacobimatrix invertierbar ist (näheres dazu in Unterabschnitt 2.2.2), können aus den kartesischen Geschwindigkeiten die Gelenksgeschwindigkeiten berechnet werden.

$$\dot{q} = J^{-1}(q)v \quad (2.4)$$

Dementsprechend:

$$F = J^{-1}(q)\tau \quad (2.5)$$

Wie aus der Gleichung 2.5 hervor geht, können die am Endeffektor anliegenden Kräfte und Momente aus den an den Gelenken anliegenden Drehmomenten berechnet werden, solange die Jacobimatrix invertierbar ist.

### 2.2.2 Singularitäten

Eine quadratische Matrix ist invertierbar, solange diese regulär ist. Eine reguläre Jacobimatrix  $J$  liegt vor, wenn die Determinante  $\det J \neq 0$  ist. Ansonsten ist die Matrix singulär. In der Robotik sind Singularitäten jene Bereiche im Arbeitsraum, an denen die Jacobimatrix singulär wird. Diese treten an den Rändern des Arbeitsraumes auf, wenn der Roboter komplett gestreckt oder eingefahren ist. Eine weitere Konfiguration bei der Singularität auftritt ist, wenn Drehachsen mehrerer Gelenke gleich ausgerichtet sind. Dadurch verliert der Roboter mindestens einen Freiheitsgrad, in welchen er sich nicht mehr bewegen oder eine Kraft aufbringen kann [7, S.151].

Da beim Bilden einer Inversen Matrix die Determinante der Matrix im Nenner vorkommt und diese in der Nähe von Singularitäten gegen 0 geht, bedeutet dies, dass dieser Term gegen unendlich geht. In weiterer Folge nach Gleichung 2.4 und Gleichung 2.5 verursachen kleine Änderungen in den kartesischen Geschwindigkeiten hohe Gelenksgeschwindigkeiten und kleine Änderungen in den Gelenkdrehmomenten hohe Kräfte und Momente am Endeffektor.

### 2.2.3 Regelung von Robotern

Der Regelung des Roboters, wie der Endeffektor Bewegungen ausführt, kommt in dieser Arbeit eine besondere Bedeutung zu. Grundsätzlich lässt sich die Regelung eines Roboters in interne und externe Regelung unterteilen.

Unter **interner Regelung** wird das Beeinflussen der Gelenkwinkel mit Berücksichtigung der Sollgrößen und der Messgrößen verstanden. Abbildung 2.5 zeigt, dass diese Gelenkregelung die Gelenksollwerte  $q_s$  und die Ist-Messgrößen Gelenkwinkel  $q$ , Achsgeschwindigkeiten  $\dot{q}$  und Motorströme bzw. Motordrehmomente  $i/\tau$  zu den Stellgrößen  $U_s$ , welche die Antriebseinheiten ansteuern, verarbeitet. Die Gelenkregelung hat die Aufgabe, die Ist-Gelenkwinkel den sich zeitlich ändernden Soll-Gelenkwinkeln nachzuführen. Eine Gelenkregelung ist in jedem Industrieroboter in einer ähnlichen Form vorhanden. Durch die mechanische Verbindung der Gelenke beeinflussen sich diese gegenseitig. Es wird dabei von einer verkoppelten und nichtlinearen Regelstrecke gesprochen, was einer übergeordneten Koordination bedarf. Darauf wird hier jedoch nicht näher eingegangen. Weiterführende Informationen zu diesem Thema in [19, S.155ff].

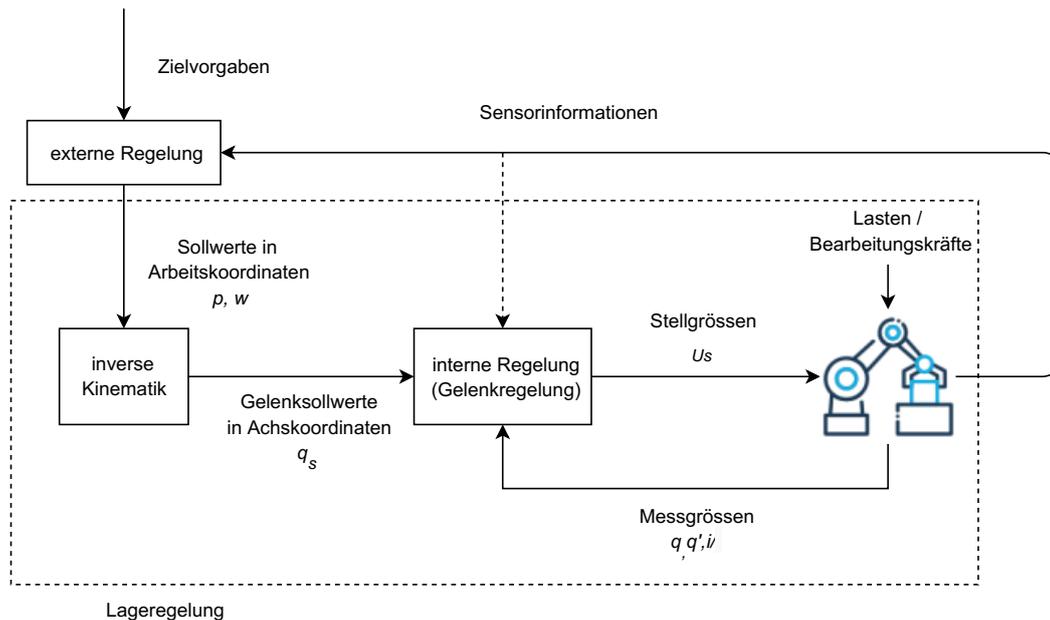


Abbildung 2.5: Unterscheidung interne und externe Regelung  
Quelle: Eigene Ausarbeitung in Anlehnung an [19, S.11/152]

Von einer **externen Regelung** wird gesprochen, wenn auf einer übergeordneten Ebene zur Gelenkregelung mithilfe zusätzlicher Informationen von Sensoren oder Messsystemen die Zielvorgaben zu Sollwerten verarbeitet werden. Diese externe Regelung kann entweder auf einer separaten Steuerung oder als Baustein in der Robotersteuerung ablaufen. Ziel dabei ist meist die Verbesserung der Qualität der Roboterbewegung, welche bei der Gelenkregelung durch biegen der Armteile bei großen Lasten oder hohen Beschleunigungen negativ beeinflusst wird vgl. [19].

Abgesehen von der vorhergehenden Unterteilung sind die Roboterbewegungen des Endeffektors auch in Lageregelung, Kraftregelung und hybride Kraft- / Lageregelung zu unterteilen.

Am weitesten verbreitet ist die **Lageregelung**, bei der der Roboter vorgegebene Positionen anfährt. Üblicherweise werden, wie in Abbildung 2.5 dargestellt, aus Sollwerten (Position  $p_s$  und Orientierung  $w_s$ ) in einem kartesischen Arbeitssystem durch Berechnung der inversen Kinematik die Gelenksollwerte für die Gelenkregelung erzeugt. Auftretende Kräfte und Momente werden dabei, sofern diese die Leistungsgrenzen des Roboters nicht übersteigen, als

Störgrößen kompensiert. Kommt der Roboter jedoch mit steifen Bauteilen in der Arbeitsumgebung in Berührung, steigen die Kräfte am Endeffektor stark an.

Treten solche Situationen auf, kann eine **Kraftregelung** angewendet werden, welche dafür sorgen soll, dass der Roboter definierte Kräfte und Momente auf den Arbeitsraum ausübt. Eine reine Kraftregelung ist nur möglich wenn der Roboter in allen Freiheitsgraden eingeschränkt ist, anderenfalls könnte er in freigegebene Richtungen dauerhaft beschleunigen. In den meisten Anwendungsfällen jedoch sind Roboter in einigen Freiheitsgraden frei und in einigen eingeschränkt vgl. [12, S.435].

Für solche Anwendungsfälle ist die **hybride Kraft- / Lageregelung** geeignet. Abbildung 2.6 stellt dar, dass eine hybride Kraft- / Lageregelung aus einem Lageregler und einem Kraftregler besteht. Welcher Regler zum Einsatz kommt, wird, je nachdem welche Koordinatenrichtung oder Drehung ein Kraft- oder Bewegungsfreiheitsgrad sein soll, durch die Auswahlmatrizen  $S$  und  $S'$  entschieden. Wobei  $S' = I - S$  gilt, wenn  $I$  die Einheitsmatrix darstellt.

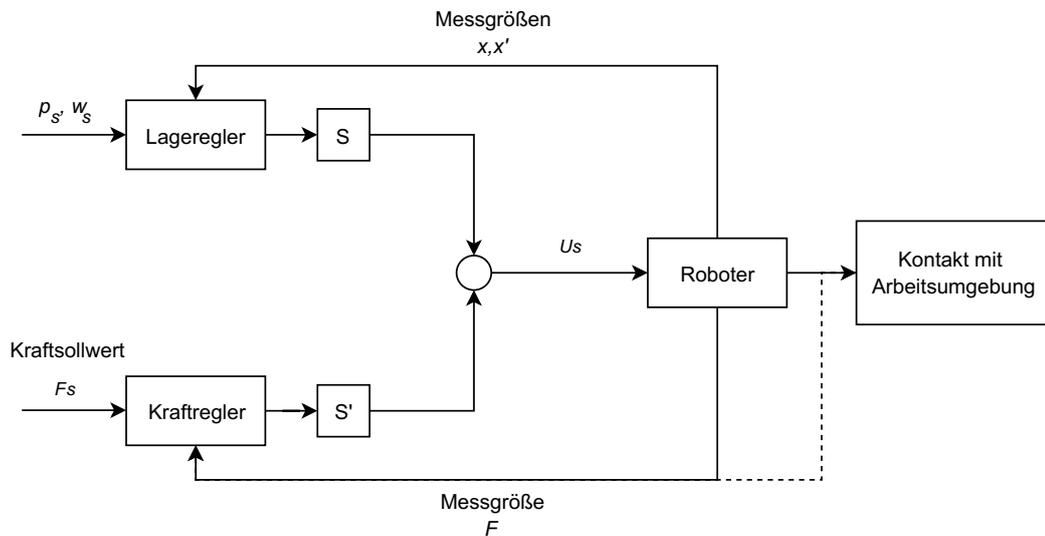


Abbildung 2.6: Hybride Kraft- / Lageregelung  
 Quelle: Eigene Ausarbeitung in Anlehnung an [19, S.201]

In Abbildung 2.6 ist ebenfalls ersichtlich, dass die Art der Rückmeldung bei der Kraftregelung unterschiedlich sein kann. Dies ist abhängig vom Konzept der Kraftregelung. Es wird dabei zwischen der Impedanz- oder Widerstandsre-

gelung (mit durchgezogener Linie) und der Admittanz- oder Nachgiebigkeitsregelung (mit strichlierter Linie) unterschieden.

Die aktive **Impedanzregelung** versucht das Verhalten des Roboters wie ein einfaches Feder-Masse-Dämpfer Modell nachzubilden. Ziel ist es, das Verhalten folgender Gleichung implementieren zu können:

$$M\ddot{x} + B\dot{x} + Kx = F_{ext} \quad (2.6)$$

Dabei ist  $x$  die Position und Orientierung im Arbeitsraum,  $M, B$  und  $K$  sind positive-definite, virtuelle, einstellbare Massen-, Dämpfungs- und Steifigkeitsmatrizen und  $F_{ext}$  stellt den Kraft- und Momentenvektor dar, der die Wechselwirkung des Roboters und der Arbeitsumgebung beschreibt. Wird Gleichung 2.6 in Gleichung 2.3 eingesetzt und mit dem dynamischen Modell des Roboters erweitert, erhält man folgende Regelungsgleichung für die Gelenkdrehmomente [12, S.444]:

$$\tau = J^T(q)(F_b - (M\ddot{x} + B\dot{x} + Kx)) \quad (2.7)$$

Wobei  $F_b$  das dynamische Modell des Roboters im Arbeitsraum darstellt [12, S.300] und  $\ddot{x}, \dot{x}$  und  $x$  direkt gemessen wird. Dabei handelt es sich um eine indirekte Kraftregelung, da sie ohne externe Sensorik auskommt und über Parameter eingestellt wird.

Da das genaue Ermitteln des dynamischen Modells des Roboters nicht so trivial ist, erarbeitete Salisbury [16] einen Denkansatz, der zu einem ähnlichen Verhalten des Roboters führen soll.

**Nachgiebigkeit durch Abschwächen der Positionsverstärkungsparameter.** Auch dieser Ansatz kommt dabei ohne externe Sensorik aus. Er formuliert dabei das Verhalten einer allgemeinen Feder mit sechs Freiheitsgraden mit

$$F = K_{px}\delta_x, \quad (2.8)$$

wobei  $K_{px}$  eine 6 x 6 Diagonalmatrix mit je drei translatorischen und drei rotatorischen Steifigkeitsparametern darstellt und  $\delta_x$  ein 6 x 1 Vektor der infinitesimalen, kartesischen Verschiebungen. Mithilfe der Eigenschaften der Jacobimatrix kann Gleichung 2.8 wie folgt erweitert werden.

$$\tau = J^T(q)K_{px}J(q)\delta_q \quad (2.9)$$

Kombiniert mit einer einfachen, gelenkbasierten Regelungsgleichung einer La-

geregelung stellt Salisbury die Ermittlung der benötigten Gelenkdrehmomente wie folgt auf.

$$\tau = J^T(q)K_{px}J(q)E + K_v\dot{E} \quad (2.10)$$

Dabei ist  $E$  die Abweichung der Sollgelenkwinkel zu den Istgelenkwinkel  $E = q_s - q$  und  $K_v$  eine Matrix mit konstanten Verstärkungsparametern.  $K_{px}$  stellt nun eine einstellbare Steifigkeit des Endeffektors im kartesischen Raum dar vgl. [7, S.333-334].

Die **Admittanzregelung** bedient sich hingegen eines Kraft-/Momentensensors, welcher  $F_{ext}$  direkt am Endeffektor misst und der Regelung zuführt. Um wieder das Verhalten von Gleichung 2.6 zu erreichen wird dieser gemessene  $F_{ext}$  in Gleichung 2.6 eingesetzt und erhält dadurch die benötigten Beschleunigungen am Endeffektor  $\ddot{x}_d$ .

$$\ddot{x}_d = M^{-1}(F_{ext} - B\dot{x} - Kx) \quad (2.11)$$

Aus den Beschleunigungen des Endeffektors und den gemessenen externen Kräften lassen sich, mithilfe der Gleichung 2.2, durch umformen die Gelenkbeschleunigungen  $\ddot{q}_d$  bestimmen, um das gewünschte Verhalten des Endeffektors zu gewährleisten [12, S.444].

## 2.3 Funktionsprinzipien der Unterstützungsverfahren des Roboters

Um die Nachführung des ABB Roboters bei der Versuchsdurchführung zu ermöglichen können nachfolgende Pakete als verwendbar erfasst werden. Bei Betrachtung der Eigenschaften der Pakete können diese mit den zuvor beschriebenen Regelungskonzepten in Verbindung gebracht werden.

### 2.3.1 Softmove

Softmove (SM) ist ein Software Paket, das den Ansatz der hybriden Kraft-/Lageregelung verfolgt. Es ist dabei dem Ansatz der Nachgiebigkeit durch abschwächen der Positionsverstärkungsparameter zuzuordnen, da es ohne zusätzliche Sensorik auskommt. Durch Einstellen der vorgegebenen Parameter ist es möglich, dass der Roboter in einer kartesischen Koordinatenrichtung die Nachgiebigkeit erhöht und, dass er sich wie eine mechanische Feder verhält je weiter er von der ursprünglichen Position entfernt ist [4]. Das Verhalten des

Roboters kann in dieser Koordinatenrichtung als Kraftregelung interpretiert werden. Je nachdem welche Koordinatenrichtung auf Kraftregelung geschaltet wird, berechnet die Robotersteuerung nach dem Prinzip von Gleichung 2.10 wie die Motorströme anzupassen sind, um diese Bewegung zu ermöglichen und in den anderen Richtungen die Lageregelung beizubehalten. Aufgrund dessen, dass dieses Paket ohne zusätzliche Sensorik auskommt, ist es kostengünstiger. Allerdings haben die robotereigene Reibung und die Stellung der Achsen einen nicht zu vernachlässigenden Einfluss auf das Verhalten des Roboters während der Bewegung. Wie in Gleichung 2.5 und Unterabschnitt 2.2.2 beschrieben, führen in der Nähe von Singularitäten kleine Änderungen der Drehmomente zu großen Änderungen der Kräfte und Momente am Endeffektor. In umgekehrter Weise bedeutet dies, dass der Roboter in diesen Situationen kaum oder nicht in zufriedenstellender Qualität auf geringe Kraftänderungen reagiert. Es kann zu einem Verhalten wie ein Stick-Slip Effekt kommen. Durch die geringe Änderung der Drehmomente aufgrund der externen Krafteinwirkungen hat die Reibung in den Gelenken und Getrieben einen stärkeren Einfluss auf das Verhalten des Roboters während aktiviertem Softmove.

Die Aktivierung des Pakets erfolgt mit folgendem Befehl [4]:

```
CSSAct [\RefFrame] [\RefOrient] SoftDir [\StiffnessNonSoftDir] [\Stiffness] [\AllowMove] [\Ramp]
```

Bedeutung der Parameter:

*[\RefFrame]*: Koordinatensystem, welches als Basis für den Befehl dient

*[\RefOrient]*: Orientierung des Koordinatensystems verändern

*SoftDir*: Kartesische Richtung, in welche die Steifigkeit verringert wird

*[\StiffnessNonSoftDir]*: Steifigkeit der verbleibenden kartesischen Richtungen

*[\Stiffness]*: Ausprägung des Feder Effekts

*[\Stiffness to Damping ratio]*: Dämpfungseffekt abhängig vom eingestellten Feder Effekt Stiffness

*[\AllowMove]*: Bewegungsinstruktionen können während aktiviertem SoftMove ausgeführt werden

*[\Ramp]*: Definiert, wie schnell SoftMove aktiviert wird

Die Parameter im Aktivierungsbefehl beeinflussen das Verhalten des Roboters während der Aktivierung und bei aktivem SoftMove. Zwei nicht aufgeführte Systemparameter gilt es zu erwähnen, welcher das Verhalten des Roboters während Softmove ebenfalls beeinflusst. Mit [\Dynamic to static fric ratio] ist es möglich auf ein sich änderndes Verhältnis zwischen statischer und dynamischer Reibung der Robotergelenke zu reagieren. Mit [\Stiffness to Damping ratio] kann ein Dämpfungseffekt abhängig vom eingestellten Feder Effekt Stiffness erzielt werden [4, S.17]. Der Parameter [\Stiffness] kann dabei mit einem Eintrag der in Gleichung 2.10 genannten Matrix  $K_{px}$  verglichen werden.

### 2.3.2 Force Control

Force Control (FC) ist ein Hard- und Software Paket, welches den Ansatz der Kraftregelung verfolgt. Es ist jedoch aufgrund der Eigenschaften der hybriden Kraft- / Lageregelung zuzuordnen. Die Kraftregelung bei Force Control ist dabei mit der Admittanzregelung zu vergleichen, da dieses Paket einen Kraftsensor erfordert. Durch diesen Kraftsensor kann auch von einer externen Regelung gesprochen werden, auch wenn diese innerhalb der Roboter Steuerung abläuft. Der in dieser Arbeit eingesetzte 6-DOF Kraftsensor ist direkt zwischen Roboterflansch und dem Werkzeug angeschraubt. Somit bekommt der Kraftregler, wie in Abbildung 2.6 mit der strichlierten Linie angedeutet, eine direkte Rückmeldung der aktuellen Kräfte und Momente am Endeffektor. Wie der Roboter auf diese Einflüsse reagieren soll kann in diesem Paket unterschiedlich eingestellt werden, dabei gibt es drei wesentliche Verhaltensweisen. Bei *FC Pressure* übt der Roboter eine konstante Kraft auf eine Oberfläche aus; er folgt einer Bahn mithilfe der Lageregelung. Um Unebenheiten auszugleichen wird die Bahn jedoch aufgrund der Kraftregelung verlassen, damit eine konstante Kraft auf die Oberfläche erhalten bleibt. Dies kann zielführend beim Polieren oder Entgraten sein. Bei *FC SpeedChange* wird der Bahngenauigkeit eine höhere Priorität zugeordnet. Dies ist beim Entgraten mit definierter Fasengröße oder beim Bohren notwendig. Der Roboter weicht bei dieser Einstellung nicht von der Bahn ab, jedoch reduziert er die Geschwindigkeit abhängig von gemessenen Kräften und Momenten oder aber auch durch ein externes Signal, beispielsweise einer auf dem Roboter montierten Bearbeitungsspindel. Bei *FC Assembly* kann der Roboter Montageprozesse vornehmen. Dabei werden die Eigenschaften der hybriden Kraft- / Lageregelung so verwendet, dass der Roboter in verschie-

denen Freiheitsgraden unterschiedliche Beweglichkeitsgrade besitzt. Dies kann von steifer Lageregelung bis nachgiebiger Kraftregelung gehen. Weiters können einige Vorgehensweisen verwendet werden wie Suchmuster, Federeffekt oder definierte Kräfte/Momente in Koordinatenrichtungen aufbringen. Für diese Arbeit kommt *FC Assembly* als geeignete Verhaltensweise in Frage.

Die Aktivierung des Pakets erfolgt mit folgendem Befehl [3]:

```
FCAct Tool [\WObj] [\ForceFrameRef] [\ForceFrameOrient] [\DampingTune]
```

Bedeutung der Parameter:

*Tool*: Daten des verwendeten Werkzeugs, Ursprung des Werkzeugs wird als Ursprung des FC Koordinatensystems verwendet

*[\WObj]*: Werkobjekt Koordinatensystem

*[\ForceFrameRef]*: Definiert, ob die Orientierung des FC Koordinatensystems über das Werkobjekt- oder das Werkzeugkoordinatensystem bestimmt wird

*[\ForceFrameOrient]*: Ermöglicht die Orientierung des FC Koordinatensystems zusätzlich zu verändern. Angabe der Orientierung in Quaternionen

*[\DampingTune]*: Verändert die Dämpfung des Roboters in Prozent der Systemparameter

Unter Dämpfung wird bei Force Control der Zwischenbereich von steifer Lageregelung und nachgiebiger Kraftregelung verstanden. Der Parameter bestimmt, wie viel Kraft oder Moment in die jeweilige Bewegungsrichtung aufgewendet werden muss, dass sich der Roboter mit einer Geschwindigkeit von  $v = 1 \text{ m/s}$  oder  $\omega = 1 \text{ rad/s}$  in diese Richtung bewegt. Grundsätzlich sind die Werte für drei translatorische und drei rotatorische Koordinatenrichtungen in den Systemparametern definiert. Diese sind jedoch im Aktivierungsbefehl oder mithilfe des Befehls *FCSetDampingTune* anpassbar.

Ebenfalls zu erwähnen ist, der im Aktivierungsbefehl nicht vorhandene, Systemparameter *LP-filter*. Das Kraft- und Momentsignal wird durch einen Tiefpassfilter bearbeitet um Rauschen im Signal zu glätten. Er wird in Abhängigkeit des verwendeten Roboters, der angewendeten Verhaltensweise und der Steifigkeit des Werkzeugs und der Arbeitsumgebung, welche sich in Kontakt mit dem Werkzeug befindet, eingestellt.

Abbildung 2.7 zeigt, wie sich die Geschwindigkeit des Roboter Endeffektor, welche der Lageregelung zugeführt wird, zusammensetzt.

Das gemessene Kraft-/Momentensignal wird von der Vorgabe abgezogen, anschließend durch den Dämpfungsparameter dividiert und nach der Filterung mittels Tiefpass ergibt sich die Geschwindigkeitsvorgabe.

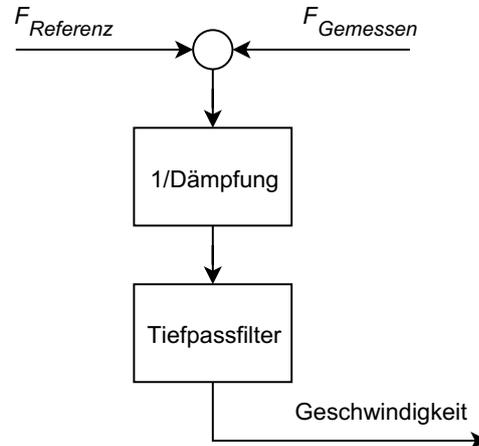


Abbildung 2.7: Dämpfung und Tiefpassfilter  
 Quelle: Eigene Ausarbeitung in Anlehnung an [3]

Bevor FC aktivierbar ist, muss die Routine *FCCalib* ausgeführt werden. Diese Routine berücksichtigt die Masse und den Schwerpunkt des angebaute Werkzeuges und führt in diesem Sinne eine Justierung der Kraftmessung durch. Es wird dadurch der Einfluss der Gravitation oder bestehende Offsets kompensiert. Das Ermitteln der benötigten Lastdaten des angebaute Werkzeuges ist mithilfe der Routine *FCLoadID* automatisiert möglich. Durch wechselndes Drehen der beiden nächstgelegenen Gelenke zum Endeffektor (Gelenke 5 und 6) liest die Steuerung an verschiedenen Stellen die Sensordaten aus und errechnet sich daraus Schwerpunkt und Masse des Werkzeuges inklusive Last. Dabei sind Schwenkbereich, die Anzahl der Messstellen und die Anzahl der Messwerte pro Messstelle, aus denen ein Mittelwert gebildet wird, einstellbar.

### 2.3.3 Sensor Synchronization

Sensor Synchronization auch Sensor Sync ist ein Hard- und Softwarepaket, das den Ansatz der Lageregelung verfolgt. Es ist der externen Regelung zuzuordnen, da ein zusätzlicher Sensor für die Position notwendig ist. Unter Sensor ist bei diesem Paket ein Encoder gemeint, der ein zweiphasiges Rechtecksignal

erzeugt. Die Phasen sollen dabei um  $90^\circ$  verschoben sein. Typischerweise können solche Encoder auf Antriebseinheiten platziert sein, welche Schutztüren oder einen Schlitten in einer Maschine bewegen. Abbildung 2.8 stellt den Encoder und weitere benötigte Komponenten dar. Die Encoder Schnittstelleneinheit wandelt die Rechtecksignale in Geschwindigkeitsmesswerte für den Roboter um. Die Lageregelung des Roboters berechnet mithilfe der Messwerte laufend die Interpolationspunkte für die Bewegung neu - mit der angepassten Geschwindigkeit. Die ursprünglich erstellte Bahn der Lageregelung bleibt jedoch dieselbe und es wird nur die Geschwindigkeit der Bewegung an das Encodersignal angepasst.

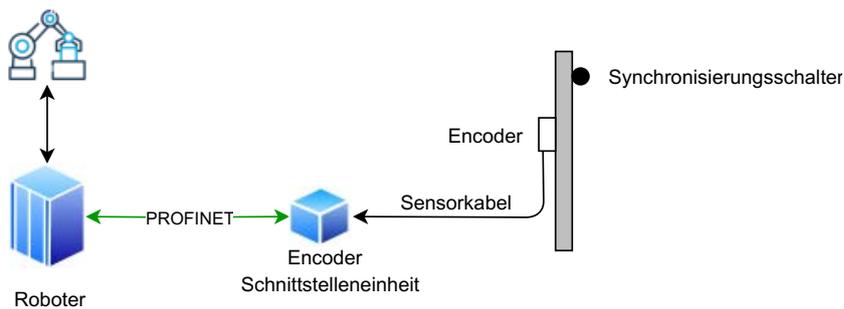


Abbildung 2.8: Überblick Sensor Synchronisation  
Quelle: Eigene Ausarbeitung in Anlehnung an [1, S.185]

Der Positionssensor stellt in dieser Arbeit das Längenmesslineal von Heidenhain in der Fügeachse dar. Dieses ist jedoch über EnDat2.2 mit EtherCAT am SPS-Rechner angeschlossen und dient als Positionswert in der Datengenerierung. Zudem gibt eine EnDat2.2 Datenschnittstelle kein zweiphasiges Rechteck-Encodersignal aus. Um dieses Problem zu umgehen gibt es eine Lösung der Firma Beckhoff. Mit der EL2521-0024 Encodersignal Simulationsklemme lässt sich aus den Signalen der Längenmessung, welche auf dem SPS-Rechner verfügbar sind, das benötigte Encodersignal erzeugen. Wie in Abbildung 2.2 ersichtlich, wird dieses simulierte Encodersignal über ein Sensorkabel an der Encoder Eingangskarte DSQC 2000 angeschlossen, welche über Profinet mit der Robotersteuerung verbunden ist. Somit kann Sensor Synchronisation wie vorgesehen verwendet werden. Allerdings bringt diese Behelfslösung weitere Fehlerquellen und vor allem Verzögerungen in den Regelkreis von Sensor Synchronisation. Bei näherer Betrachtung von Anhang A.3 und Anhang A.4 ergeben sich Zykluszeiten im Bereich bis  $20 \mu\text{s}$  für Messlineal und Encoder Simulationsklemme. Mit

dem Wissen, dass sich bei modernen SPS-Rechnern und EtherCAT Verbindungen Zykluszeiten unter 5 ms realisieren lassen, kann insgesamt eine Zykluszeit von 5 ms für die zusätzliche Verzögerung angenommen werden.

Wie in Unterabschnitt 2.2.3 beschrieben, steigen Kräfte und Momente am Endeffektor des Roboters durch Kontakt mit steifen Bauteilen der Arbeitsumgebung bei einer Lageregelung stark an. Um generell die Verwendbarkeit des Pakets zu beurteilen, müssen die Genauigkeit bzw. die Grenzen des gesamten Regelkreises betrachtet werden. Die Lageregelung mithilfe von Sensor Synchronization muss so exakt sein, dass die Kräfte, welche durch Positionsdifferenzen zwischen Fügeachse und Roboter entstehen, gering genug sind, um die in Unterabschnitt 2.1.1 aufgeführten Grenzen einzuhalten. Folgende Grenzen und Genauigkeiten sind in [1] angegeben: Die untere Grenze der Geschwindigkeit, welcher der Roboter mit Sensor Synchronization folgen kann, beträgt je Encodertyp zwischen 4 und 8 mm/s, die Obere je nach Robotertyp etwa 1000 mm/s. Die Genauigkeit der Nachführung wird mit  $\pm 50$  ms angegeben. Durch Hinzuzählen der Verzögerung der Behelfslösung von 5 ms ergibt sich eine Toleranz von  $+ 55$  ms und  $- 45$  ms. Eine Beispielrechnung soll die Angaben der Genauigkeit verdeutlichen. Die zulässige Geschwindigkeit des externen Geräts, an welchem der Encoder angeschlossen ist, darf folgende Werte nicht übersteigen, damit der Roboter innerhalb der Toleranzen bleibt.

Bei der Berechnung wird als Toleranz die in Anhang A.1 aufgeführte Pfadwiederholbarkeit des Roboters von 0,13 mm angenommen. Mit dieser Toleranz ist kein wesentlicher Anstieg der Kräfte und Momente am Endeffektor zu erwarten. Für die Berechnung wird die größere Toleranzseite mit 55 ms verwendet.

$$v = \frac{s}{t} = \frac{0,13 \text{ mm}}{0,055 \text{ s}} = 2,36 \text{ mm/s} \quad (2.12)$$

Gleichung 2.12 zeigt, dass die zulässige Fügegeschwindigkeit im Vergleich zu den üblichen Geschwindigkeitsprofilen der Robotik vgl. [5, S.1693] sehr gering ist und sich vor allem unter der Mindestgeschwindigkeit von 4 - 8 mm/s, welche für das Paket erforderlich ist, befindet. Für Fügeprozesse werden bei thyssenkrupp Presta meist Geschwindigkeiten im Bereich  $v = 3 - 10$  mm/s verwendet, was ebenfalls unter oder nahe der Mindestgeschwindigkeit liegt. Dazu wird eine zweite Beispielrechnung mit einer Fügegeschwindigkeit von  $v = 10$  mm/s

angestellt und es ergibt sich:

$$s = v * t = 10 \text{ mm/s} * 0,055 \text{ s} = 0,55 \text{ mm} \quad (2.13)$$

Gleichung 2.13 zeigt, dass die Abweichung zwischen Roboter und Fügeachse bis zu 0,55 mm betragen kann. Diese Abweichung muss die Elastizität des Roboters mit angebautem Werkzeug ausgleichen können, ohne dass sich die Kräfte und Momente am Endeffektor wesentlich erhöhen.

Die Aktivierung des Pakets erfolgt durch das Ausführen von mehreren Befehlen [1]:

*ActUnit SSYNC1*: Aktiviert den Encoder als separate Einheit im Roboter unter dem Namen SSYNC1

*WaitSensor SSYNC1*: Verbindet die erstellte Sensoreinheit SSYNC1

*SyncToSensor|On*: Startet die synchronisierte Bewegung des Roboters mit dem externen Gerät

### 2.3.4 Externally Guided Motion

Externally Guided Motion (EGM) ist ein Softwarepaket, welches den Ansatz der Lageregelung verfolgt. Es ist auch der externen Regelung zuzuordnen, da je nach Einstellung des Pakets ein zusätzlicher Sensor für die Position notwendig ist. Der wesentliche Unterschied zu Sensor Synchronization besteht darin, dass der Sensor nicht direkt mithilfe einer Eingangskarte an die Robotersteuerung angeschlossen wird. Die Übertragung der Positionsdaten erfolgt bei EGM mittels des in Unterabschnitt 2.1.3 beschriebenen UDP-Protokoll. Mithilfe der UDP-Übertragung kann ein stabiler Datenaustausch mit 250 Hz erfolgen, was einer Zykluszeit von 4 ms entspricht. Ähnlich wie bei Force Control ermöglicht EGM drei verschiedene Verwendungsmöglichkeiten [2]. Bei *EGM Position Stream* sendet die Robotersteuerung über UDP laufend die Roboterpositionen an ein externes Gerät. Somit folgt der Roboter einer programmierten Bahn und das externe Gerät folgt mithilfe der Positionsdaten dem Roboter. Bei *EGM Position Guidance* bekommt der Roboter über UDP Positionsdaten eines externen Geräts übermittelt und folgt diesen. Bei *EGM Path Correction* folgt der Roboter einer programmierten Bahn und verwendet die über UDP übertragenen Positionsdaten zur Korrektur der Bahn.

*EGM Path Correction* ist für die Anwendung bei diesem Versuchsaufbau nicht geeignet, da folgende Einschränkungen gelten: Einerseits muss das externe Gerät auf dem Roboter montiert sein und andererseits können nur Positionskorrekturen in Y- und Z-Richtung und nicht in X-Richtung, was hier der Bahnrichtung (Tangente der Bahn) entspricht, durchgeführt werden. Weiters sind keine Ausrichtungskorrekturen möglich. Genau die Korrekturen in Bahnrichtung und der Ausrichtung sind jedoch wesentlich für die Untersuchung der Anwendbarkeit. *EGM Position Stream* ist grundsätzlich eine Möglichkeit, da die Positionsdaten des Roboters an ein externes Gerät gesendet werden. Das heißt für diesen Anwendungsfall, dass der Roboter die Positionsdaten an die Fügeachse senden würde und diese entsprechend die Fügegeschwindigkeit anpasst. Eine grundsätzliche Idee dieser Arbeit ist jedoch, dass sich der Roboter abhängig von der Fügeachse während des Fügeprozesses mitbewegt und nicht umgekehrt. Somit können auch die Parameter des Fügeprozesses gesammelt bei der Fügeachse eingestellt und gespeichert werden. Für einen weiteren Ausblick ist diese Option als alternative Möglichkeit jedenfalls geeignet.

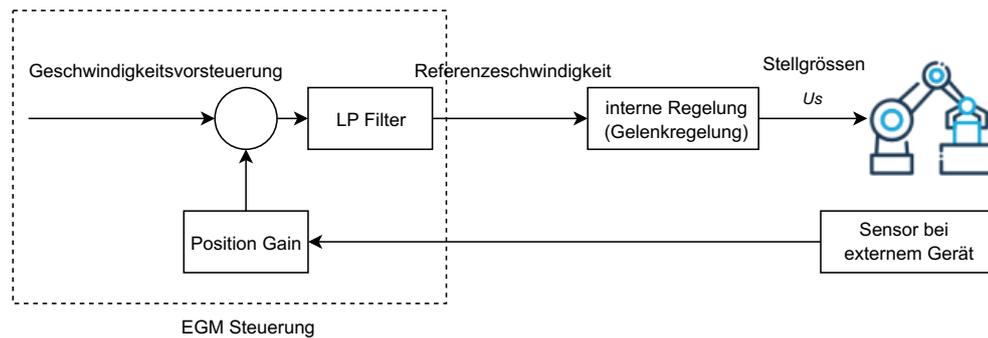


Abbildung 2.9: EGM Regelkreis

Quelle: Eigene Ausarbeitung in Anlehnung an [2]

*EGM Position Guidance* erfüllt grundsätzlich die Voraussetzungen der zugrunde liegenden Idee, indem der Roboter über UDP Positionsdaten eines externen Geräts erhält und diesen folgt. Es ist somit geeignet, jedoch ist eine Einschränkung zu beachten. Bei *EGM Position Guidance* ist es nicht möglich Linearbewegungen mit dem Roboter durchzuführen, da die Interpolationsfunktionalität nicht vorhanden ist. Wie in Abbildung 2.9 ersichtlich, werden die Positionsdaten des externen Gerätes direkt in der EGM-Steuerung verarbeitet. Dazu werden die eingehenden Sensorsignale mit einem Proportionalfaktor verstärkt und anschließend mit dem ebenfalls in Unterabschnitt 2.3.2 verwendeten Tiefpassfilter

gefiltert. Dieses Ausgangssignal der EGM-Steuerung dient direkt als Eingangssignal der Gelenkregelung. Dies bedeutet, dass die Bewegung des Roboters während des Fügevorgangs von der Startposition, der Stellung der Gelenke und den eingehenden Positionsdaten abhängt. Sollte bei einem Fügeprozess eine lineare Bewegung gefordert sein, müssen die linearen Positionsvorgaben der Fügeachse in so kurzen Abständen an den Roboter gesendet werden, dass sich dadurch die Bewegung des Roboters auch linear verhält. Sozusagen ersetzen dann die Positionsdaten der Fügeachse die Interpolationsfunktion. So würde beispielsweise bei einer UDP Zykluszeit von 4 ms und einer Fügegeschwindigkeit von 10 mm/s alle 40 µm eine neue Position gesendet. Dies kann für einen Interpolationspunkt als ausreichend genau angenommen werden.

Die Genauigkeit an sich hängt einerseits von der Zykluszeit der UDP Verbindung von mindestens 4 ms und andererseits von den Verzögerungen des externen Gerätes und der Robotersteuerung ab. Als Gesamtverzögerung ist eine Zeit von 20 ms angegeben [2, S.17]. Dies ist weniger als die Hälfte der Verzögerung von Sensor Synchronisation, wodurch auch in einer Beispielrechnung mit einer Fügegeschwindigkeit von  $v = 10 \text{ mm/s}$  sich nachstehendes Ergebnis zeigt:

$$s = 10 \text{ mm/s} * 0,02 \text{ s} = 0,2 \text{ mm} \quad (2.14)$$

Wie Gleichung 2.14 zeigt, beträgt die Abweichung des Roboters zur Fügeachse bei niedrigen Fügegeschwindigkeiten maximal 70 µm mehr als die Pfadwiederholbarkeit des Roboters. Es ist zu erwarten, dass diese Abweichung die Elastizität des Roboters mit angebautem Werkzeug ausgleichen kann, ohne dass sich die Kräfte und Momente am Endeffektor wesentlich erhöhen.

Die Aktivierung des Pakets erfolgt durch das Ausführen von nachstehenden Befehlen [2]:

*EGMGetId egmID1*: Lösen einer EGM ID egmID1 für das externe Gerät  
*EGMSetupUC ROB\_1, egmID1, "default",EGMsensor:\pose*: Baut eine UDP Verbindung egmID1 mit default Konfiguration und EGMsensor als Datenquelle mit kartesischem Koordinatensystem für Roboter 1 auf  
*EGMActPose egmID1\Tool:= tTestTool,...*: Aktiviert einen EGM Prozess mit egmID1 mit dem Werkzeug tTestTool; dabei müssen weitere Koordinatensysteme und Toleranzen, zur Festlegung ab wann eine Position als erreicht gilt, als Parameter angegeben werden.

*EGMRunPose egmID1, Mode,...*: Führt die vom externen Gerät *egmID1* geführte Bewegung durch. *Mode* gibt dabei die Art des Stops der Bewegung an. Weiters können noch Bewegungsrichtungen, Rampenzeiten und Offsets angegeben werden.

## 2.4 Fügeoperationen zur Beurteilung der Prozessqualität

Folgendes Kapitel beschreibt die verschiedenen Fügeversuche, um das Verhalten des Roboters in den unterschiedlichen Unterstützungsverfahren zu beurteilen. Die dazu notwendigen Füge- oder Greiferwerkzeuge sind einerseits die bereits erwähnte Stahlplatte in der Form eines Tischtennisschlägers (Werkzeugname: *tTennisRacket*), andererseits ein pneumatisches Greiferwerkzeug (Werkzeugname: *tCover*) für den Transport und das Fügen des Getriebegehäuses, wie in Unterabschnitt 2.4.4 beschrieben. Zur korrekten Anwendung des Werkzeugs auf dem Roboter und für die Kompensation der Gravitationskräfte im Kraftsensor sind entsprechende Werkzeugdaten notwendig. Tabelle 2.2 beschreibt die Daten der beiden verwendeten Werkzeuge. Die Daten der Schwerpunkte und des Gewichts wurden dabei durch Ausführen der Routine *FC Calib* ermittelt. TCP steht dabei für Tool Center Point. Die Positionsangaben sind auf das grundlegende Werkzeugkoordinatensystem am Flansch des Roboters bezogen.

Parameter	Einheit	tTennisRacket	tCover
Masse	kg	2.84	5,94
TCP X	mm	0	-14,27
TCP Y	mm	0	24,73
TCP Z	mm	171,48	234,84
Schwerpunkt X	mm	-0,53	14,66
Schwerpunkt Y	mm	-0.18	-36,37
Schwerpunkt Z	mm	57.22	95,71

Tabelle 2.2: Roboter Werkzeugdaten  
Quelle: Ausarbeitung des Verfassers

Weiters werden zwei Koordinatensysteme im Bereich der Fügeachse verwendet. Das in Abbildung 2.10 dargestellte Koordinatensystem ist dabei das Werkob-

jektkoordinatensystem des Roboters in der Fügeachse. In diesem werden die Positionen des Roboters definiert und dienen zur Anwendung der Unterstützungspakete. Weiters sind die Kräfte und Momente in der Auswertung nach diesem Koordinatensystem ausgerichtet. Für die Definition der Fügepositionen wird ein weiteres Koordinatensystem verwendet, welches in Z nach oben, bis zum Kontaktpunkt von Roboterwerkzeug und Aufnahme verschoben ist.

### 2.4.1 Einfache Versuche

Um zu überprüfen, ob die Messdaten der beiden Kistler Kraftsensoren zueinander in ausreichendem Maße abgestimmt sind, wird wie in Abschnitt 1.3 beschrieben, ein Teil der MSA, die Linearitätsprüfung (Verfahren 4) durchgeführt. Die Auswertungsart, die hinter der Linearitätsprüfung steckt, ist die Varianzanalyse ANOVA. Diese Analyse testet, ob sich Mittelwerte mehrerer unabhängiger Stichproben voneinander unterscheiden. Zur Durchführung der ersten Linearitätsprüfung fährt die Fügeachse solange aus, bis das Fügewerkzeug mit der Aufnahme in Kontakt tritt. Abbildung 2.10 stellt die Konfiguration dieser Linearitätsprüfung dar. Beide Kistlersensoren werden so mit derselben Belastung beaufschlagt, sofern der Roboter keine zusätzliche Kraft generiert, und es wird in fünf Stufen eine immer höhere Kraft aufgebracht. Dabei liegt die Kraftspanne im Bereich der zu erwartenden Fügekräfte. Für die Versuchsdurchführung kommen die Kraftstufen 500, 1250, 2000, 3000 und 5000 N zur Anwendung. Das Programm der Fügeachse setzt sich wie folgt zusammen. Die Vorpositionierung erfolgt mit einer Geschwindigkeit von 80 mm/s und einer Fügegeschwindigkeit von 3 mm/s. Diese Fügegeschwindigkeit wird solange gehalten, bis die Fügeachse einen Kraftwert von 500N erreicht hat. Nach einer Haltezeit von 0,5s erfolgt der Wechsel auf die nächste Kraftstufe. Analog werden die weiteren Kraftstufen angewendet.

Währenddessen sind die Messsysteme aktiv (siehe auch Abbildung 2.11) und die SPS speichert die gemessenen Werte in die in Unterabschnitt 2.1.3 angegebene CSV-Datei. Die Messwerte werden dabei mit einer Zykluszeit von 4 ms durch die SPS aufgezeichnet. Bei der nachfolgenden Verarbeitung wird ein Mittelwert aus 20 Werten pro Kraftstufe und Messdurchgang gebildet. Bei thyssenkrupp Presta gilt die Vorgabe, dass 12 Messdurchgänge für die Linearitätsprüfung durchzuführen sind [13, S.9-11]. Die Berechnung der Linearität selbst erfolgt mithilfe der Software Q-DAS Solara. Die Mittelwerte sind über die Eingabe-

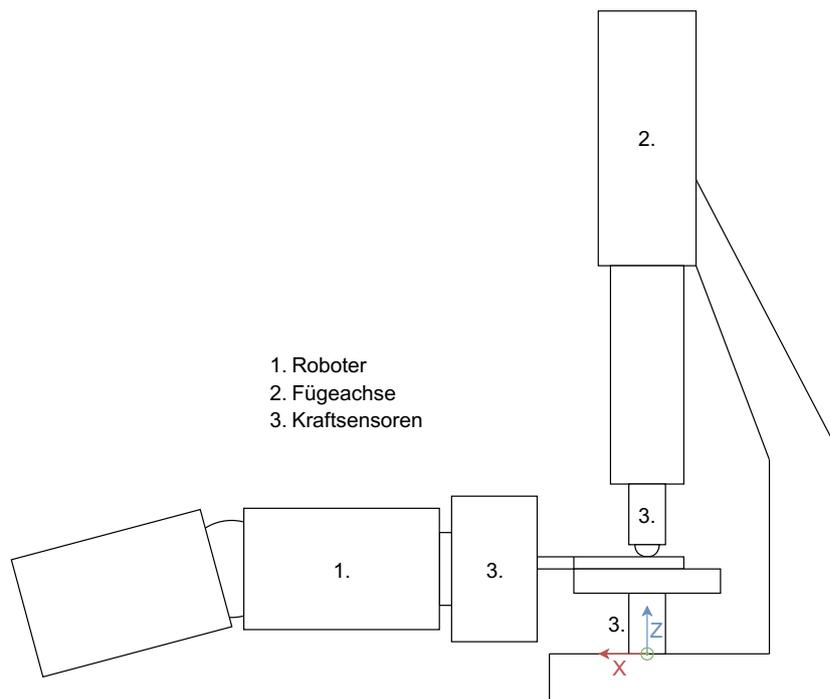


Abbildung 2.10: Konfiguration Linearitätsprüfung Kistler Kraftsensoren  
Quelle: Eigene Ausarbeitung

maske einzutragen. Weiters sind die Daten zu den Messmitteln und ein Merkmal mit zugehöriger Toleranz anzugeben. Hier wird als Merkmal eine Fügekraft von 5000 N in Bezug auf nachfolgende Fügeversuche (siehe Unterabschnitt 2.4.4) definiert. Als Toleranz dient die in Unterabschnitt 2.1.1 angegebene Genauigkeit von 2 % des Kistler Kraftsensors, was eine Toleranz von  $\pm 50$  N bedeutet, da der Messbereich auf 5000 N eingestellt ist. Nach Eingabe aller Daten kann die Linearitätsprüfung durchgeführt werden. Zur komfortableren Bedienung während den Versuchen ist dieser Ablauf in einer Schleife mit variabler Anzahl an Durchgängen eingebettet. Die Schleife stellt dabei die Schrittkette in der SPS dar. Weiters ist die übergeordnete Steuerung des Roboters in dieser Schrittkette integriert. Die Ansteuerung des Roboters erfolgt durch gesammelte Schritte, sogenannte Steps und kann mittels einer Schrittnummer identifiziert werden. Ein Step stellt dabei alle Bewegungen des Roboters dar, die während des aktuellen Status der Schrittkette auf der SPS durchgeführt werden können, ohne dass Kollisionen entstehen vgl. [10, S.17-18], [18]. Beispielsweise kann ein Roboter ein neues Bauteil abholen, während ein Modul eine Grundstellungsfahrt durchführt, oder ein Roboter bekommt die Freigabe sich in ein Modul zu bewegen,

indem er die entsprechende Schrittnummer von der SPS erhält. Die Anwendung der Steps für die Steuerung des Roboters ist in Abbildung 2.11 ersichtlich.

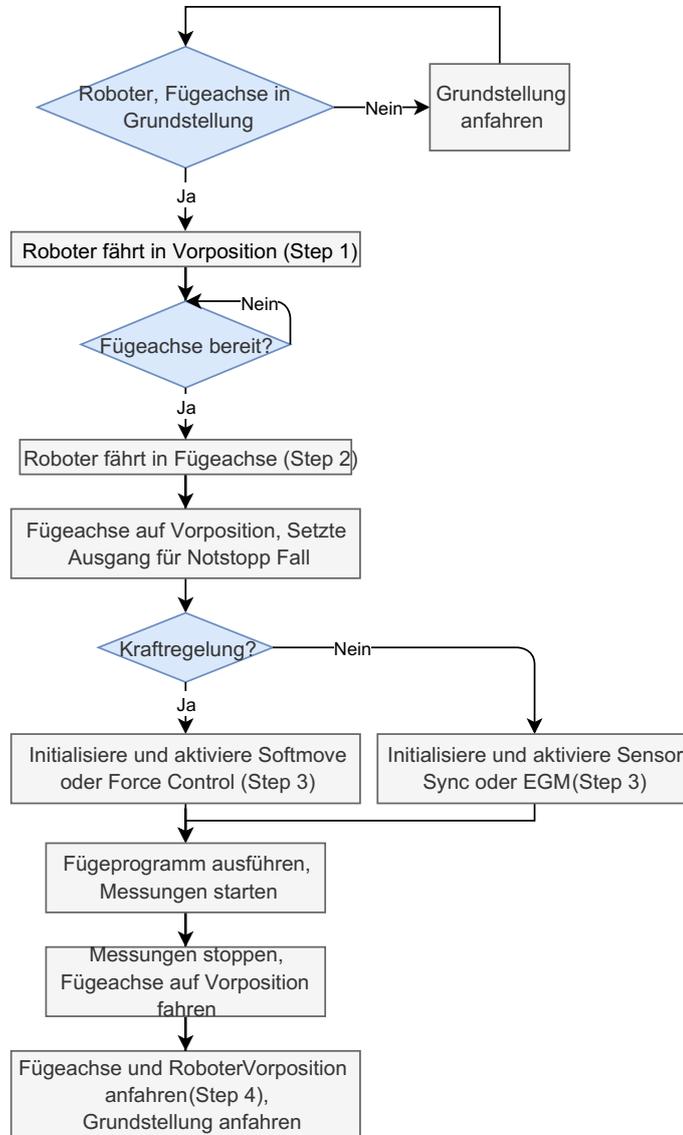


Abbildung 2.11: Grundlegender Ablauf Fügeprozess  
Quelle: Eigene Ausarbeitung

Um die grundsätzliche Funktion der Nachführung in Force Control zu prüfen, wird der Roboter ohne Belastung in der Luft durch die Achse bewegt (ähnlich wie in Abbildung 1.2 dargestellt). Dieser Versuch gibt Auskunft über die auftretenden Kräfte am Roboter Endeffektor ohne die zusätzlichen Einflüsse ei-

nes realen Fügeprozesses. In Abbildung 2.11 ist der Ablauf dieses Fügeversuchs als Flowchart dargestellt. Dabei sind auch die entsprechenden Schrittnummern angegeben, die von der SPS an den Roboter gesendet werden. Die Kräfte des Roboter-Kraftsensors und des oberen Kistler-Kraftsensors auf der Fügeachse sind gleichzeitig auch für die Linearitätsprüfung der beiden Komponenten geeignet. Dabei ist zu beachten, dass bei dieser Linearitätsprüfung die Umrechnung des Sensor Koordinatensystems in das kartesische Koordinatensystem durch die Robotersteuerung mit bewertet wird. Die Berechnung findet dabei laufend in einem separaten Roboter Programm statt. Die Kräfte- und Momente, aber auch die Position und die Orientierung des Roboters werden dabei ausgelesen und in vorzeichenlose Ganzzahlen umgewandelt. Diese Ganzzahlen, übermittelt über Profinet, interpretiert die SPS wieder als Gleitkommazahlen. Die Gleitkommazahlen in der SPS stellen die Messwerte der Positionen 5 - 16 der Tabelle 2.1 dar.

Aufgrund dessen, dass der Kistler-Kraftsensor nur einen Freiheitsgrad messen kann, die Z-Kraft, kann für die Linearitätsprüfung ebenfalls nur die Z-Komponente des Roboter-Kraftsensors herangezogen werden. Die Kraft, die notwendig ist, um den Roboter durch die Fügeachse zu bewegen, ist, wie in Unterabschnitt 2.3.2 beschrieben, abhängig von der Geschwindigkeit der Fügeachse und von den Dämpfungsparametern in Force Control. Für die Durchführung der zweiten Linearitätsprüfung kommt eine Geschwindigkeit von  $v = 5 \text{ mm/s}$  zum Einsatz. Um die unterschiedlichen Kraftstufen zu generieren, werden die Dämpfungsparameter 100, 200, 300, 700, 1400 % der Systemeinstellungen von  $1400 \text{ N s/m}$ , in Z-Richtung, verwendet. Die Dämpfungsparameter, multipliziert mit der Geschwindigkeit der Fügeachse, ergeben berechnete Kraftstufen von 7, 14, 21, 49 und 98 N. Die Dämpfungsparameter der restlichen fünf Freiheitsgrade werden auf 1000 % gestellt. Für die Bewertung der Linearität wird ein Nennmaß von 100 N verwendet. Dies hat jedoch keinen direkten Einfluss auf das Ergebnis. Wesentlich dafür ist die Toleranz [13, S.9-11]. Bei dieser Linearitätsprüfung ist das Ziel, die Toleranz so anzupassen, dass ein noch fähiges Messmittel vorliegt, um die Größenordnung der erreichbaren Toleranz ermitteln zu können. Die benötigten Mittelwerte werden ähnlich wie bei der ersten Linearitätsprüfung erzeugt, jedoch mit dem Unterschied, dass pro Kraftstufe jeweils 12 Durchgänge notwendig sind. Ergebnisse dieser Verifikation sind in Unterabschnitt 3.1.2 aufgeführt. Weiters gilt es zu erwähnen, dass die Kraft- und Momentsignale des Roboter-Kraftsensors nicht nur während der Lineari-

tätsprüfung oder aktiviertem Force Control in Verwendung sind, sondern auch als Messwerte für die Auswertung bei der Anwendung der weiteren Unterstützungsverfahren.

### 2.4.2 Federpaket

Um die Anforderung an die Nachführung des Roboters zu erhöhen, wird eine Federpaket Baugruppe eingesetzt. Wie Abbildung 2.12 zeigt, besteht diese aus zwei zylindrischen Stahlscheiben, welche mit 4 gehärteten Zylinderstiften zueinander geführt sind. Jeweils 45° verdreht dazu befinden sich 4 Druckfedern, die eine benötigte Fügekraft simulieren. Zum Erzeugen einer Vorspannung von ca. 500 N und damit das Federpaket eine in sich geschlossene Baugruppe darstellt, ist eine Schraube zentral im Federpaket verbaut. Das Federpaket stellt einen Federweg von 20 mm und einen berechneten Fügekraftverlauf von ca. 500 N in vorgespannter und bis ca. 1400 N in komprimierter Lage zur Verfügung.

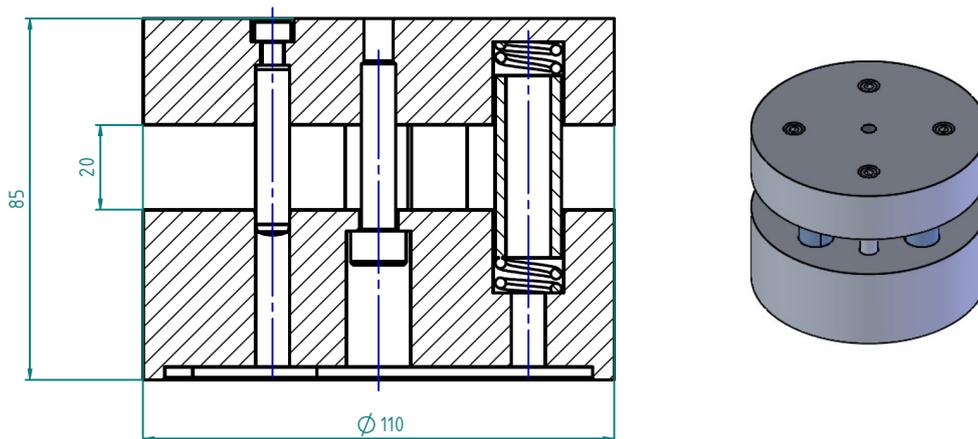


Abbildung 2.12: Baugruppe Federpaket  
Quelle: Eigene Ausarbeitung

Als neue Einflüsse beim Fügeprozess können dabei die Belastung durch die Federkraft während des Fügevorganges, die Reibung der Führungsstifte und die innere Reibung der Federn (diese ist bei zylindrischen Druckfedern allerdings vernachlässigbar [21, S.302]) identifiziert werden. Für die Durchführung des Fügeversuches ist kein eigenes Fügewerkzeug in der Fügeachse notwendig, es kommt eine Distanzhülse als Schutz des Kistler Kraftsensors zur Anwendung.

Die Verwendung des Federpaketes zur Simulation von Fügekräften verursacht allerdings ein Gefahrenpotential für den Roboter. Dies muss bei der Programmierung der Grundstellungsfahrt speziell beachtet werden. Tritt nämlich während des Fügeprozesses, wenn der Roboter durch die Fügeachse auf das Federpaket gedrückt wird, ein Not-Stopp Ereignis ein, ist der Roboter eingespannt. Würde nun erst die Fügeachse und anschließend der Roboter eine herkömmliche Grundstellungsfahrt durchführen, würde der Roboter während der Grundstellungsfahrt der Fügeachse die Kraft des Federpakets aufnehmen müssen. Da diese je nach Komprimierung des Federpakets zwischen 500 und 1400 N beträgt, würde die zulässige Kraft  $F_h$  aus Gleichung 2.1 überschritten werden. Um dies zu vermeiden ist, abhängig vom verwendetem Roboterverfahren, die Kraft- oder Lageregelung bei der Grundstellungsfahrt zu aktivieren, bevor die Fügeachse eine Grundstellungsfahrt durchführt. Somit führen Roboter und Fügeachse mindestens solange, bis das Federpaket auf Ausgangsposition ausgefedert ist, synchron eine Grundstellungsfahrt durch. Danach kann die Fügeachse die Grundstellungsfahrt abschließen und nachfolgend der Roboter. Dies ist nicht nur bei der Verwendung des Federpakets zu beachten, sondern auch allgemein bei Fügeprozessen mit Bauteilen, die ein federndes Verhalten während des Fügens verursachen.

### 2.4.3 Lager fügen

Um das Verhalten der Nachführung des Roboters an realen Bedingungen zu untersuchen, werden ein Lager und ein Getriebedeckel verwendet. Abbildung 2.13 stellt den Fügeablauf dar. Um das Lager aufnehmen zu können, wird in das Tischtennisschlägerwerkzeug des Roboters eine kreisförmige Vertiefung gefräst. Zwei weitere, kreisförmige Taschen kommen ebenfalls hinzu, um Magnete zu platzieren. Diese sollen das Lager während der Bewegungen des Roboters in Position halten. Das Lager wird von Hand eingelegt und von den Magneten in Position gehalten. Ebenso ist der Getriebedeckel auf der Bauteilaufnahme in der Fügeachse zu platzieren. Der Roboter bewegt das Lager in die Vorposition vor der Fügeachse. Der weitere Fügeprozess findet analog zu Abbildung 2.11 statt, jedoch mit dem Unterschied, dass abweichende Fügegeschwindigkeiten und Positionen verwendet werden. Als Fügestrategie kommt das Fügen auf Signal zum Einsatz. Das Signal stellt dabei die interne Kraftmessung der Fügeachse dar. Dies bedeutet, dass die Fügeachse den Fügevorgang mit dem Erreichen einer

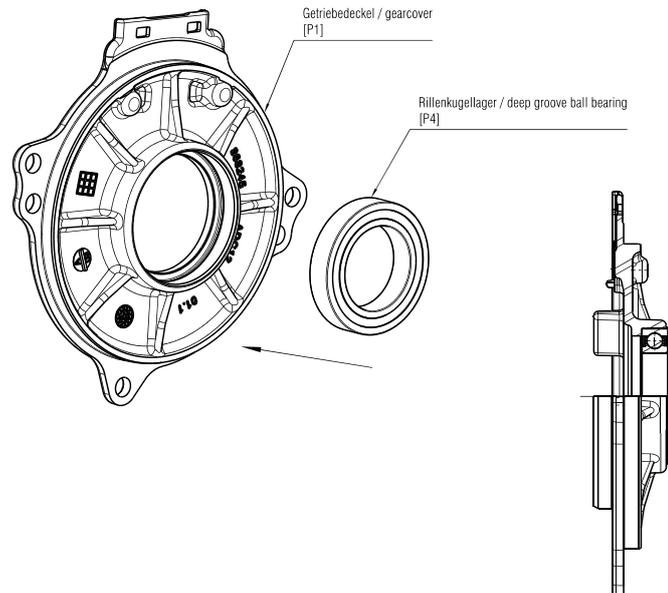


Abbildung 2.13: Fügeablauf Lager einpressen  
Quelle: thyssenkrupp Presta

Kraftschwelle beendet. Aus den Montagedokumenten geht hervor, dass eine Fügekraft im Bereich von mindestens 100 N bis maximal 5000 N zu erwarten ist.

### 2.4.4 Getriebegehäuse mit Welle auf Deckel fügen

Mit diesem Fügeversuch wird nicht nur das Fügen des Getriebegehäuses auf den Getriebedeckel durchgeführt, sondern auch das Durchschieben der Getriebewelle durch das Lager, welches im Getriebedeckel verpresst ist. In Abbildung 2.14 ist die Anordnung der Komponenten während des Fügeablaufes ersichtlich. Die Toleranzpaarung zwischen Lager und Getriebewelle ist eine Übergangspassung, wodurch das Lager von Hand mit geringer Kraft aufschiebbar sein soll. Dabei ist die Positionierung und Orientierung des Lagers zur Getriebewelle am Beginn des Aufschiebens entscheidend, da die Kräfte bei schrägem Ansetzen stark steigen können. Somit besteht die Anforderung an die Nachführung des Roboters nicht nur im Halten der Bauteile während dem Fügen, sondern auch im korrekten Ausrichten der Bauteile am Beginn des Aufschiebens. Den Montagedokumenten ist zu entnehmen, dass während dem Einpressen des Getriebede-

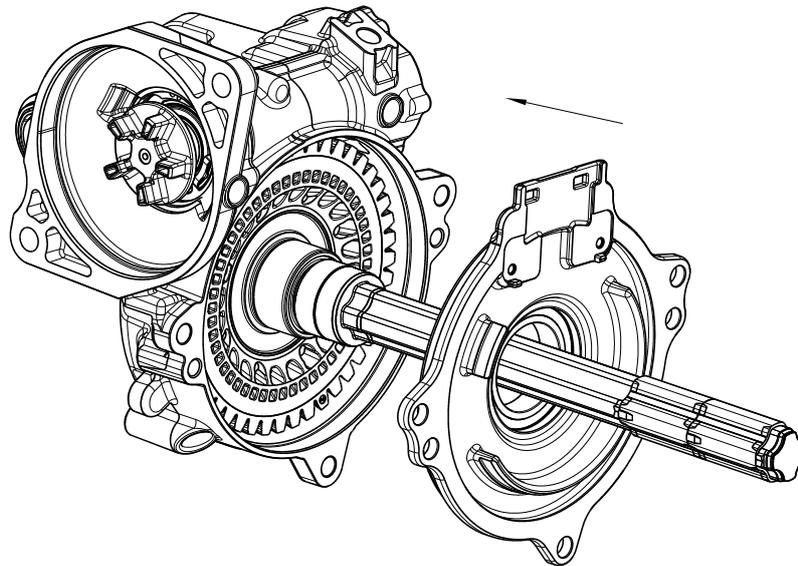


Abbildung 2.14: Fügeablauf Getriebegehäuse auf Deckel fügen mit Welle auf Lager schieben  
Quelle: thyssenkrupp Presta

ckels eine Fügekraft im Bereich von mindestens 1000 N bis maximal 4500 N zu erwarten ist.

Für diesen Versuchsablauf ist ein passendes Werkzeug für den Roboter notwendig, damit der Roboter das Getriebegehäuse und die Getriebewelle halten und bewegen kann. Ebenso bedarf es einer definierten Aufnahmeposition, an der die beiden zu fügenden Bauteile vom Roboter abgeholt werden können. Abbildung 2.15 zeigt den Robotergreifer in der Vorposition zum Abholen der, auf der Aufnahmeposition bereitgestellten Komponenten.

In diesem Werkzeug sind zwei pneumatische Greifer verbaut, die einerseits das Getriebegehäuse und andererseits die Getriebewelle fixieren. Die Greifer werden über die beiden, auf dem Roboter befestigten Pneumatikventile, angesteuert. Somit erweitert sich der Roboter Step 1 (siehe Abbildung 2.11) um eine Vorposition zum Abholen der Komponenten, eine Abholposition in der die Komponenten gegriffen werden und um eine Zwischenposition für den Transport. In Abbildung 2.15 lässt sich auch erkennen, dass die Getriebewelle gekürzt ist. Dies ist dabei dem begrenzten Fügweg der Promess Achse und der Tatsache geschuldet, dass der untere Kistler Kraftsensor sowie die Getriebewelle in kon-

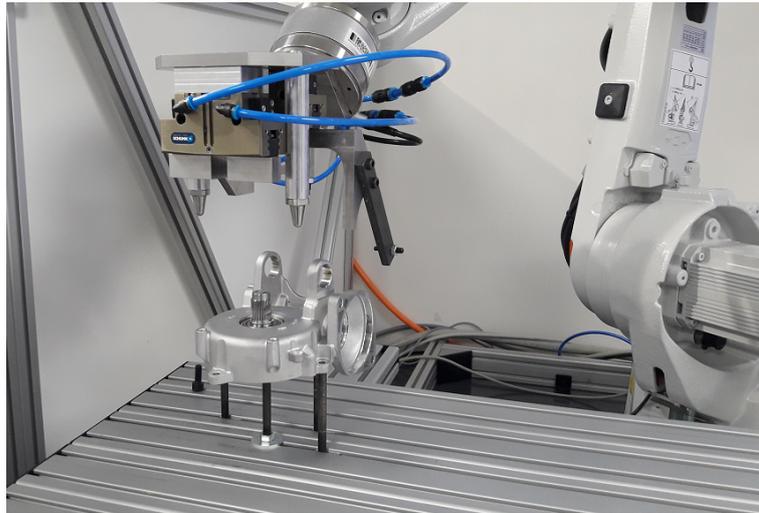


Abbildung 2.15: Robotergreifer mit Aufnahme für Getriebegehäuse und Welle  
Quelle: eigene Ausarbeitung

zentrischer Lage zueinander gefügt werden. Der Lagersitz ist jedoch vollständig erhalten, was somit keine negativen Einflüsse auf den Fügeversuch erwarten lässt.

# 3 Durchführung der Versuche und Ergebnisse

Im folgenden Kapitel werden die Ergebnisse der zuvor erläuterten Fügeoperationen dargestellt. Dabei wird zuerst darauf eingegangen, ob die Messsysteme die notwendigen Kriterien erfüllen, damit diese auch verwendet werden können. Anschließend werden in einer Übersicht die weiteren durchgeführten Versuche und die dazugehörigen Auswertungen dargestellt.

## 3.1 Verifizierung der Messwerte

### 3.1.1 Linearität Kistler-Kraftsensoren in der Fügeachse

Durch zwölfmaliges Ausführen der in Unterabschnitt 2.4.1 beschriebenen Schleife, werden die Messwerte für die Linearitätsprüfung erstellt. Die Berechnung der Mittelwerte erfolgt in einem Matlab Script, das automatisiert die CSV Dateien einliest, daraus Matrizen mit den Messwerten generiert und dann die Mittelwerte der einzelnen Kraftstufen bildet. Die Matrizen haben ein Format, das es ermöglicht, die Werte direkt in die Software Q-DAS Solara zu übernehmen. Das Matlab Script ist unter Anhang B.1 einsehbar.

In Abbildung 3.1 sind jeweils die 12 Mittelwerte der fünf Kraftstufen zu erkennen. Die Auswertung, welche in Anhang C.1 aufgeführt ist, ergibt eine systematische Messabweichung, den Bias von 2,15%. Die aktuellen Vorgaben der thyssenkrupp Presta von 2018 für die Linearitätsprüfung definieren für ein fähiges Messmittel einen Bias von  $\leq 5\%$  und für ein bedingt fähiges Messmittel  $\leq 10\%$  [13, S.9-11]. Somit ist die Linearitätsprüfung erfolgreich und die Kistler Kraftsensoren können für die Versuchsdurchführung verwendet werden.

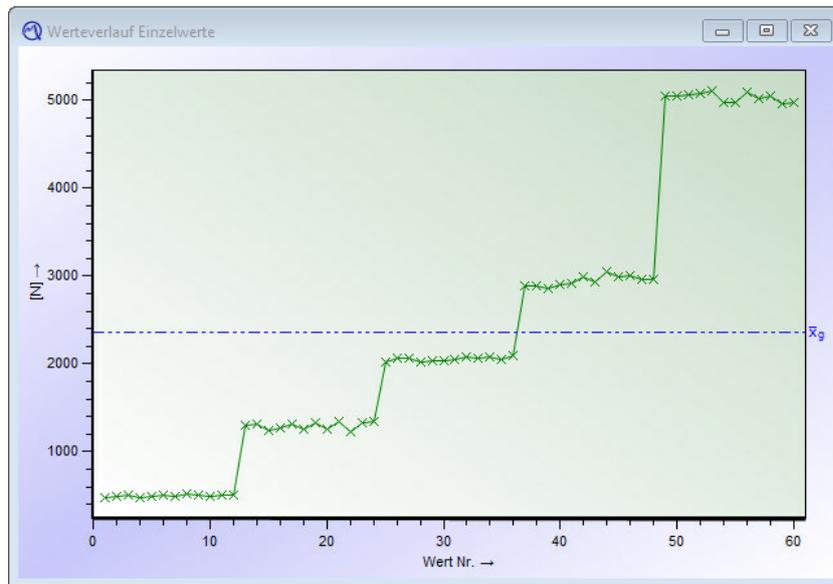


Abbildung 3.1: Mittelwerte der Kraftstufen für die Linearitätsprüfung des oberen Kistler-Kraftsensors in Solara  
Quelle: eigene Ausarbeitung

### 3.1.2 Linearität des oberen Kistler-Kraftsensors in der Fügeachse und des Roboter-Kraftsensors

Um die Schrittkette möglichst einfach zu halten, sind bei dieser Linearitätsprüfung je Kraftstufe 12 Durchgänge auszuführen, da ein Umschalten der Dämpfungsparameter des Roboters während der Bewegung der Fügeachse ohne weitere Programmierung nicht möglich ist. Um die benötigten Mittelwerte zu erhalten, muss dadurch das Matlab Script angepasst werden. Anhang B.2 zeigt, dass jeweils die CSV Dateien der aktuellen Kraftstufe eingelesen werden und ein Mittelwert gebildet wird. Dies erfolgt analog für jede Kraftstufe durch Wechseln des aktuellen Ordners.

In Abbildung 3.2 sind die Messwerte der fünf Kraftstufen des Roboter-Kraftsensors ersichtlich. Bei genauerer Betrachtung fällt auf, dass die tatsächlichen Kraftstufen, speziell die höchste Kraftstufe, mit einem gemessenen und gemittelten Wert von 70 N von den berechneten abweichen. Da jedoch bei der Linearitätsprüfung die Abweichung zwischen den zwei Messmitteln betrachtet wird, hat dies keinen Einfluss auf die Berechnung. Die in Anhang C.2 aufgeführte Auswertung zeigt, dass der Roboter Kraftsensor bei einer Toleranz von  $\pm 8$  N einen Bias von 4,84 % aufweist und somit als fähig einzustufen ist.

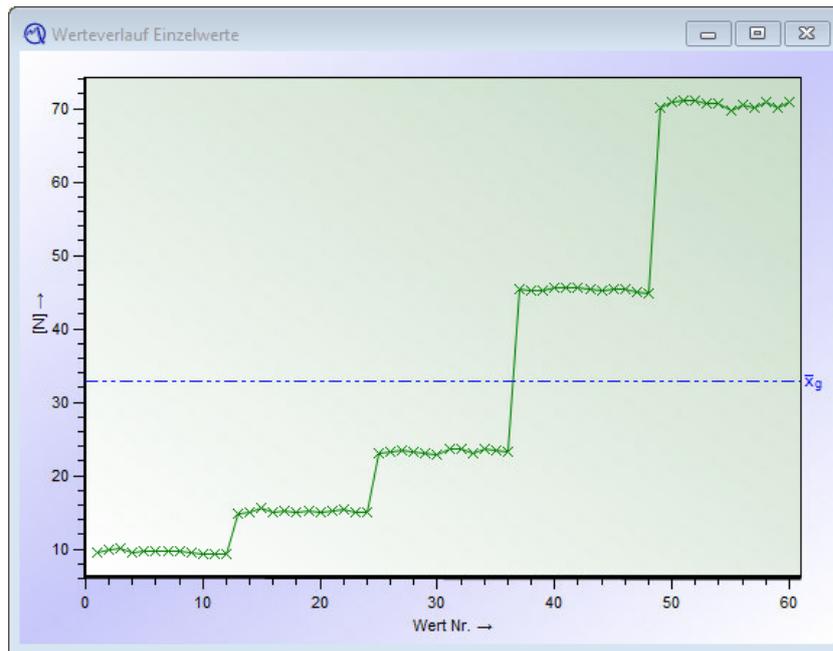


Abbildung 3.2: Mittelwerte der Kraftstufen für die Linearitätsprüfung des Roboter-Kraftsensors in Solara

Quelle: eigene Ausarbeitung

### 3.1.3 Beurteilung der Messwerte

Die SPS schreibt die Messwerte, die einerseits von den Längen- und Kraftmessgeräten über EtherCAT und andererseits vom Roboter über Profinet übertragen werden, in eine CSV-Datei. Bei Betrachtung eines Ausschnitts einer solchen CSV-Datei in Abbildung 3.3 ist auffallend, dass bei den Werten ab Spalte 5 mehrfach hintereinander gleiche Werte vorhanden sind. Die Werte ab Spalte 5 sind die über Profinet übertragenen Messwerte des Roboters wie in Tabelle 2.1 beschrieben. Dies deutet darauf hin, dass das Auslesen, das Umrechnen auf Ganzzahlen und das Senden über Profinet in der Robotersteuerung mehr als 4 ms in Anspruch nimmt. Dies führt weiters zu einer Verschiebung der Messwerte. So misst der obere Kistler Kraftsensor in Spalte 3 eine Kraft von ca. 3 N in Z-Richtung, während der Roboter über Profinet einen Wert von ca.  $-0,7$  N sendet. Erst im vierten Zyklus kommen plausible Werte über Profinet bei der SPS an.

Da sich nach Recherche keine Möglichkeit zur Beschleunigung bzw. Ermittlung der Zykluszeiten der ABB Steuerung ergeben hat, wird die Zykluszeit, mit

	1	2	3	4	5	6	7	8
1	1	131.0814	3.1817	0.0033	0.4690	0.0959	-0.7791	-0.0307
2	2	131.0417	13.1367	0.0033	0.4696	-0.1177	-0.8367	0.0209
3	3	131.0013	21.1737	0.0242	0.5284	-0.1177	-0.8367	0.0209
4	4	130.9607	19.6047	0.0285	0.5284	-0.8098	13.8710	0.2974
5	5	130.9202	18.5459	0.1482	1.0182	0.6751	20.7995	0.0983
6	6	130.8798	26.9640	0.2272	1.0182	0.6751	20.7995	0.0983
7	7	130.8400	35.5848	0.1317	2.8845	1.3629	20.7995	0.0983
8	8	130.7991	35.9658	-0.0274	2.8845	1.3629	18.2260	-0.0955
9	9	130.7582	30.3880	-0.0714	4.3716	1.3629	18.2260	-0.0955
10	10	130.7191	27.1689	-0.0549	4.3716	0.2980	36.9506	0.2324
11	11	130.6794	29.9493	0.0384	2.9204	1.3828	34.1497	0.0010

Abbildung 3.3: Ausschnitt von Messwerten in einer CSV Datei  
Quelle: eigene Ausarbeitung

der die SPS die Messwerte speichert, auf 8 ms angehoben, da mehrfach gleiche Werte keinen neuen Informationsgehalt einbringen. Die Kompensation der Verzögerung der Messwerte über Profinet erfolgt in der Auswertung mit Matlab. Dazu werden die über Profinet übertragenen, verzögerten Messwerte (Spalten 5-16), um drei Zeilen nach vorne gerückt. Somit ist die korrekte Zuordnung zu den Messwerten der restlichen, an der SPS angeschlossenen Messsysteme wieder gegeben. Durch das Vorrücken der Daten fehlen am Ende der Messdaten drei Zeilen der Roboter messwerte. Um ein Verfälschen der Daten durch das Füllen mit Nullen zu vermeiden, werden die letzten drei Zeilen gelöscht.

## 3.2 Durchführung der Fügeversuche

Im folgenden Abschnitt werden jeweils Versuchsergebnisse mit den Unterstützungsverfahren Softmove und Force Control dargestellt. Versuche mit Sensor Synchronization und Externally Guided Motion konnten einerseits aufgrund der Anfangs fehlenden Encoder Simulationsklemme EL2521-0024, andererseits aufgrund der aufwändigeren Versuche mit Force Control und Softmove nicht durchgeführt werden. Es wurde während der Projektphase entschieden, den Fokus mehr auf diese beiden Verfahren zu legen. Tabelle 3.1 soll eine Übersicht über die durchgeführten Versuche und die dazu verwendeten Parameter geben. Bei Softmove werden weitere Parameter und Systemeinstellungen auf der Grundeinstellung belassen [4]. Bei Force Control kommen nachstehende Grundeinstellungen zum Einsatz, welche dem Auslieferungszustand des Roboters

Unterstützungsverfahren				Fügeachse
<b>Force Control</b>	damp X/Y	damp Z	damp RX, RY, RZ	v
Federpaket fügen	5000 %	100 %	1000 %	10 mm/s
Lager einpressen	5000 %	100 %	1000 %	5 mm/s
Getriebegehäuse fügen	1000 %	100 %	1000 %	3 mm/s
<b>Softmove</b>	Stiffness	Stiffness to Damping ratio	Dynamic to static fric ratio	v
Federpaket fügen	0 %	0,5	0,7	10 mm/s
Lager einpressen	0 %	0,5	0,7	5 mm/s
Getriebegehäuse fügen	35 %	2	0,4	3 mm/s

Tabelle 3.1: Durchgeführte Versuche mit Parameter  
Quelle: Ausarbeitung des Verfassers

entsprechen. Es sind dies die Dämpfung bei Bewegung in kartesische Richtungen von  $D_F = 1400 \text{ N s/m}$  und bei Drehung um die kartesischen Richtungen von  $D_T = 300 \text{ N m s/rad}$ . Diese sind in der Robotersteuerung in Konfiguration/Motion/FC Kinematics einstellbar. Der *LP-filter* wird auf 3 Hz belassen. Dies gilt als Empfehlung, wenn Roboterwerkzeug und Arbeitsumgebung aus Metall und dadurch eine höhere Steifigkeit haben. Weitere Parameter und Systemeinstellung entsprechen ebenfalls den Grundeinstellungen [3].

Für die Analyse der Fügeversuche kommt ein Matlab Script zum Einsatz, welches die Daten aus den CSV-Dateien unabhängig von der Anzahl der Dateien einliest. Dabei unterscheidet das Script zwischen den Endungen 'SM' und 'FC' beim Dateinamen und liest nur die Daten des ausgewählten Unterstützungsverfahrens ein. Die Messwerte werden in Matrizen geschrieben. Danach erfolgt eine Auswertung des Fügekraftverlaufes der einzelnen Versuche, dargestellt in einem Kraft-Weg Diagramm, um vorhandene Ausreißer zu erkennen. Dargestellt werden dabei die Kraft des oberen und des unteren Kistler Kraftsensors. Weiters erfolgt eine Mittelwertbildung der Messwerte der einzelnen Versuche und eine erneute Darstellung der gemittelten Fügekraftkurve. Als nächste Auswertung erfolgt eine Differenzbildung zwischen den Kräften des oberen und des unteren Kraftsensors. Diese wird mit den kartesischen Kräften des Roboters ebenfalls als Kraft-Weg Diagramm angezeigt. Dies erfolgt in der ähnlicher Art und Wei-

se für die Drehmomente des Roboter Kraftsensors ohne die Differenzkraft. Für die weitere Verwendung und die Archivierung der Diagramme dient die Matlab-Funktion 'PrintPdf', welche die Diagramme im PDF-Format abspeichert. Die Diagramme der Fügeversuche werden, neben den einzelnen Erläuterungen in dieser Arbeit gesammelt in Anhang D angeführt.

### 3.2.1 Fügeversuche mit dem Federpaket

Beim Fügeversuch mit dem Federpaket bewegt die Fügeachse den Roboter erst in der Luft; dies findet im Bereich von 130 bis 83,5 mm statt. Bei 83,5 mm trifft das Roboterwerkzeug auf das Federpaket auf. Das Einfedern des Federpakets findet bis zur Position 72,5 mm statt.

#### Fügeversuche mit Force Control:

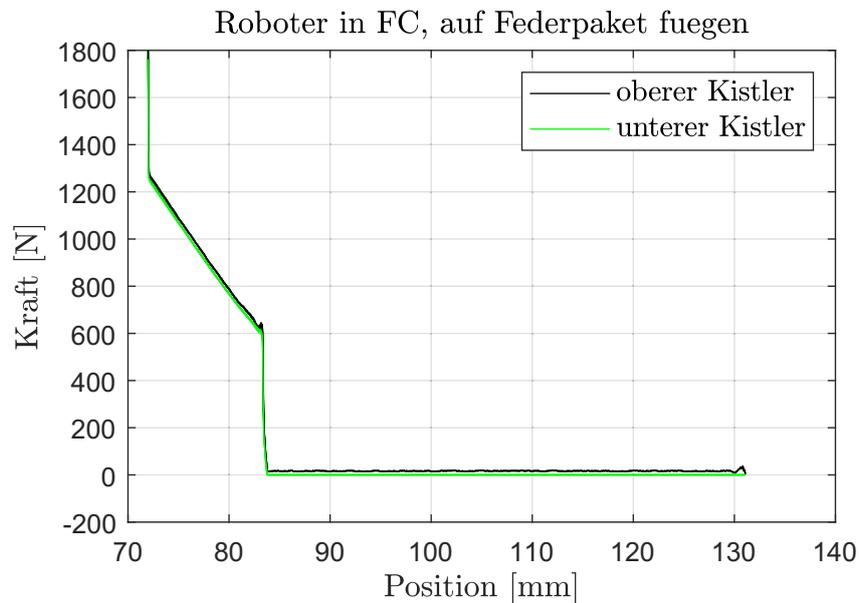


Abbildung 3.4: Fügekraftverläufe beim Federpaket mit FC

Quelle: eigene Ausarbeitung

Die Fügekraftlinien der fünf Fügeversuche in Abbildung 3.4 liegen nahe übereinander, somit sind keine Ausreißer vorhanden. Die Hookesche Gerade der Federkennlinie ist gut ersichtlich, allerdings stellt das Federpaket in der Realität einen Fügekraftverlauf von ca. 600 bis 1250 N bei einem Fügeweg von 11 mm zur Verfügung. Dies weicht von den berechneten Werten ab, ist aber für die

Durchführung der Versuche nicht weiter von Relevanz. Die gemittelte Fügekurve stellt ein ähnliches Bild dar und ist in Anhang D.1 aufgeführt.

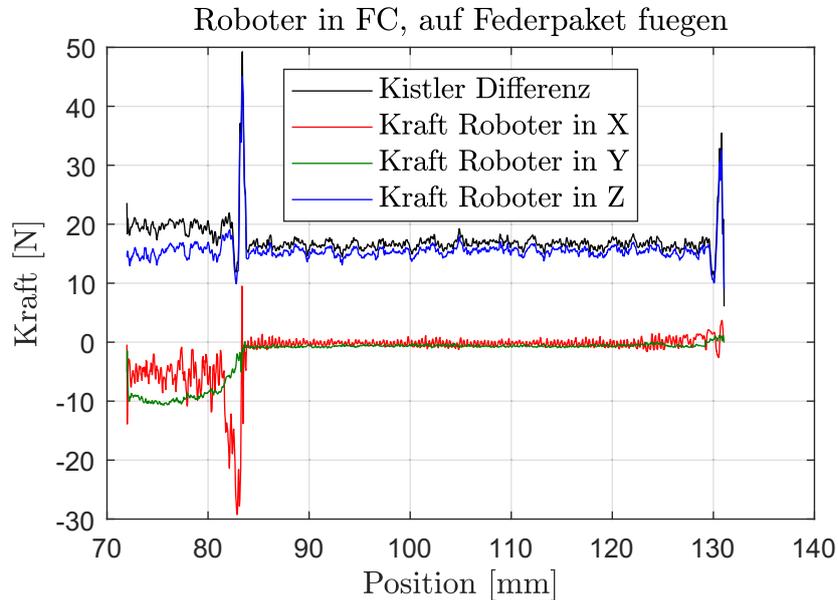


Abbildung 3.5: Abgeleitete Kräfte beim Federpaket mit FC  
Quelle: eigene Ausarbeitung

Abbildung 3.5 zeigt einerseits die Differenzkraft des oberen Kistler-Kraftsensors und des unteren Kistler-Kraftsensors in der Fügeachse sowie andererseits die Kräfte, welche vom Roboter-Kraftsensor gemessen werden. Die aufgenommene Kraft des Roboters in Z-Richtung folgt genau der Differenzkraft der beiden Kistler-Kraftsensoren, bis auf einen Offset von 1,5 N und mit Ausnahme der Kraftspitzen bei den Kontaktpunkten. Weiters bleibt die Z-Komponente des Roboters vor und nach dem Kontakt mit dem Federpaket im Bereich von 13 bis 16 N konstant. Dies deckt sich mit einer berechneten Z-Kraft des Roboter-Kraftsensors von 14 N bei der eingestellten Fügegeschwindigkeit von 10 mm/s. Beim Kontakt des Fügewerkzeugs mit dem Federpaket ist zu erkennen, dass die vom Roboter gemessenen Kräfte in X- und Y-Richtung ansteigen. Ebenfalls zu erwähnen ist die Spreizung zwischen Differenzkraft der Kistler-Kraftsensoren und gemessener Kraft in Z-Richtung beim Roboter, welche nach dem Kontakt größer wird. Bei Betrachtung der am Roboter anliegenden Drehmomente in Anhang D.1 geht hervor, dass beim Kontakt mit dem Federpaket das Drehmoment um die Y-Achse sprunghaft ansteigt.

### Fügeversuche mit Softmove:

Bei den Fügekraftlinien der fünf Fügeversuche in Anhang D.2 ist zu erkennen, dass während dem Einfedern des Federpakets eine Streuung zwischen den einzelnen Fügekraftverläufen vorhanden ist. Diese ist allerdings nicht als Ausreißer zu werten. Die gemittelte Kurve bestätigt dies, da es zu keiner wesentlichen Veränderung kommt. Auffallend ist aber die Tatsache, dass bei der Verwendung von Softmove die benötigte Fügekraft zur Erreichung des Einfedern des Federpakets wesentlich höher ist.

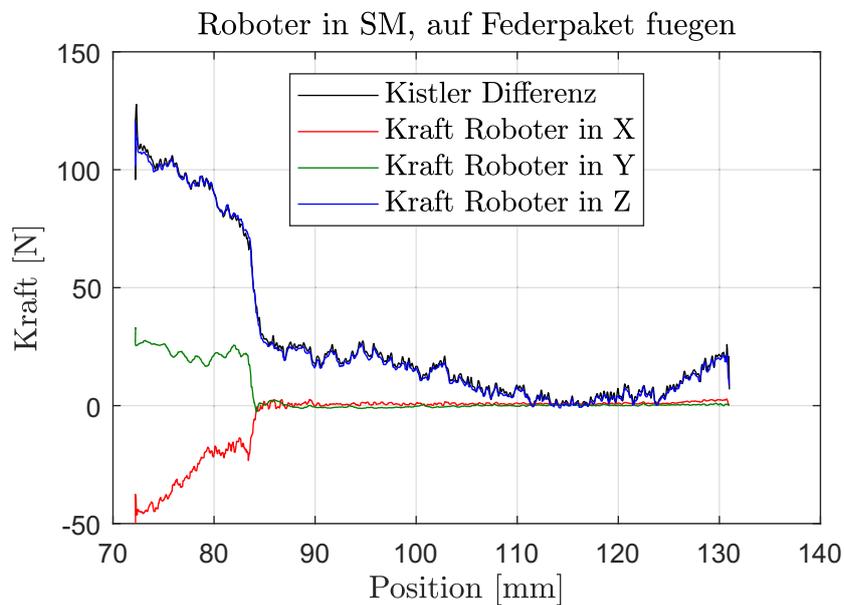


Abbildung 3.6: Abgeleitete Kräfte beim Federpaket mit SM  
Quelle: eigene Ausarbeitung

Abbildung 3.6 zeigt, dass die Differenzkraft der Kistler-Kraftsensoren und die Z-Kraft des Roboter-Kraftsensors bis auf einen Offset von 1,5 N zueinander stimmen. Jedoch schwankt die über den Roboter abgeleitete Z-Kraft im Bereich von 0 bis 21,5 N und steigt während dem Einfedern des Federpakets auf 120 N an. Die Y-Kräfte des Roboters steigen, wenn auch nicht im selben Maße, ebenfalls an, während in X-Richtung eine negative Kraft auftritt. Allerdings bewegen sich beide Kräfte während der Bewegung des Roboters in der Luft konstant in der Nähe von 0 N.

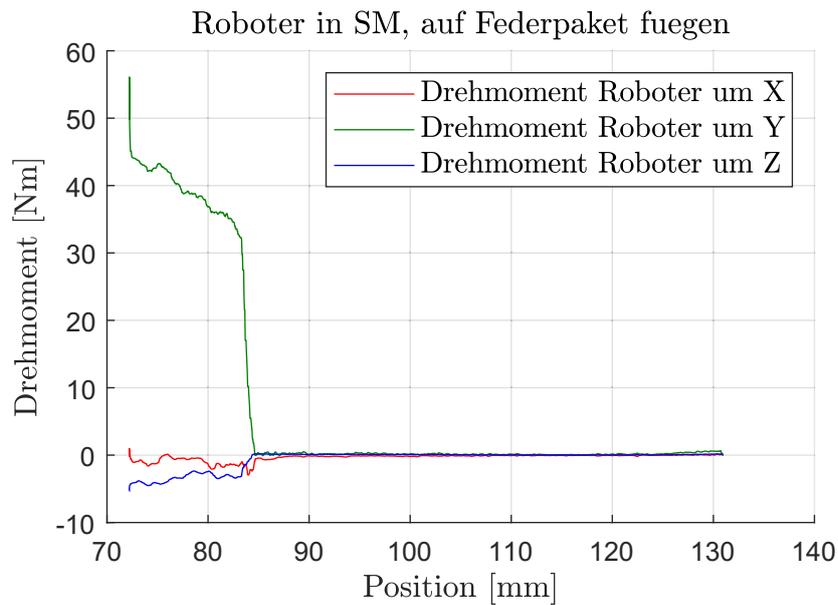


Abbildung 3.7: Abgeleitete Drehmomente beim Federpaket mit SM  
Quelle: eigene Ausarbeitung

Die über den Roboter abgeleiteten Drehmomente in Abbildung 3.6 bewegen sich ebenfalls vor dem Einfedern in der Nähe von 0 N m. Mit Beginn des Einfedervorganges steigt das Drehmoment um die Y-Achse stark an auf 56 N m am Ende des Einfedervorganges.

### 3.2.2 Fügeversuche Lager einpressen

Beim Fügeversuch Lager einpressen bewegt die Fügeachse das Roboterwerkzeug mit eingelegtem Lager erst in der Luft; dies findet im Bereich von 62 bis 42,5 mm statt. Bei 42,5 mm trifft das Lager auf den Getriebedeckel auf. Das Einpressen des Lagers findet bis zur Position 38,5 mm statt.

#### Fügeversuche mit Force Control:

Beim Betrachten der fünf einzelnen Fügekurven in Abbildung 3.8 ist festzustellen, dass eine Fügekraftkurve sich von den Restlichen abhebt. Dies ist darauf zurückzuführen, dass bei den Fügeversuchen dieselben Komponenten mehrmals verwendet wurden und es dadurch zu einem Abfall der Fügekräfte gekommen ist. Das Lager einpressen benötigt Fügekräfte zwischen 200 und 530 N, was sich mit dem erwarteten Bereich in Unterabschnitt 2.4.3 deckt.

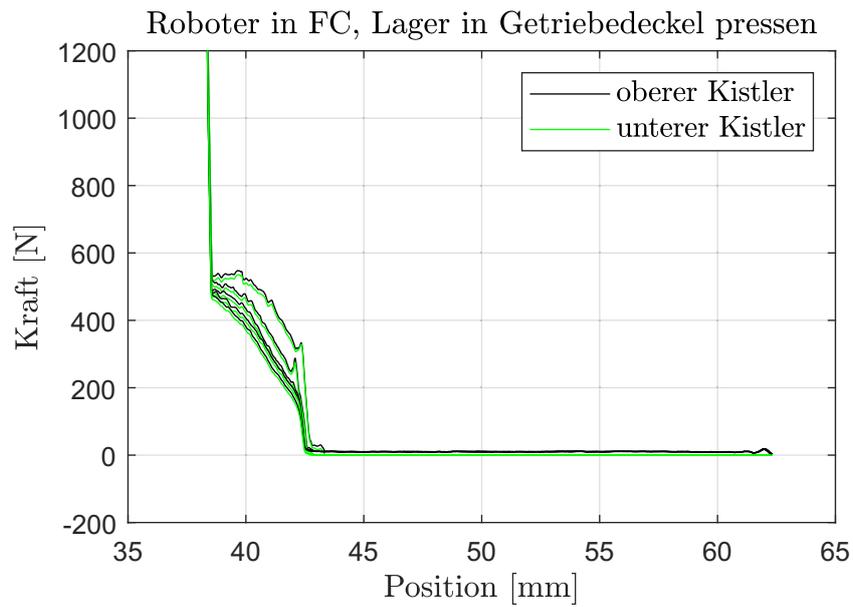


Abbildung 3.8: Fügekraftverläufe beim Lager einpressen mit FC  
Quelle: eigene Ausarbeitung

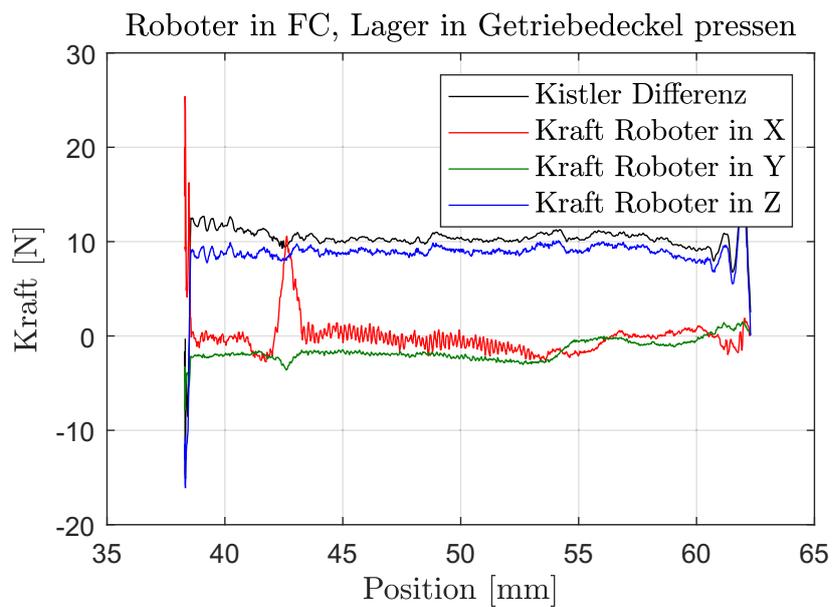


Abbildung 3.9: Abgeleitete Kräfte beim Lager einpressen mit FC  
Quelle: eigene Ausarbeitung

Die in Abbildung 3.9 dargestellten abgeleiteten Kräfte des Roboters zeigen ein konstantes Verhalten im Bereich von 8,5 bis 10 N nach dem Einschwingen

der Kraftspitzen bei den Kontaktpunkten. Die berechnete Z-Kraft des Roboters von 7 N bei der eingestellten Fügegeschwindigkeit von 5 mm/s zeigt ein abweichendes Bild. Die Spreizung der Differenzkraft der Kistler-Kraftsensoren und der Z-Kraft des Roboter-Kraftsensors nehmen während dem Lager einpressen zu. Ebenfalls erwähnenswert ist die Kraftspitze der X-Kraft des Roboters beim Auftreffen des Lagers auf den Getriebedeckel. Die Kraftspitzen am Ende des Fügevorganges lassen sich mit der Nachgiebigkeit der Aufnahme erklären. Die Betrachtung der Drehmomente in Anhang D.3 liefert bis auf eine Drehmomentspitze von  $-14 \text{ N m}$  um die X-Achse am Ende des Fügevorgangs keine erwähnenswerten Besonderheiten.

#### Fügeversuche mit Softmove:

Die fünf einzelnen Fügekurven in Anhang D.4 zeigen keine Ausreißer und eine geringe Streuung. Allerdings zeigt sich wiederum, dass die benötigte Fügekraft für das Einpressen des Lagers höher ist als bei der Anwendung von Force Control.

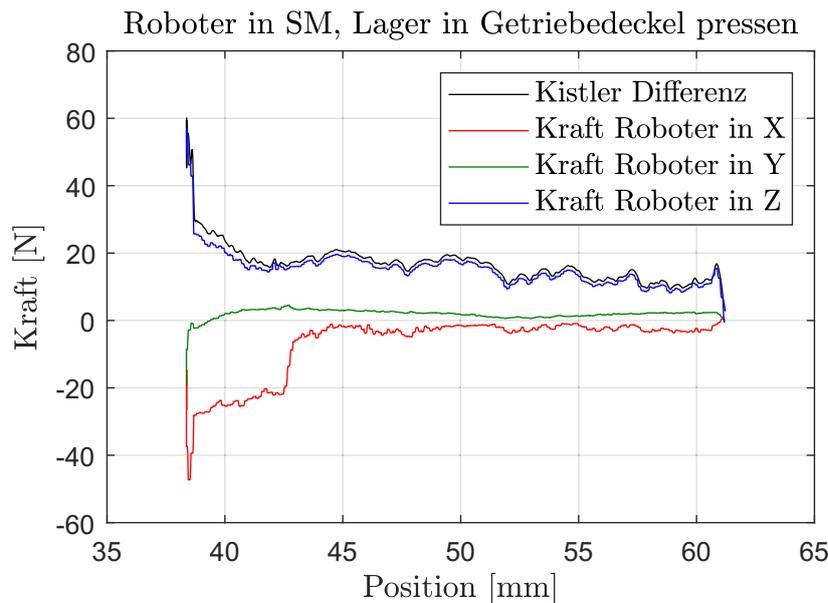


Abbildung 3.10: Abgeleitete Kräfte beim Lager einpressen mit SM  
Quelle: eigene Ausarbeitung

In Abbildung 3.10 ist erkennbar, dass die Differenzkraft der Kistler-Kraftsensoren und die Z-Kraft des Roboter-Kraftsensors während des gesamten Fügevorgangs ansteigen. Die Kräfte des Roboters bewegen sich während dem Einpressen des

Lagers zwischen 20 und 60 N in positiver Z- und negativer X- und Y-Richtung. Die Auswertung der Drehmomente in Anhang D.4 zeigt eine Drehmomentspitze von 24 N m um die Y-Achse. Ansonsten bewegen sich die Drehmomente in einem niedrigen Bereich von -1 und 0,5 N m.

### 3.2.3 Fügeversuche Getriebegehäuse mit Welle auf Deckel fügen

Beim Fügeversuch Getriebegehäuse mit Welle auf Deckel fügen positioniert der Roboter das Getriebegehäuse mit der Welle nach dem Abholen über dem Getriebedeckel. Die Fügeachse bewegt den Roboter bis zum Kontaktpunkt von der Welle mit dem Lager im Getriebedeckel bei der Position 101,8 mm. Anschließend erfolgt das Durchschieben der Welle durch das Lager; dies findet im Bereich von 101,8 bis 91 mm statt. Bei 90,5 mm trifft das Getriebegehäuse auf den Getriebedeckel auf. Bei Position 88,6 mm ist der Getriebedeckel vollständig eingepresst.

#### Fügeversuche mit Force Control:

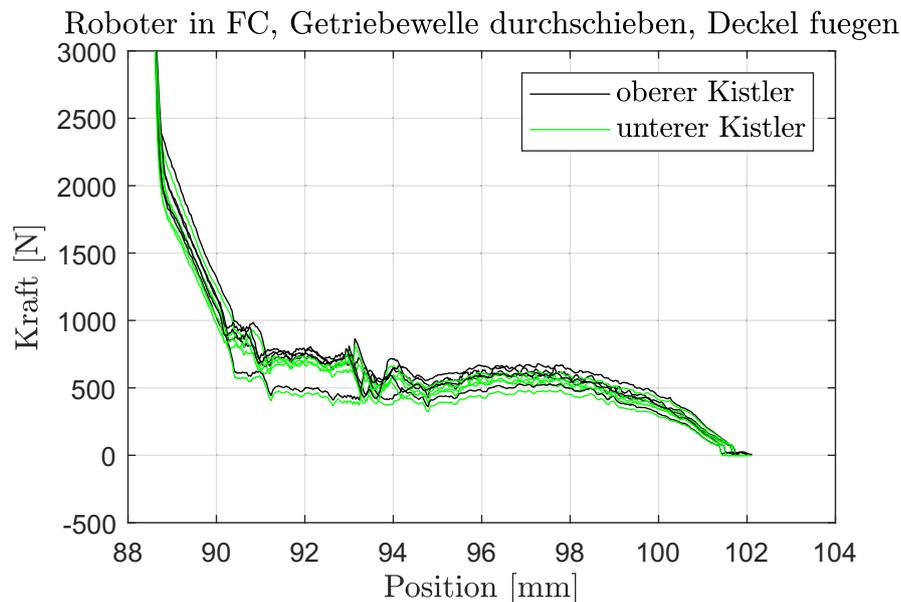


Abbildung 3.11: Fügekraftverläufe beim Getriebegehäuse mit Welle auf Deckel fügen mit FC

Quelle: eigene Ausarbeitung

Die fünf einzelnen Fügekurven in Abbildung 3.11 lassen erkennen, dass die benötigten Kräfte für das Durchschieben der Getriebewelle maximal 650 N betragen. Dies ist grundsätzlich zu hoch, da die Welle von Hand durch das Lager verschiebbar sein sollte. Weiters ist zu erkennen, dass für das Fügen des Getriebedeckels Fügekräfte im Bereich von 820 bis 2450 N auftreten und sich somit im erwarteten Bereich bewegen. Eine Fügekurve kann als Ausreißer gesehen werden; diese wird für die weitere Berechnung daher entfernt. Die Streuung beim Durchschieben des Lagers kann damit erklärt werden, dass dieselben Komponenten mehrmals verbaut wurden, da nur eine gekürzte Getriebewelle verfügbar war.

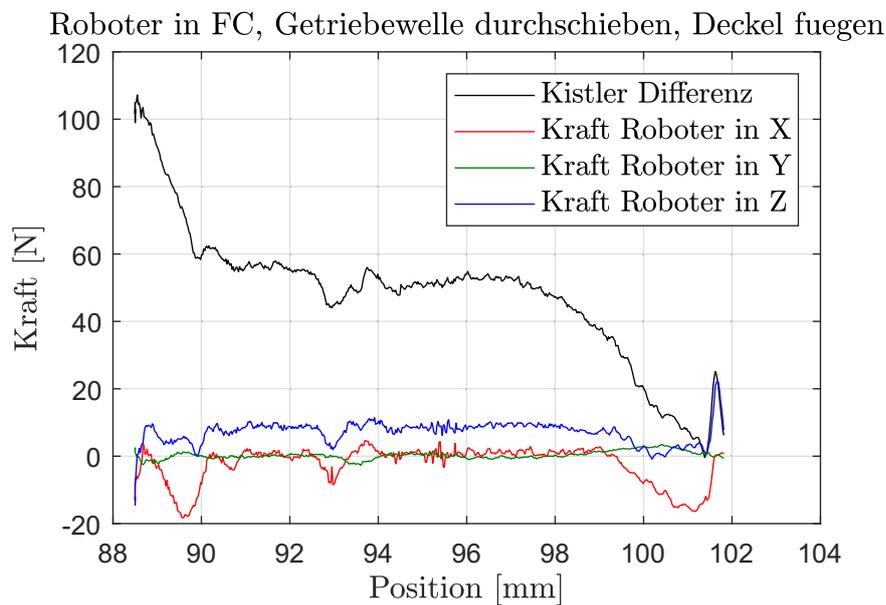


Abbildung 3.12: Abgeleitete Kräfte beim Getriebegehäuse mit Welle auf Deckel fügen mit FC  
Quelle: eigene Ausarbeitung

Das Diagramm in Abbildung 3.12 zeigt, dass sich die Z-Kraft des Roboter-Kraftsensors hauptsächlich im Bereich von 6 bis 11 N sich bewegt. Die berechneten Z-Kraft des Roboter von 4,2 N bei der eingestellten Fügegeschwindigkeit von 3 mm/s wird dabei überschritten. Auffallend ist, dass die Spreizung der Differenzkraft der Kistler-Kraftsensoren und der Z-Kraft des Roboter-Kraftsensors deutlich auseinander geht und die Differenzkraft bis auf 107 N ansteigt. Die X- und Y-Kräfte bleiben dabei mit bis  $-20$  N auf einem niedrigen Niveau. Die Drehmomente beim Roboter in Anhang D.5 bewegen sich, bis auf eine Spitze

von  $-12 \text{ N m}$  am Ende des Fügevorganges im Bereich von  $-6$  bis  $5,5 \text{ N m}$ .

**Fügeversuche mit Softmove:**

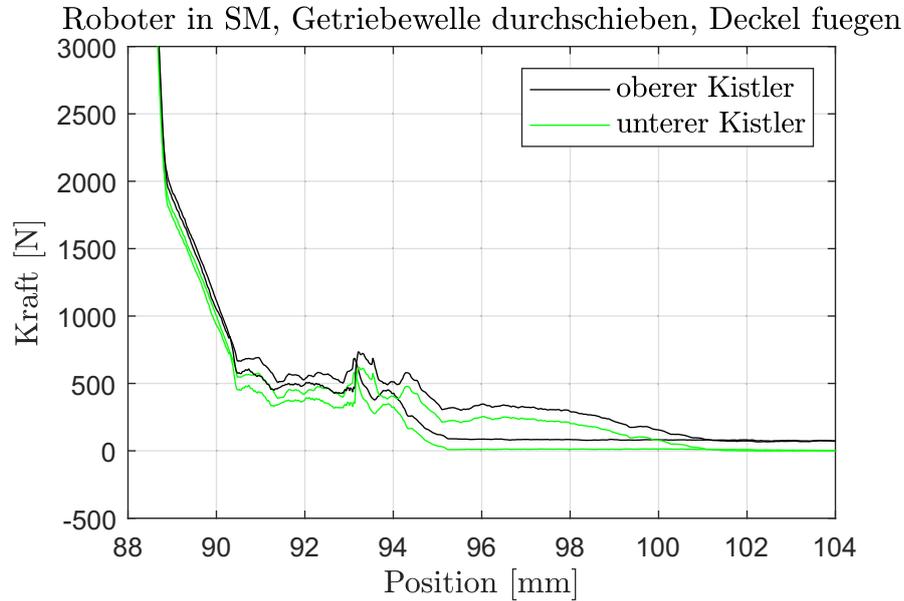


Abbildung 3.13: Fügekraftverläufe beim Getriebegehäuse mit Welle auf Deckel fügen mit SM  
Quelle: eigene Ausarbeitung

Mit Softmove konnten bei den Fügeversuchen mit dem Getriebegehäuse nur zwei Durchgänge durchgeführt werden. Zudem mussten die Parameter wie in Tabelle 3.1 ersichtlich verändert werden, da der Roboter sich ohne externe Einflüsse nach unten bewegt hat. In Abbildung 3.13 ist ersichtlich, dass bei einem Versuch die Kräfte beim Durchschieben der Getriebewelle im Bereich von 80 bis 350 N und beim anderen Versuch konstant bei 80 bis 100 N liegen. Beim Getriebedeckel fügen ist die Streuung minimal.

Wie in Abbildung 3.14 dargestellt, steigen die Differenzkraft der Kistler- Kraftsensoren und die Z-Kraft des Roboter-Kraftsensors bis auf 200 bzw. 170 N an. Die Spreizung der Kräfte ändert sich dabei nur gering. Die X-Kraft des Roboters bewegt sich im Bereich von  $-44$  bis  $57 \text{ N}$ , die Y-Kraft in der Nähe von  $0 \text{ N}$ . Die Auswertung der Drehmomente in Anhang D.6 zeigt einen Anstieg des Drehmoments um die Y-Achse gegen Ende des Durchschiebens der Getriebewelle bis zu einer Drehmomentspitze von  $59 \text{ N m}$  am Ende des Fügevorgangs des Getriebedeckels.

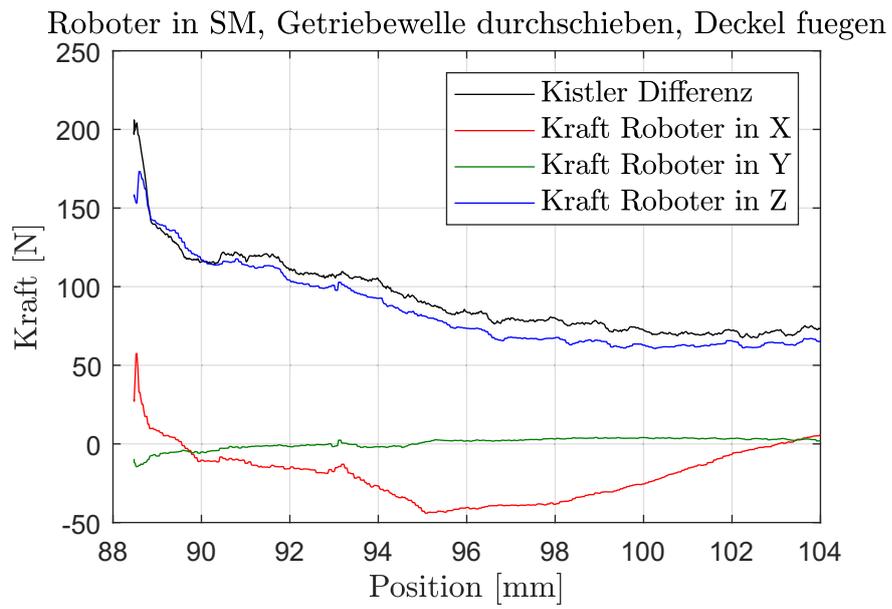


Abbildung 3.14: Abgeleitete Kräfte beim Getriebegehäuse mit Welle auf Deckel fügen mit SM  
Quelle: eigene Ausarbeitung

## 4 Interpretation

In diesem Kapitel werden die in Kapitel 3 dargestellten Ergebnisse mit den in Abschnitt 1.2 und Unterabschnitt 2.1.1 gestellten Anforderungen verglichen und die daraus ableitbaren Zusammenhänge interpretiert.

Die Linearitätsprüfung in Unterabschnitt 3.1.2 weist den ABB Kraftsensor inklusive der Umrechnung in das kartesische Koordinatensystem, bei einer Toleranz von mindestens  $\pm 8\text{ N}$ , als fähiges Messsystem aus. Dies gilt für den Fall, dass der Roboter keinen weiteren Einflüssen außer dem Kontakt mit der Fügeachse ausgesetzt ist. Somit kann auch eine Kompensation der Fügekräfte mit dieser Genauigkeit erfolgen. Ob dies wiederum ausreichend ist, hängt von dem Nennmaß und der zur Verfügung stehenden Toleranz des Fügeprozesses ab. Eine zufriedenstellende Erklärung, warum der Roboter die berechnete Kraft der letzten Kraftstufe der Linearitätsprüfung (siehe Unterabschnitt 2.4.1[S.31]) von  $98\text{ N}$  mit der gemessenen Kraft von  $70\text{ N}$  (siehe Unterabschnitt 3.1.2[S.39]) nicht erreicht hat, kann nicht gegeben werden. Dies bedarf weiterer Untersuchungen.

Die Betrachtung der Fügeprozesse mit dem Federpaket in Unterabschnitt 3.2.1 zeigt, dass sich mit Force Control die Z-Kraft des Roboter-Kraftsensors unter Belastung der Federkraft nur unwesentlich ändert und sich in einem konstanten Bereich bewegt. Dieser Bereich deckt sich zudem mit den berechneten Fügekräften und ist weit von der Belastungsgrenze des Roboters mit  $F_h = 196,2\text{ N}$  entfernt. Auch die Kraftspitzen der Kontaktpunkte sind entsprechend, sodass diese vom Roboter aufgenommen werden können. Die Spreizung der Differenzkraft der Kistler-Kraftsensoren und der Z-Kraft des Roboter-Kraftsensors kann mit Reibungseffekten im Federpaket erklärt werden. Allerdings steigt speziell das Drehmoment um die Y-Achse beim Kontakt des Roboterwerkzeugs mit dem Federpaket sprunghaft an und bleibt bis zum Ende des Fügevorganges auf diesem Niveau. Eine Ursache dafür kann der Umstand sein, dass bei Orientierungsänderungen um die Y-Achse, aufgrund der Konfiguration des Roboters während des Fügens bei den meisten Roboter gelenken die Position nachgeregelt

werden muss (vgl. Admittanzregelung in Unterabschnitt 2.2.3) und dadurch ein höheres Drehmoment anliegt. Dies kann allerdings nicht weiter belegt werden. Da das Drehmoment um die Y-Achse die sechste Achse nur in geringem Maße belastet, ist als maximales Belastungsdrehmoment das Drehmoment  $M_{4,5} = 36,3 \text{ N m}$  anwendbar. Dies wird hierbei nicht überschritten und liegt im zulässigen Bereich. Somit ist Force Control für Anwendungen, die mit dem Federpaket vergleichbar sind, als Unterstützungsverfahren geeignet. Ebenso lässt sich eine Kompensation der Fügekräfte realisieren, welche entweder direkt mit den Kräften des Roboterkraftsensors oder mit einer zuvor berechneten fixen Kompensation arbeitet.

Bei der Anwendung von Softmove in Kombination mit dem Federpaket werden die in Unterabschnitt 2.3.1 beschriebenen Eigenschaften von Softmove deutlich. Die Z-Kraft des Roboter-Kraftsensors ändert sich während des gesamten Fügeweges. Dabei fällt diese vorerst gegen  $0 \text{ N}$  ab und steigt dann, speziell bei Kontakt mit dem Federpaket, deutlich an. Die Kräfte überschreiten dabei allerdings nicht die zulässige Belastungskraft, jedoch steigt das Drehmoment um die Y-Achse auf  $56 \text{ N m}$  an, was die Belastungsgrenze übersteigt. Die Gründe dafür können wie bei Force Control mit der Anordnung des Roboters erklärt werden. Weitere Gründe sind die Einspannung des Roboterwerkzeuges zwischen Fügewerkzeug und Federpaket sowie der Umstand, dass die Achsen 4 und 6 nahezu gleich ausgerichtet und somit nahe an einer Singularität sind. Dadurch werden, wie in Unterabschnitt 2.3.1 beschrieben, hohe Kräfte und Momente am Endeffektor bewirkt, wobei sich gleichzeitig die Momente in den Gelenken nur minimal ändern. Auffallend ist im Vergleich zu den Versuchen mit Force Control auch, dass die Fügekurve der unteren Kraftmessung, also abzüglich der durch den Roboter abgeleiteten Kraft, bei Softmove auf einem  $150 \text{ N}$  höheren Niveau liegt. Ein Erklärungsversuch dafür ist, dass durch die höheren Kräfte und Momente, die auf den Roboter und somit auch auf das Federpaket wirken, höhere Reibeffekte verursacht werden. Dazu sind jedoch weiterführende Untersuchungen notwendig.

Die Fügeversuche Lager fügen mit Force Control zeigen ein mit den Erkenntnissen des Federpaket fügen vergleichbares Bild. Die Z-Kraft des Roboter-Kraftsensors ist höher als die berechnete Kraft, dafür aber konstant und stimmt bis auf den konstanten Faktor mit der Differenzkraft der Kistler-Kraftsensoren

überein. Dies ermöglicht eine Kompensation mit den Daten des Kraftsensors. Ob eine berechnete, fixe Kompensation möglich ist, hängt von den Maß- und den Toleranzanforderungen eines möglichen Produktes ab. Weiters zeigt die Kraftspitze der X-Kraft in Abbildung 3.9 die Vorteile einer Kraft-/Lageregelung auf. Die Kraftspitze deutet darauf hin, dass die Positionierung des Lagers bzw. des Roboterfügewerkzeug in X-Richtung nicht exakt ist. Das Lager wird auf die Fase gedrückt, wodurch eine Kraft in X-Richtung entsteht. Dadurch bewegt sich der Roboter in X-Richtung, bis die Kraft wieder abfällt und die Positionierung korrigiert ist.

Bei der Anwendung von Softmove beim Lager einpressen sind die über den Roboter abgeleiteten Kräfte und Momente unterhalb der Belastungsgrenzen des Roboters. Die Z-Kraft des Roboter-Kraftsensors steigt, bis auf die Kraftspitze am Ende des Fügevorgangs, leicht an. Es zeigt sich jedoch wieder, dass die benötigten Kräfte für das Einpressen des Lagers deutlich höher als bei Force Control sind. Durch die ansteigende Z-Kraft des Roboter-Kraftsensors ist eine berechnete fixe Kompensation nur bedingt möglich. Für einfache Fügeoperationen, wie das Lager einpressen mit großen Toleranzen der Fügekräfte, wäre der Einsatz von Softmove möglich.

Bei den Versuchen mit der Getriebewelle und dem Deckel zeigt sich, dass die Spreizung zwischen Differenzkraft der Kistler-Kraftsensoren und Z-Kraft des Roboter-Kraftsensors deutlich auseinander geht. Ob dies mit Effekten der Reibung während dem Durchschieben der Welle und der Umformarbeit während des Deckelfügens zusammenhängt, müssen weitere Fügeversuche in derselben Konstellation ohne Roboter klären. Die am Roboter auftretenden Kräfte und Momente sind jedenfalls unter den Belastungsgrenzen. Der Verlauf der Z-Kraft des Roboter-Kraftsensors ist nicht so konstant wie beim Federpaket oder beim Lager einpressen, aber eine Kompensation der Fügekraft mithilfe der Daten des Kraftsensors ist möglich. Bei einer berechneten fixen Kompensation kommt es auf die Anforderungen an die Krafttoleranzen des Anwendungsfalls an.

Bei Betrachtung der abgeleiteten Kräfte des Roboters bei der Anwendung von Softmove in Abbildung 3.14 zeigt sich ein konträres Bild zu den Versuchen mit Force Control. Die Spreizung der Differenzkraft der Kistler-Kraftsensoren und die Z-Kraft des Roboter-Kraftsensors ist um ein vielfaches kleiner, wenn auch insgesamt auf einem höheren Niveau. Dies stellt die zuvor aufgestellte

Theorie der Reibungs- und Umformeffekte in Frage. Dies kann anhand der vorhandenen Daten nicht vollständig geklärt werden. Bei Betrachtung der über den Roboter abgeleiteten Kräfte und Momente zeigt sich, dass die Z-Kraft des Roboter-Kraftsensors nahe an die Belastungsgrenze und das Drehmoment mit 59 N m deutlich über die Belastungsgrenze kommt. Die Z-Kraft des Roboter-Kraftsensors steigt von 61 bis 170 N an, dies macht eine berechenbare Kompensation unmöglich.

Zusammenfassend betrachtet können zu diesen Versuchen folgende Aussagen getroffen werden: Eine Nachführung des Roboters in vergleichbaren Fügeprozessen ist mit Force Control möglich. Bei der Anwendung von Softmove müssen jeweils anwendungsspezifische Untersuchungen durchgeführt werden, um die Verwendbarkeit zu bestätigen. Eine fähige Kompensation kann mit den Messdaten von Force Control ebenfalls durchgeführt werden. Bei Softmove ist dies abhängig von den zur Verfügung stehenden Toleranzen und muss ebenfalls in den Untersuchungen bestätigt werden.

# 5 Zusammenfassung und Ausblick

In dieser Arbeit wurde die Interaktion zwischen einem Industrieroboter und einer Fügeachse während einer Fügeoperation untersucht. Mit den in Kapitel 1 beschriebenen Zielen, ob sich eine solche Nachführung grundsätzlich realisieren lässt und ob die Leistungsgrenzen des Roboters dabei nicht überschritten werden. Ebenfalls galt es zu untersuchen, ob eine fähige Kompensation der abgeleiteten Kräfte über den Roboter möglich ist.

Dazu wurde in Kapitel 2 der Versuchsaufbau beschrieben, der für die Durchführung der Untersuchungen dienen soll. Anschließend wurden die Regelungskonzepte Lageregelung, Kraftregelung und hybride Kraft-/Lageregelung beschrieben. Diese Konzepte bilden die Grundlage der zur Verfügung stehenden Unterstützungsverfahren des Roboters. Nachfolgend sind die Eigenschaften und die Anwendung der Unterstützungsverfahren beschrieben sowie abschließend der Aufbau und der Zweck der Fügeversuche.

Kapitel 3 startet mit der Verifikation der Messwerte des Versuchsaufbaus. Die Linearitätsprüfung ergibt dabei, dass die Kraftmessung des Roboterkraftsensors ab einer Toleranz von  $\pm 8\text{ N}$  die Fähigkeitskriterien der thyssenkrupp Presta erfüllt. Anschließend sind die Ergebnisse der unterschiedlichen Fügeversuche dargestellt und beschrieben.

Die Interpretation in Kapitel 4 stellt fest, dass Force Control als Unterstützungsverfahren für eine Nachführung während eines Fügeprozesses geeignet ist und eine Kompensation der abgeleiteten Fügekraft mit den Kraftsensorwerten des Roboters erfolgen kann. Je nach Anwendungsfall kann auch eine zuvor berechnete fixe Kompensation verwendet werden. Softmove erfüllt die Anforderungen an die Belastungsgrenzen bei 2 von 3 Fügeversuchen nicht und eine Kompensation der Fügekräfte ist nur bedingt möglich. Ob Softmove verwendbar ist muss anwendungsbezogen geprüft werden.

Als Ausblick kann folgende Vorgehensweise definiert werden. Bei den Versuchen konnte festgestellt werden, dass bei der Anwendung von Force Control Abweichungen zwischen der berechneten und der gemessenen Kraft vorhanden und diese nicht zufriedenstellend erklärbar sind. Dazu sind weitere Untersuchungen und Analysen, aber auch Abklärungen mit dem Roboterhersteller notwendig. Dies gilt auch für den Umstand, dass die Spreizung der Differenzkraft der Kistler-Kraftsensoren und der Z-Kraft des Roboter-Kraftsensors bei Force Control während des Fügeprozesses größer wird und bei Softmove nicht. Die Begründung des hohen Drehmoments um die Y-Achse ist weiter zu untersuchen, damit diese ausreichend belegt werden kann. Um die bessere Anwendungsmöglichkeit von Softmove zu prüfen, wäre ein Versuchsaufbau notwendig, bei dem speziell auf die Lage und Positionierung der Robotergelenke geachtet wird. Damit lässt sich darstellen, ob auch mit Softmove eine Nachführung innerhalb der Belastungsgrenzen mit fähiger Kompensation der Kräfte realisierbar ist. Die Untersuchungen mit den Unterstützungsverfahren Sensor Synchronization und Externally Guided Motion wurden nicht durchgeführt und sind als ein zentraler Punkt des Ausblicks zu sehen.

# Literatur

- [1] ABB Robotics (Hrsg.): *Application manual Controller Software 6.11 Dokument Nr.: 3HAC050798-003 Revision: M.* 2020.
- [2] ABB Robotics (Hrsg.): *Application manual Externally Guided Motion 6.11 Dokument Nr.: 3HAC073319-003 Revision: C.* 2020.
- [3] ABB Robotics (Hrsg.): *Application manual Force Control 6.10 Dokument Nr.: 3HAC050377-001 Revision: F.* 2019.
- [4] ABB Robotics (Hrsg.): *Application manual SoftMove 6.07 Dokument Nr.: 3HAC050977-001 Revision: D.* 2018.
- [5] ABB Robotics (Hrsg.): *Technisches Referenzhandbuch - RAPID Instruktionen, Funktionen und Datentypen 6.08 Dokument Nr.: 3HAC050917-003 Revision: H.* 2018.
- [6] Beckhoff Automation (Hrsg.) [Online]: *C6930 / Schaltschrank-Industrie-PC*. 2021. URL: <https://www.beckhoff.com/de-ch/produkte/ipc/pcs/c69xx-kompakte-industrie-pcs/c6930.html> (besucht am 20.06.2021).
- [7] John J. Craig. *Introduction to robotics: mechanics and control*. 3. edition. Upper Saddle River, NJ: Pearson Prentice Hall, 2005. 400 S. ISBN: 978-0-201-54361-2.
- [8] EtherCAT Technology Group (Hrsg.) [Online]: *EtherCAT Technology*. 2021. URL: <https://www.ethercat.org/de/technology.html> (besucht am 13.07.2021).
- [9] Franziska Leitner. „Roboter gestützter Fügeprozess“. Bachelorarbeit. Dornbirn: FH Vorarlberg, Aug. 2020.
- [10] Georg Malin. „Virtuelle Roboterinbetriebnahme mittels Simulation und Offline-Programmierung“. Bachelorarbeit. Dornbirn: FH Vorarlberg, Juni 2019.

- [11] Kistler (Hrsg.): *Quarz-Kraftmesselemente Betriebsanleitung 9301B-9371B*. 2013.
- [12] Kevin M. Lynch und Frank C. Park. *Modern robotics: mechanics, planning, and control*. OCLC: 983881868. Cambridge: Cambridge University Press, 2017. 528 S. ISBN: 978 1 107 15630 2.
- [13] Massimo Anselmi, thyssenkrupp Presta AG (Hrsg.) [Intern]: *CD-00015-DE\_ Statistische Auswerteverfahren Version 11*. 2019. (Besucht am 22.03.2021).
- [14] Promess Montage (Hrsg.) [Online]: *Elektro-mechanische Fügeachse UFM Klassik*. 2021. URL: <https://www.promessmontage.de/produkte/fuegemodule/ufm-klassik> (besucht am 08.03.2021).
- [15] Roland Schnurr (Hrsg.) [Online]: *MSA - Messsystemanalyse*. 2021. URL: <https://www.sixsigmablackbelt.de/msa-messsystemanalyse-messmittelfaehigkeit/> (besucht am 12.07.2021).
- [16] J. Kenneth Salisbury. „Active stiffness control of a manipulator in cartesian coordinates“. In: *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*. 1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes. 1980, S. 95–100. DOI: 10.1109/CDC.1980.272026.
- [17] Bruno Siciliano (Hrsg.) *Springer handbook of robotics: with 84 tables*. Berlin: Springer, 2008. 1611 S. ISBN: 978 3 540 23957 4.
- [18] thyssenkrupp Presta AG [Intern]: *TV-01942-Erstellen Roboterprogramm mit SPS Master V3, gültig ab 30.09.2015, Eschen*. In der Datenbank Xeri verfügbar. 2015. (Besucht am 22.03.2021).
- [19] Wolfgang Weber. *Industrieroboter: Methoden der Steuerung und Regelung*. 4., aktualisierte Auflage. München: Hanser, 2019. 237 S. ISBN: 978-3-446-45952-6.
- [20] Günter Wellenreuther und Dieter Zastrow. *Automatisieren mit SPS - Übersichten und Übungsaufgaben: von Grundverknüpfungen bis Ablaufsteuerungen ; Wortverarbeitungen und Regelungen*. 7. Auflage. Wiesbaden: Springer Fachmedien, 2016. 413 S. ISBN: 978 3 658 11199 1.
- [21] Herbert Wittel u. a. *Roloff/Matek Maschinenelemente: Normung, Berechnung, Gestaltung*. 19., überarbeitete und erweiterte Auflage. Wiesbaden: Vieweg+Teubner Verlag, 2009. ISBN: 978-3-8348-0689-5.

# Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Dornbirn, am 06.08.2021

Georg Malin

# Anhang

## A Datenblätter

- A.1 Datenblatt ABB IRB 2600 . . . . .
- A.2 Datenblatt Ladungsverstärker Kistler 5074A . . . . .
- A.3 Datenblatt Heidenhain Längenmesslineal . . . . .
- A.4 Datenblatt Beckhoff Encoder Simulationskarte . . . . .

## B Programm Code Dateien

- B.1 Matlab Code Linearität Kistler Kraftsensoren . . . . .
- B.2 Matlab Code Linearität Kraftsensor oben in der Fügeachse und Kraftsensor ABB . . . . .
- B.3 Matlab Code Auswertung Fügeversuche . . . . .

## C MSA Linearität

- C.1 Linearitätsprüfung Kraftsensoren in der Fügeachse . . . . .
- C.2 Linearitätsprüfung Kraftsensor oben in der Fügeachse und Kraftsensor ABB . . . . .

## D Diagramme der Fügeversuche

- D.1 Force Control Federpaket . . . . .
- D.2 Softmove Federpaket . . . . .
- D.3 Force Control Lager einpressen . . . . .
- D.4 Softmove Lager einpressen . . . . .
- D.5 Force Control Getriebewelle durchschieben, Deckel fügen . . . . .
- D.6 Softmove Getriebewelle durchschieben, Deckel fügen . . . . .

# A Datenblätter

1. Datenblatt ABB IRB 2600 (2 Seiten)
2. Datenblatt Ladungsverstärker Kistler 5074A (3 Seiten)
3. Datenblatt Heidenhain Längenmesslineal (2 Seiten)
4. Datenblatt Beckhoff Encoder Simulationskarte (1 Seite)



ROBOTICS

## IRB 2600

ABB further extends its mid range robot family



The IRB 2600 is the second model in the sharp generation range with enhanced and new capabilities. It is a compact robot with a high payload capacity. The design has been optimized for targeted applications like arc welding, material handling and machine tending. The IRB 2600 is available in three variants, with options for floor, wall, shelf, tilted or inverted mounting configurations.

### Sharp accuracy

With the best accuracy in its class, the IRB 2600 can help you increase output with higher process speeds and lower scrap rates, resulting in improved productivity. This is particularly useful in process applications, such as arc welding. The high accuracy is achieved by use of the patented TrueMove™ motion control software.

### Short cycle times

Thanks to the compact and optimized design resulting in a low weight, the IRB 2600 can cut the cycle times of the industry benchmark by up to 25%. The patented QuickMove™ motion control software ensures that the maximum acceleration achievable is highest in its class, together with high maximum speeds. The benefit is increased production capacity and high productivity.

### Large working range

The combination of large working range and flexible mounting makes it possible to reach machines to be served without interfering with auxiliary equipment. By optimising the robot placement, you'll benefit from higher productivity. Flexible mounting is also very useful when you are simulating the best position for your application.

### Compact design

The IRB 2600 has the same small foot print as IRB 4600. With the IRB 2600 you can create your production cell with reduced floorspace by placing the robot closer to the served machines. The robots' design makes it easier for the lower arm to reach straight downwards.

### Best protection available

ABB has the most comprehensive protection program for industrial robots on the market. The IRB 2600 has IP67 as standard and FoundryPlus 2 as option.

### Main applications

- Arc Welding
- Assembly
- Material Handling
- Machine Tending
- Material Removal
- Cleaning/Spraying
- Dispensing
- Packing

## Specification

Robot version	Reach (m)	Handling capacity (kg)	Wrist torque capacity (Nm)	Axis 4 & 5	Axis 6
IRB 2600-20/1.65	1.65	20	36.3	16.7	
IRB 2600-12/1.65	1.65	12	21.8	10	
IRB 2600-12/1.85	1.85	12	21.8	10	
Number of axes	6+3 external (up to 36 with MultiMove)				
Protection	Standard IP67; optional FoundryPlus 2				
Mounting	Floor, wall, shelf, tilted, inverted				
Controller	IRC5 Single Cabinet				

## Performance (according to ISO 9283)

	Position repeatability	Path repeatability
IRB 2600-20/1.65	0.04 mm	0.13 mm
IRB 2600-12/1.65	0.04 mm	0.14 mm
IRB 2600-12/1.85	0.04 mm	0.16 mm

## Technical information

### Electrical Connections

Supply voltage	200-600 V, 50/60 Hz
Energy consumption	3.4 kW

### Physical

Robot base	676 x 511 mm
Robot height	
IRB 2600-20/1.65	1382 mm
IRB 2600-12/1.65	1382 mm
IRB 2600-12/1.85	1582 mm
Robot weight	
IRB 2600-20/1.65	272 kg
IRB 2600-12/1.65	272 kg
IRB 2600-12/1.85	284 kg

### Environment

Ambient temperature for mechanical unit	
During operation	+5°C (41°F) up to +50°C (122°F)
During transportation and storage	-25°C (13°F) up to +55°C (131°F)
During short periods (max. 24 h)	up to +70°C (158°F)
Relative humidity	Max. 95%
Noise level	Max. 69dB(A) IRB2600-12/1.85 Max. 72dB(A) IRB2600-20/1.65; IRB2600-12/1.65
Safety	Double circuits with supervision, emergency stops and safety functions, 3-positions enable device.
Emission	EMC/EMI-shielded
Options	Foundry Plus 2

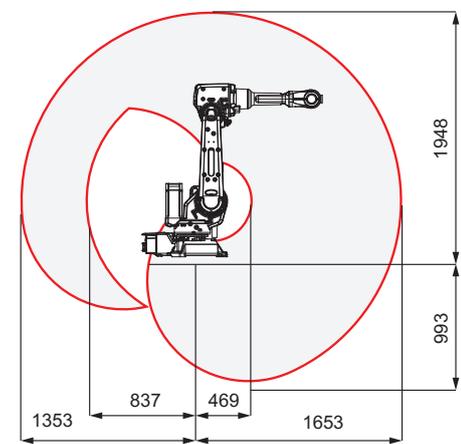
Data and dimensions may be changed without notice.

## Movement

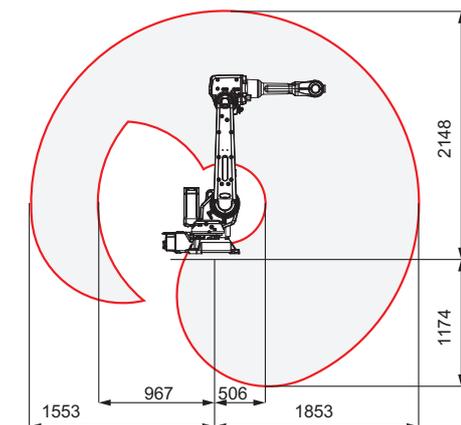
Axis movement	Working range	Axis max speed
Axis 1 rotation	+180° to -180°	175°/s
Axis 2 arm	+155° to -95°	175°/s
Axis 3 arm	+75° to -180°	175°/s
Axis 4 rotation	+400° to -400°	360°/s
	Max. rev: +251 to -251	
Axis 5 band	+120° to -120°	360°/s
Axis 6 turn	+400° to -400°	500°/s
	Max. rev: +274 to -274	

A supervision function prevents overheating in applications with intensive and frequent movements.

### Working range, IRB 2600-20/1.65, IRB 2600-12/1.65



### Working range, IRB 2600-12/1.85



[abb.com/robotics](http://abb.com/robotics)

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB. Copyright © 2020 ABB. All rights reserved.

## Digitaler industrieller Ladungsverstärker

Typ 5074A...

### Ladungsverstärker für Industrial Ethernet basierte Systeme

Der Kistler Typ 5074A... ist ein universeller industrieller Messverstärker für piezoelektrische Sensoren und erlaubt die zuverlässige und präzise Erfassung dynamischer und quasistatischer Vorgänge. Der Verstärker unterstützt die wichtigsten industriellen Ethernet Standards und bietet abhängig davon sowohl Echtzeitmessung mit schnellsten Zykluszeiten als auch einige Sonderfunktionen zur anwendungsspezifisch konfigurierbaren Datenerfassung.



- Varianten mit 1, 2, 3 oder 4 Kanälen
- weiter Messbereich von 20pC bis 1 000 000pC
- Varianten für EtherCat, ProfiNet und Ethernet/IP
- variables Prozessabbild für optimierte Netzauslastung
- interne 24-Bit-Datenerfassung
- Echtzeitausgabe in Zyklen bis 100µs pro Kanal
- Ausgabe mit bis zu 50kSps pro Kanal durch Unterstützung von Oversampling und Synchronisation
- flexible Tiefpassfilter und zuschaltbarer Hochpassfilter
- Erfassung von Spitzenwerten
- Berechnung des Integralwerts
- Skalierung von Sensorsignalen
- Statusanzeigen für Netzwerk- und Kanalstatus
- Handshake für Änderungen des Gerätestatus
- Konfiguration über Standard-Programmierschnittstelle der Steuerung, keine zusätzliche Software erforderlich
- kaskadierbare Ausführung; mehrere Busteilnehmer nacheinander anschliessbar
- Schutzklasse IP67 mit FKM/FPM Dichtelementen

#### Beschreibung

Der Kistler Typ 5074A... bietet echtzeitfähige Erfassung von piezoelektrischen Signalen in den meist verbreiteten industriellen Ethernet Typen und kommuniziert ausschliesslich über diese Schnittstelle.

Der 5074A... bietet eine hohe Performance und überträgt bis zu 50kSps pro Kanal. So ist auch die Echtzeitübertragung von Messdaten bei Zykluszeiten bis minimal 100µs möglich, wodurch auch kritische Prozessregelungen realisiert werden können. Auf dem 5074A... werden keine Daten dauerhaft gespeichert. Sämtliche Einstellungen liegen auf der Steuerung, wodurch die Datenhoheit auch garantiert beim Kunden liegt. Der einstellbare Tiefpassfilter erlaubt die gezielte Glättung von Störsignalen bereits im Verstärker. Mit dem zuschaltbaren Hochpassfilter<sup>1)</sup> können schnelle Prozesse ohne Unterbruch der Messung auch im Dauerbetrieb überwacht werden. Dank der frühen Digitalisierung der analogen Messgrössen bereits im Verstärker und der integrierten galvanischen Trennung liefert der 5074A... störungsresistent und zuverlässig Messwerte an das übergeordnete System.

#### Anwendung

Der Typ 5074A... wurde gezielt für den Einsatz in modernen, schnellen industriellen Ethernet Systemen entwickelt. Dank der hohen Performance eignet sich der Verstärker beispielsweise für Echtzeit-Regelungsaufgaben in Füge- und Schneidprozessen, aber auch für hochauflösende Prozessaufzeichnung bei Injektoren und Druckpulsationsmessungen. Dank Schutzklasse IP67 und Dichtelementen aus FKM/FPM ist auch der dezentrale Einsatz ausserhalb des Schaltschranks unter erschwerten Bedingungen mit flüssigen Medien, diversen Gasen und auch einigen Laugen möglich.

#### Beeinflussung und Anpassung des Prozessabbildes

Die Verarbeitung vieler Analogwerte unter kurzen Zykluszeiten kann die Auslastung des Netzwerks massiv erhöhen. Der Typ 5074A... ermöglicht die flexible Anpassung des Prozessabbildes und damit eine Optimierung der Netzauslastung entsprechend der Kundenbedürfnisse.

Mögliche Massnahmen zur Reduktion des Prozessabbildes sind:

- Verwendung von 16-Bit Werten anstelle 32-Bit
- Ausschalten von nicht benötigten Funktionen (Flieskomma, Integral und Spitzenwerte)

Änderungen an der Messkanal-Konfiguration, werden frühestens auf dem nächstfolgenden Messzyklus übernommen und mit dem Statusbit „Parameter active“ an den Master bestätigt.

#### Skalierung der Eingangssignale

Über den Parameter „Scaling Divisor“ wird die Eingangsgrösse skaliert. Typischerweise wird für den Divisor die Empfindlichkeit des Messelementes gemäss Kalibrierschein eingetragen. Die übertragenen 32-Bit REAL Messwerte entsprechen damit der physikalischen Grösse gemäss Kalibrierung des Messelementes. Die 16-Bit Integer Werte werden zusätzlich über den Bereichs-

<sup>1)</sup> Die Filterfrequenz des Hochpassfilters ist abhängig vom eingestellten Messbereich

endwert skaliert und entsprechen dann klassischen Eingangswerten.

#### Messbereichseinstellung

Die Messbereichseinstellung erfolgt über den Parameter „Range“ und hat zwei grundlegende Funktionen. Der 5074A verfügt über 3 interne Messsegmente pro Kanal. Entsprechend der Bereichseinstellung wird das bestmögliche Segment gewählt um höchste Präzision über mehrere Dekaden zu gewährleisten. Zusätzlich normiert der eingestellte Messbereichsendwert die Integer Messwerte auf 16-Bit.

#### Einstellbarer Tiefpassfilter

Der Tiefpassfilter wird durch die interne Signalverarbeitung im FPGA gerechnet. Die benötigte Filterfrequenz kann aus vordefinierten Werten des Parameters „Filter-Frequency“ gewählt werden. Tiefere Filterfrequenzen bewirken eine längere Verzögerung des Signals.

#### Oversampling Faktor

Für die Aufzeichnung und Analyse von Signalen bei denen die Netzwerkzyklen zu langsam sind kann ein Oversampling Faktor auf den 16-Bit Messwerten angewendet werden. Entsprechend dem gesetzten Faktor wird das Netzwerkintervall in bis zu 50 Subintervalle pro Kanal geteilt und zusätzliche Messwerte aufgezeichnet, denen durch die Steuerung ein genauer Zeitstempel zugeordnet werden kann. Aufgrund des massiven Einflusses auf das Prozessabbild wirkt der Oversampling Faktor nur auf den skalierten 16-Bit Integer Werten.

#### Zeitkonstante

Die zuschaltbare Zeitkonstante ist ein Hochpassfilter und wirkt direkt in der analogen Hardware. Dabei wird das Messsignal entsprechend einer Kondensator-Entladekurve gegen Null abgebaut. Die Zeitkonstante ist abhängig vom gewählten Messbereich und kann nicht verändert werden.

#### Spitzenwert Nachführung

Die Spitzenwerte werden über die zyklischen Ausgangsparameter „Peak control“ gesteuert und können auch während der laufenden Messung zurückgesetzt werden. Dies kann beispielsweise zur Überwachung von Prozessphasen während der aktiven Messung genutzt werden.

Die Spitzenwernerfassung kann wahlweise ungefiltert oder mit dem pro Kanal gewählten Filter erfolgen. So können Störsignale anwendungsspezifisch entweder erfasst oder weggefiltert werden.

#### Integralfunktion

Die Integralfunktion wird über den Parameter „Integral control“ gesteuert und kann auch während der aktiven Messung zugeschaltet werden.

Die interne Berechnung ist durch die hohe Datenrate präziser und entlastet das Kundensystem.

### Technische Daten

#### Ladungsverstärker

Anzahl Kanäle		1, 2, 3, 4
Messbereich pro Kanal (FSO)	pC max.	±20 ... 1 000 000
Messfehler bei FSO ≥100 pC	% FSO	<±0,5
Messfehler bei FSO <100 pC	% FSO	<±1,0
Drift		
25 °C, max. relative Feuchte rF of 60 %, nicht-kondensierend	pC/s	<±0,05
25 °C, max. relative Feuchte rF of 70 %, nicht-kondensierendtyp.	pC/s	<±0,05
50 °C, max. relative Feuchte rF of 50 %, nicht-kondensierend	pC/s	<±0,3
Reset-Operate-Sprung	pC	<±2
Eingangssignal ohne Beschädigung: Spannung (dauernd)	V	±10
Frequenzbereich		
Q <900 pC (Cg = 200 pC) (-3 dB, Kabelkap. <1 nF)	kHz	»0 ... <20
Q <31k pC (Cg = 7 nF) (-3 dB, Kabelkap. <1 nF)	kHz	»0 ... <10
Q <1M pC (Cg = 240 nF) (-3 dB, Kabelkap. <1 nF)	kHz	»0 ... <2
Operate-Reset Zeit		
Q <900 pC	us	<100
Q <31k pC	us	<300
Q <1M pC	ms	<1
Reset-Operate Zeit	us	<20
Bereichsumschaltzeit		
≤900 pC → >900 pC	ms	<60
≤31 kpC → >31 kpC	ms	<60
>900 pC → ≤900 pC	ms	<1
≤31 kpC → >31 kpC	ms	<1
Hardware Hochpassfilter, typ. (Zeitkonstante)		
Q <900 pC	s	0,03 (±6%)
Q <31k pC	s	1,05 (±6%)
Q <1M pC	s	36 (±7%)
Eingangsbezogenes Rauschen (typ. pp)		
Q <900 pC	pC	<0,6
Q <31k pC	pC	<21
Q <1M pC	pC	<720

### Datenerfassung

Auflösung (DeltaSigma)	bit	24
Samplerate	ksps	105
Gruppenlaufzeit Signalverarbeitung (zzgl. Gruppenlaufzeit Tiefpassfilter)	µs	160
TP-Filter 4.Ordnung (Typ Bessel) Grenzfrequenz (-3 dB)	Hz	1...20 000 in Stufen 1/2/5/10 pro Dekade Filter Aus = 20 000
Gruppenlaufzeit (Gesamtsystem) Tiefpass		
TP= Aus (20 000 Hz)	ms	<0
TP= 10 000 Hz	ms	<0,023
TP= 5 000 Hz	ms	<0,057
TP= 2 000 Hz	ms	<0,158
TP= 1 000 Hz	ms	<0,327
TP= 500 Hz	ms	<0,663
TP= 200 Hz	ms	<1,67
TP= 100 Hz	ms	<3,36
TP= 50 Hz	ms	<7,62
TP= 20 Hz	ms	<17,7
TP= 10 Hz	ms	<34,6
TP= 5 Hz	ms	<68,2
TP= 2 Hz	ms	<169
TP= 1 Hz	ms	<337

### Allgemeine Daten

Betriebstemperaturbereich	°C	-20 ... 65
Lagertemperatur	°C	-40 ... 85
Vibrationsfestigkeit IEC60068 Teil 2-6 (58 ...150 Hz konstant)	gp	1
Stossfestigkeit IEC60068 Teil 2-27 (11ms)	g	200
Schutzart nach EN60529 (nur bei montierten Kabeln und/oder abge- deckten Steckern)	IP	67
Gehäusematerial		Alu-Druckguss
Gewicht	g	420
Montageposition		beliebig

### Stromversorgung

Speisespannung	VDC	18 ... 30
Stromaufnahme bei 24 V	mA	<200
Ausgangsstrom für Kaskadierung bei 24V , bei dt +10 °C	A	<1,5
Ueberspannungsfestigkeit, 40 ms/max	V	55
Galvanische Trennung gegenüber Messkreis und digitalen Inputs typ. (nicht sicherheitsrelevant)	VDC	<50

### Industrial Ethernet Kommunikation

Hardware	Standard Ethernet IEEE 802.3 100 Base-Tx
	Transformatorgekoppelt

### Profinet IO

Slave gemäss den Standards der		PNO
Unterstützte Protokolle vorgesehen für		RT, IRT
Aktualisierungsrate minimal	µs	250

### EtherCAT

Slave gemäss den Standards der		ETG
Unterstützte Protokolle vorgesehen für		CoE, DC
Aktualisierungsrate, minimal	µs	100

### Ethernet/IP

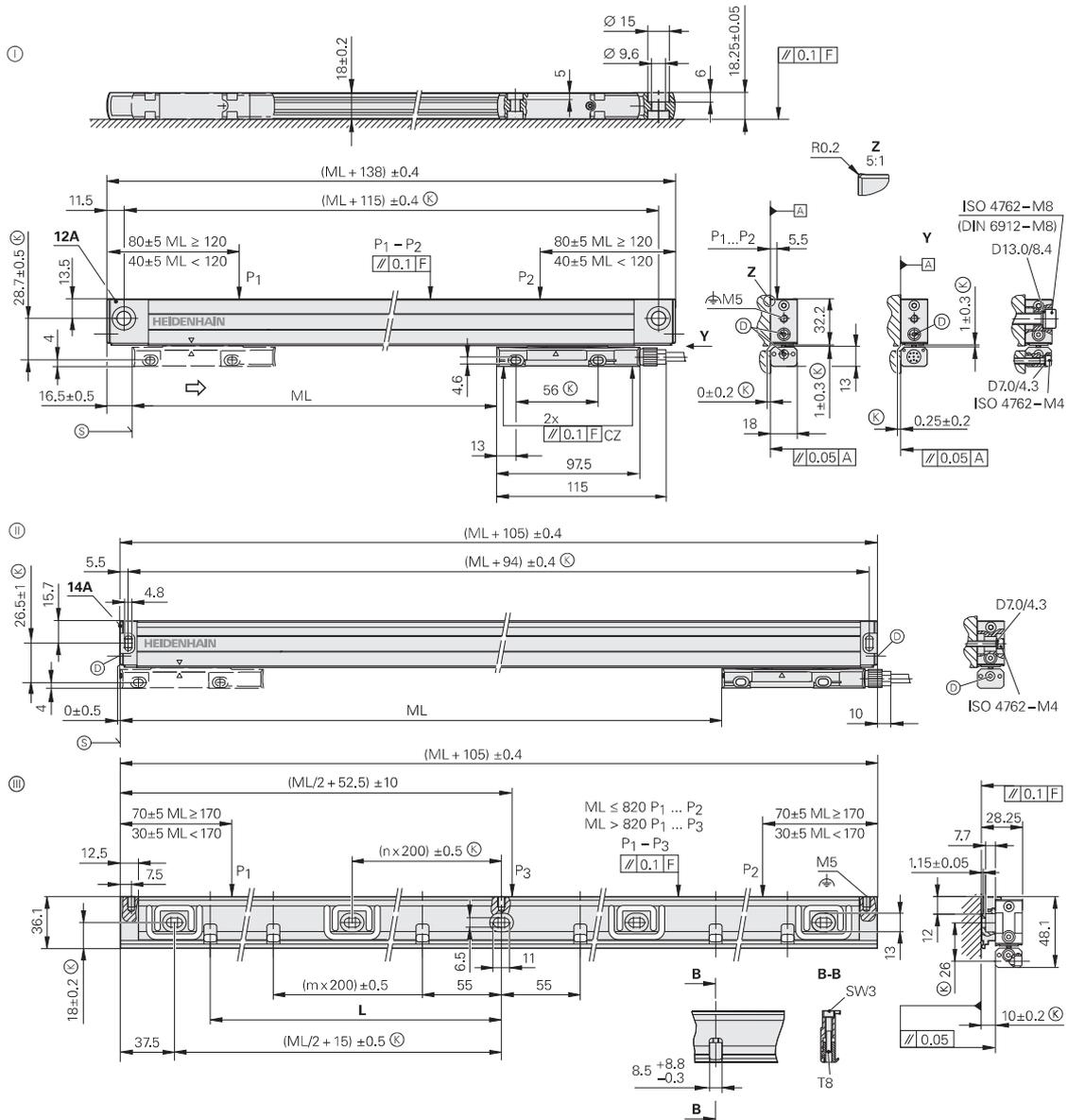
Slave gemäss den Standards der		ODVA
Unterstützte Protokolle vorgesehen für		CIP
Aktualisierungsrate, minimal	µs	1 000

5074A\_003-332d-12-18

# Baureihe LC 400

Absolutes Längenmessgerät mit kleinprofiligem Maßstabsgehäuse

- Für beengte Einbauverhältnisse



ML	70	120	170	220	270	320	370	420	470	520	570	620	670	720	770	820	920	1020	1140	1240	1340	1440	1540	1640	1740	1840	2040
L	37.5	55	75	100	115	140	175	200	225	250	275	300	325	350	375	400	450	500	555	610	655	710	760	810	855	910	1010

mm

Tolerancing ISO 8015  
ISO 2768 - m H  
< 6 mm: ±0.2 mm

- ⊙ = Endstück 12A; Anbau mit und ohne Montageschiene
- ⊙ = Endstück 14A; Anbau mit Montageschiene  
(Bei direkter Befestigung mit M4-Schrauben eingeschränkte technische Daten)
- ⊙ = Montageschiene MSL 41
- F = Maschinenführung
- P = Messpunkte zum Ausrichten
- ⊙ = Kundenseitige Anschlussmaße
- ⊙ = Druckluftanschluss
- ⊙ = Beginn der Messlänge ML (= 20 mm absolut)
- ⇨ = Bewegungsrichtung der Abtasteinheit für steigende Positionswerte



Technische Daten	LC 415 	LC 415	LC 485
<b>Maßverkörperung</b> Längenausdehnungskoeffizient	DIADUR-Glasmaßstab mit Absolutspur und Inkrementalspur, Teilungsperiode 20 µm $\alpha_{\text{therm}} \approx 8 \cdot 10^{-6} \text{ K}^{-1}$ (Montageart ☉/☉); mit <i>Montageschiene</i> : $\alpha_{\text{therm}} \approx 9 \cdot 10^{-6} \text{ K}^{-1}$ (Montageart ☉)		
<b>Genauigkeitsklasse*</b>	±3 µm, ±5 µm		
<b>Messlänge ML*</b> in mm	Montageschiene* oder Spannelemente* bis ML 1240 optional, ab ML 1340 notwendig 70 120 170 220 270 320 370 420 470 520 570 620 670 720 770 820 920 1020 1140 1240 1340 1440 1540 1640 1740 1840 2040		
<b>Funktionale Sicherheit</b> für Anwendungen bis	<ul style="list-style-type: none"> <li>SIL-2 nach EN 61508</li> <li>Kategorie 3, PL „d“ nach EN ISO 13849-1:2015</li> </ul>	–	
PFH (pro Achse)	$\leq 15 \cdot 10^{-9}$ (bis 6000 m über NN)	–	
Sichere Position <sup>1)</sup>	Gerät: ±550 µm (SM = 220 µm)	–	
	<i>mechanische Ankopplung</i> : Fehlerausschlüsse für das Lösen von Gehäuse und Abtasteinheit (Seite 21)		
<b>Schnittstelle</b>	EnDat 2.2		
Bestellbezeichnung	EnDat22		EnDat02
Messschritt <i>bei ±3 µm</i> <i>bei ±5 µm</i>	0,001 µm 0,010 µm		0,005 µm 0,010 µm
Taktfrequenz (Rechenzeit $t_{\text{cal}}$ )	$\leq 16 \text{ MHz}$ ( $\leq 5 \text{ µs}$ )		$\leq 2 \text{ MHz}$ ( $\leq 5 \text{ µs}$ )
<b>Inkrementalsignale</b>	–		$\sim 1 V_{\text{SS}}$ (20 µm)
Grenzfrequenz –3 dB	–		$\geq 150 \text{ kHz}$
<b>Elektrischer Anschluss</b>	separates Adapterkabel (1 m/3 m/6 m/9 m) am Montagefuß steckbar		
Kabellänge	$\leq 100 \text{ m}^{2)}$		$\leq 150 \text{ m}^{2)}$
Versorgungsspannung	DC 3,6 V bis 14 V		
Leistungsaufnahme (maximal)	3,6 V: $\leq 1,1 \text{ W}$ ; 14 V: $\leq 1,3 \text{ W}$		
<b>Verfahrgeschwindigkeit</b>	$\leq 180 \text{ m/min}$ (max. Beschleunigung in Messrichtung $\leq 100 \text{ m/s}^2$ )		
<b>Erforderliche Vorschubkraft</b>	$\leq 5 \text{ N}$		
<b>Vibration</b> 55 Hz bis 2000 Hz <i>bei Einwirkung auf</i>	<i>Abtasteinheit</i> : $\leq 200 \text{ m/s}^2$ (EN 60068-2-6) <i>Gehäuse ohne Montageschiene</i> : $\leq 100 \text{ m/s}^2$ (EN 60068-2-6) <i>Gehäuse mit Montageschiene, Kabelausgang rechts</i> : $\leq 150 \text{ m/s}^2$ , <i>links</i> : $\leq 100 \text{ m/s}^2$ (EN 60068-2-6) <i>Schock</i> 11 ms $\leq 300 \text{ m/s}^2$ (EN 60068-2-27)		
<b>Arbeitstemperatur</b>	0 °C bis 50 °C		
<b>Schutzart</b> EN 60529 <sup>3)</sup>	IP53 bei Anbau nach Anbauhinweisen im Prospekt; IP64 bei Sperrluft über DA 400		
<b>Masse</b>	Gerät: 0,2 kg + 0,55 kg/m Messlänge; <i>Montageschiene</i> : 0,9 kg/m		

\* Bei Bestellung bitte auswählen

<sup>1)</sup> Nach Positionswertvergleich können in der Folge-Elektronik weitere Toleranzen auftreten (Hersteller der Folge-Elektronik kontaktieren)

<sup>2)</sup> Mit HEIDENHAIN-Kabel; Taktfrequenz  $\leq 8 \text{ MHz}$

<sup>3)</sup> In der Anwendung muss das LC gegen Eindringen von Festkörpern und Flüssigkeit geschützt sein

**2.1.2 EL2521 - Technische Daten**

Technische Daten	EL2521-0000	EL2521-0024	EL2521-0124	EL2521-0025
Digitale Ausgänge	1 Kanal (2 Differenzausgänge A, B)	1 Kanal (2 potentialfreie FET-Ausgänge)	1 Kanal (2 FET-Ausgänge) 1 x 24 V-Ausgang	1 Kanal (2 potentialfreie FET-Ausgänge), negativ schaltend
Digitale Eingänge	2 (+T, +Z)		1 x Latch-Eingang (5..24 V <sub>DC</sub> )	2 (+T, +Z)
Signalspannung	RS422-Pegel	5..24 V <sub>DC</sub> , externe Speisung	24 V <sub>DC</sub> , externe Speisung	5..24 V <sub>DC</sub> , externe Speisung
Lastart	RS422 und Optokoppler, min. 120 Ω	beliebig		
Ausgangsstrom max. (je Kanal)	50 mA, RS422-Spezifikation	max. 1 A		
Schaltswelle "T" und "Z" Eingänge	High: >5 V (max. 24 V <sub>DC</sub> ), Low: <1 V			
Grundfrequenz	0 ... 500 kHz, Default: 50 kHz			
Auflösung	max. 15 Bit			
Schrittweite	10 mHz			
Spannungsversorgung für Elektronik	über den E-Bus			
Stromaufnahme aus dem E-Bus	typ. 280 mA (lastabhängig)			
Potenzialtrennung	500 V (E-Bus/Feldspannung)			
Bitbreite im Prozessabbild	def. 16 Bit Data in, 16 Bit Data out, 16 Bit Control, 16 Bit Status			
Konfiguration	über TwinCAT System Manager			
Gewicht	ca. 55 g			
Zulässiger Umgebungstemperaturbereich im Betrieb	0°C ... + 55°C			
Zulässiger Umgebungstemperaturbereich bei Lagerung	-25°C ... + 85°C			
Zulässige relative Luftfeuchtigkeit	95%, keine Betauung			
Abmessungen (B x H x T)	ca. 15 mm x 100 mm x 70 mm			
Montage <a href="#">[► 29]</a>	auf 35 mm Tragschiene nach EN 60715			
Vibrations- / Schockfestigkeit	gemäß EN 60068-2-6 / EN 60068-2-27, siehe auch <a href="#">Montagevorschriften [► 33]</a> für Klemmen mit erhöhter mechanischer Belastbarkeit			
EMV-Festigkeit / Aussendung	gemäß EN 61000-6-2 / EN 61000-6-4			
Schutzart	IP20			
Einbaulage	beliebig			
Zulassung	CE <a href="#">ATEX [► 49]</a> <a href="#">cULus [► 52]</a>		CE <a href="#">cULus [► 52]</a>	

# B Programm Code Dateien

1. Matlab Code Linearität Kistler Kraftsensoren (1 Seite)
2. Matlab Code Linearität Kraftsensor oben in der Fügeachse und Kraftsensor ABB (1 Seite)
3. Matlab Code Auswertung Fügeversuche (3 Seiten)
4. Schrittkette Beckhoff (siehe digitaler Anhang)
5. Hauptprogramm ABB Roboter (siehe digitaler Anhang)

---

```

1 %%
2 % Generate Mean-values for MSA Linearity of Force
3 addpath("G:\001_Projekt\006_intern\P03648 Robot guided joining - ABB
  \13 Presentation\Messreihen");
4 clc
5 clear
6 % Read in files
7 % Read in all FC.csv files in current folder
8     csvFileList = dir('*FC.csv');
9     numfiles = length(csvFileList);
10    csvFileData = cell(1, numfiles);
11 % Import data to csvFileData data struct
12 for k = 1:numfiles
13     csvFileData{k} = importdata(csvFileList(k).name);
14 end
15 % Create Arrays out of csv-file data struct and find minimum length
16 for i = 1:numfiles
17     eval(sprintf('T%d = csvFileData{1,%d};',i,i));
18     if i == 1
19         len = length (T1);
20         idx = 1;
21     end
22     if len > eval(sprintf('length(T%d)',i))
23         len = eval(sprintf('length(T%d)',i));
24         idx = i;
25     end
26 end
27 % calculate averaged force values of kistler, 20 values each force
  step
28 for i = 1:numfiles
29     % 1.Step 500N
30     Kupper(1,i) = eval(sprintf("mean(T%d(250:270,3));",i));
31     Klower(1,i) = eval(sprintf("mean(T%d(250:270,4));",i));
32     FCS(1,i) = eval(sprintf("mean(T%d(250:270,5));",i)); %Fx
33     FCS(2,i) = eval(sprintf("mean(T%d(250:270,6));",i)); %Fy
34     FCS(3,i) = eval(sprintf("mean(T%d(250:270,7));",i)); %Fz
35     % 2.Step 1250N
36     Kupper(2,i) = eval(sprintf("mean(T%d(340:360,3));",i));
37     Klower(2,i) = eval(sprintf("mean(T%d(340:360,4));",i));
38     % 3.Step 2000N
39     Kupper(3,i) = eval(sprintf("mean(T%d(410:430,3));",i));
40     Klower(3,i) = eval(sprintf("mean(T%d(410:430,4));",i));
41     % 4.Step 3000N
42     Kupper(4,i) = eval(sprintf("mean(T%d(490:510,3));",i));
43     Klower(4,i) = eval(sprintf("mean(T%d(490:510,4));",i));
44     % 5.Step 5000N
45     Kupper(5,i) = eval(sprintf("mean(T%d(560:580,3));",i));
46     Klower(5,i) = eval(sprintf("mean(T%d(560:580,4));",i));
47     FCS(4,i) = eval(sprintf("mean(T%d(560:580,5));",i)); %Fx
48     FCS(5,i) = eval(sprintf("mean(T%d(560:580,6));",i)); %Fy
49     FCS(6,i) = eval(sprintf("mean(T%d(560:580,7));",i)); %Fz
50 end

```

---

---

```

1 %%
2 % Generate Mean-values for MSA Linearity of Force, Kistler and Robot
3 addpath("G:\001_Projekt\006_intern\P03648 Robot guided joining - ABB
4 \13 Presentation\Messreihen");
5 clear
6 clc
7 % Read in files
8 for st = 1:5
9     switch st
10        case 1
11            cd 'G:\001_Projekt\006_intern\P03648 Robot guided
12 joining - ABB\13 Presentation\Messreihen\20210706_MSA_Linearität
13 Roboter\100%v1400';
14        case 2
15            cd 'G:\001_Projekt\006_intern\P03648 Robot guided
16 joining - ABB\13 Presentation\Messreihen\20210706_MSA_Linearität
17 Roboter\200%v1400';
18        case 3
19            cd 'G:\001_Projekt\006_intern\P03648 Robot guided
20 joining - ABB\13 Presentation\Messreihen\20210706_MSA_Linearität
21 Roboter\300%v1400';
22        case 4
23            cd 'G:\001_Projekt\006_intern\P03648 Robot guided
24 joining - ABB\13 Presentation\Messreihen\20210706_MSA_Linearität
25 Roboter\700%v1400';
26        case 5
27            cd 'G:\001_Projekt\006_intern\P03648 Robot guided
28 joining - ABB\13 Presentation\Messreihen\20210706_MSA_Linearität
29 Roboter\1400%v1400';
30    end
31
32    % Read in all FC.csv files in current folder
33    csvFileList = dir('*FC.csv');
34    numfiles = length(csvFileList);
35    csvFileData = cell(1, numfiles);
36
37    % Import data to csvFileData data struct
38    for k = 1:numfiles
39        csvFileData{k} = importdata(csvFileList(k).name);
40    end
41
42    % Create Arrays out of csv-file data struct and find minimum
43    length
44    for i = 1:numfiles
45        eval(sprintf('T%d = csvFileData{1,%d};',i,i));
46        if i == 1
47            len = length (T1);
48            idx = 1;
49        end
50        if len > eval(sprintf('length(T%d)',i))
51            len = eval(sprintf('length(T%d)',i));

```

---

---

```

1 function [] = AnalyseJoiningDataMA(Mode, heading)
2 % ANALYSEJOININGDATA Analysis of the measurement data out of csv
  files
3 %
4 % Mode: 1: Force Control, 2: Softmove
5 %
6 % heading: is to define the title after "robot is in <Mode> "...
7 addpath("G:\001_Projekt\006_intern\P03648 Robot guided joining - ABB
  \13 Presentation\Messreihen");
8 % Read in files according to Mode variable
9 switch Mode
10     case 1
11         % Read in all FC csv files in current folder
12         csvFileList = dir('*FC.csv');
13         numfiles = length(csvFileList);
14         csvFileData = cell(1, numfiles);
15
16         sProcess = 'FC';
17     case 2
18         % Read in all SM csv files in current folder
19         csvFileList = dir('*SM.csv');
20         numfiles = length(csvFileList);
21         csvFileData = cell(1, numfiles);
22
23         sProcess = 'SM';
24 end
25 % Generate heading for plots
26 sTitle = strcat("Roboter in ",sProcess,", ", heading);
27 % Import data to csvFileData data struct
28 for k = 1:numfiles
29     csvFileData{k} = importdata(csvFileList(k).name);
30 end
31 % Create Arrays out of csv-file data struct and find minimum length
32 for i = 1:numfiles
33     eval(sprintf('T%d = csvFileData{1,%d};',i,i));
34     if i == 1
35         len = length (T1);
36         idx = 1;
37     end
38     if len > eval(sprintf('length(T%d)',i))
39         len = eval(sprintf('length(T%d)',i));
40         idx = i;
41     end
42 end
43 % Check number of columns present, define length of
  ForceControlSensor
44 % vector
45 [~,cols] = size(T1);
46 ColsFCS = cols - 4;
47 % compensate delay (3 elements) of robot measurements due to
  calculation and sending through
48 % Profinet, delete last 3 rows

```

---

---

```

49 len = len - 3;
50 for i = 1:numfiles
51     for j = 5:1:cols
52         eval(sprintf('T%d(:,%d) = [T%d(4:end,%d); T%d(end,%d); T
%d(end,%d); T%d(end,%d)];',i,j,i,j,i,j,i,j,i,j));
53     end
54     Ttemp = eval(sprintf("T%d(1:(end-3),:);",i));
55     eval(sprintf("clear T%d;",i));
56     eval(sprintf("T%d = Ttemp;",i));
57     clear Ttemp;
58 end
59 Kup = zeros(len,1);           % upper Kistler force sensor
60 Klow = zeros(len,1);         % lower Kistler force sensor
61 FCS = zeros(len,ColsFCS);    %
    Fx,Fy,Fz,Tx,Ty,Tz,Px,Py,Pz,Ax,Ay,Az  FCS: ForceControlSensor
62 % Calculate averaged curves until the length of shortest vector
63 for i = 1:numfiles
64     Kup = Kup + eval(sprintf("T%d(1:len,3);",i));
65     Klow = Klow + eval(sprintf("T%d(1:len,4);",i));
66     % sum up values from T column 5 to ColsFCS,
    Fx,Fy,Fz,Tx,Ty,Tz,Px,Py,Pz,Ax,Ay,Az
67     for j = 1:ColsFCS
68         FCS(:,j) = FCS(:,j) + eval(sprintf("T%d(1:len,
%d);",i,4+j));
69     end
70 end
71 Kup = Kup/numfiles;
72 Klow = Klow/numfiles;
73 FCS = FCS/numfiles;
74 % Difference upper and lower Kistler
75 Kdiff = Kup - Klow;
76 % Plot different curves
77 C = [1 0 0; 0 0.5 0; 0 0 1; 1 0 1; 1 1 0; .5 .6 .7]; % Cell array
    of colors 'r','dg','b','m','y'
78 % Plot RAW curves
79 for i = 1:numfiles
80     eval(sprintf("plot(T%d(:,2),T%d(:,3),'k');",i,i));
81     hold on
82     eval(sprintf("plot(T%d(:,2),T%d(:,4),'g');",i,i));
83 end
84 grid on
85 title(sTitle,'Interpreter','Latex','FontSize',12)
86 xlabel("Position [mm]","Interpreter','Latex','FontSize',12)
87 ylabel('Kraft [N]','Interpreter','Latex','FontSize',12)
88 legend('oberer Kistler','unterer
    Kistler','Interpreter','Latex','FontSize',12)
89 % Save figure to pdf
90 PrintPdf(sProcess," raw curves")
91 % Plot averaged CURVE with the length measurements of the shortest
    vector
92 figure()
93 eval(sprintf("plot(T%d(:,2),Kup,'k')",idx));
94 hold on
95 eval(sprintf("plot(T%d(:,2),Klow,'g')",idx));

```

---

---

```

96 grid on
97 title(sTitle, 'Interpreter', 'Latex', 'FontSize', 12)
98 xlabel('Position [mm]', 'Interpreter', 'Latex', 'FontSize', 12)
99 ylabel('Kraft [N]', 'Interpreter', 'Latex', 'FontSize', 12)
100 legend('oberer Kistler', 'unterer
Kistler', 'Interpreter', 'Latex', 'FontSize', 12)
101 % Save figure to pdf
102 PrintPdf(sProcess, " averaged")
103 % Plot averaged robot FORCES and Kistler difference with the
length measurements of the shortest vector
104 figure()
105 eval(sprintf("plot(T%d(:,2),Kdiff,'k')",idx));
106 hold on
107 for i = 1:3
108     eval(sprintf("plot(T%d(:,2),FCS(:,
%d),'color',C(%d,:))",idx,i,i));
109 end
110 grid on
111 title(sTitle, 'Interpreter', 'Latex', 'FontSize', 12)
112 xlabel('Position [mm]', 'Interpreter', 'Latex', 'FontSize', 12)
113 ylabel('Kraft [N]', 'Interpreter', 'Latex', 'FontSize', 12)
114 legend('Kistler Differenz', 'Kraft Roboter
in X', 'Kraft Roboter in Y', 'Kraft Roboter in
Z', 'Location', 'northeast', 'Interpreter', 'Latex', 'FontSize', 12)
115 % Save figure to pdf
116 PrintPdf(sProcess, " forces")
117 % Plot TORQUES
118 figure()
119 eval(sprintf("plot(T%d(:,2),Kdiff,'k')",idx));
120 hold on
121 for i = 4:6
122     eval(sprintf("plot(T%d(:,2),FCS(:,
%d),'color',C(%d,:))",idx,i,i-3));
123 end
124 grid on
125 title(sTitle, 'Interpreter', 'Latex', 'FontSize', 12)
126 xlabel('Position [mm]', 'Interpreter', 'Latex', 'FontSize', 12)
127 ylabel('Kraft [N], Drehmoment
[Nm]', 'Interpreter', 'Latex', 'FontSize', 12)
128 legend('Kistler Kraft Differenz', 'Drehmoment Roboter
um X', 'Drehmoment Roboter um Y', 'Drehmoment Roboter um
Z', 'Location', 'northeast', 'Interpreter', 'Latex', 'FontSize', 12)
129 % Save figure to pdf
130 PrintPdf(sProcess, " torque")
131 end

```

*Published with MATLAB® R2018b*

## C MSA Linearität

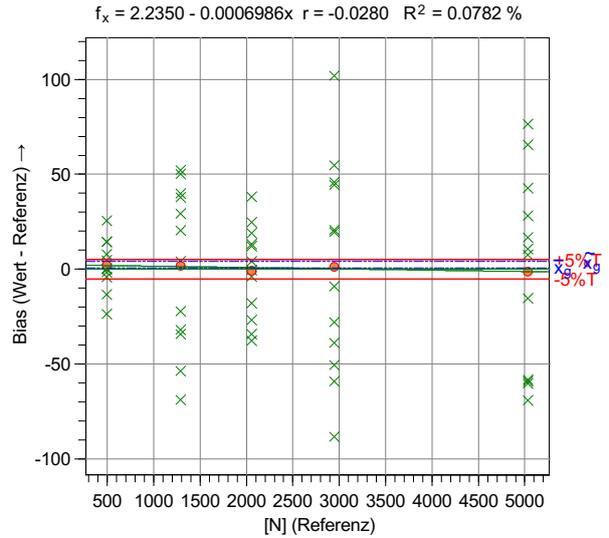
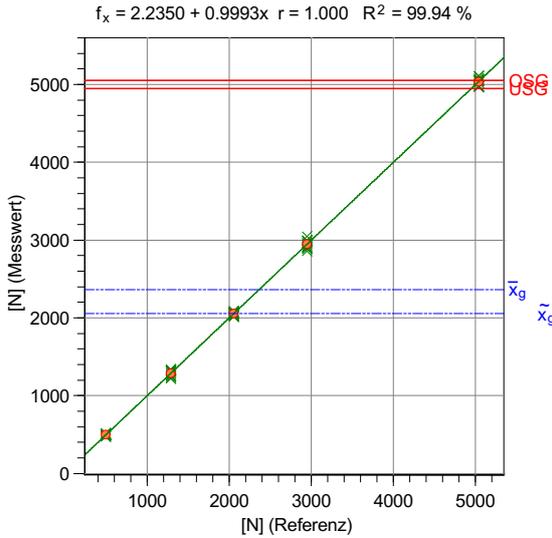
1. Linearitätsprüfung Kraftsensoren in der Fügeachse (1 Seite)
2. Linearitätsprüfung Kraftsensor oben in der Fügeachse und Kraftsensor ABB (1 Seite)

# Messsystemanalyse



Bericht Werk	TK - Presta Steering AG	Bearb.Name Prüfername	malgeo malgeo	Prüfbeg. Prüfende	31.05.2021 31.05.2021
Prfm.Nr. Prfm.Bez. Prüfort	SN.638331 Kistler 10kN M1661	Prfm.Aufl. Prüfgmd. Masch. Nr.	0.001 ibn 2365	Teilnr. Teilebez. Modul	261
Merkm.Nr. Merkm.Bez.	1	Nennm. Einheit	5000.000 N	OSG USG	5050.000 4950.000
Prfm. MSA Prfm. MSA Prfm. MSA	Kistler 20kN		SNr.: SNr.: SNr.:	SN.1195219	$\Delta$ n %
					50.000 -50.000

Bemerkung



n	$\bar{x}_g$ Ref.	$X_{A,1}$	$X_{A,2}$	$X_{A,3}$	$X_{A,4}$	$X_{A,5}$	$X_{A,6}$	$X_{A,7}$	$X_{A,8}$	$X_{A,9}$	$X_{A,10}$	$X_{A,11}$	$X_{A,12}$	$\bar{x}_{gj}$	$s_j$
1	495.6897	482.306	491.356	500.421	471.877	495.424	510.020	495.983	521.186	510.187	493.994	497.987	503.333	497.8397	13.0181
2	1288.197	1292.147	1308.667	1234.447	1265.937	1317.427	1253.987	1326.147	1256.097	1340.277	1219.267	1327.997	1338.247	1290.057	42.4533
3	2054.527	2016.717	2066.507	2058.647	2020.267	2027.597	2036.637	2050.567	2079.197	2067.657	2073.387	2053.937	2092.737	2053.657	24.1449
4	2945.257	2886.147	2894.547	2856.857	2906.457	2917.137	2989.667	2936.077	3047.227	2991.237	2999.917	2965.887	2964.817	2946.327	56.0236
5	5035.587	5042.967	5052.257	5063.597	5078.347	5111.917	4977.387	4976.327	5101.037	5020.117	5046.227	4966.327	4975.167	5034.307	51.1108

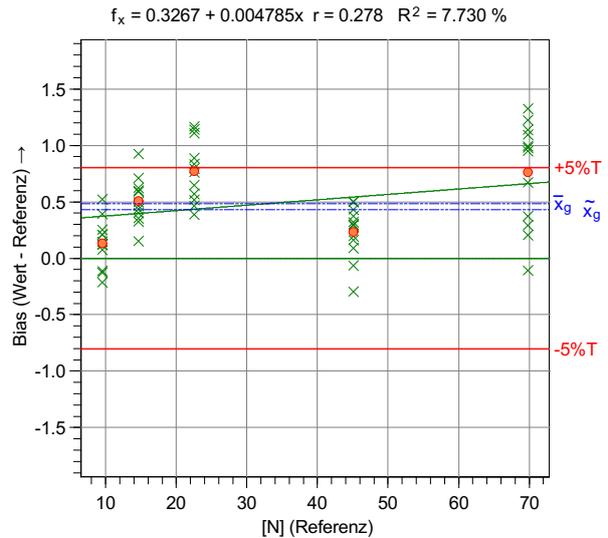
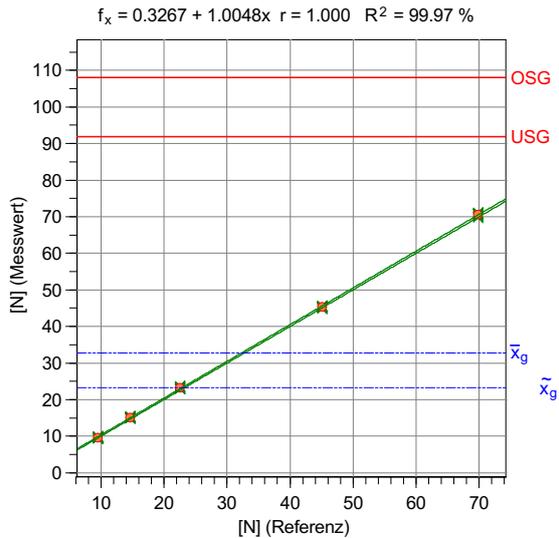
Teilnr.	1				Teilebez.				
Merkm.Nr.	1				Merkm.Bez.				
USG	=	4950.000	$T_m$	=	5000.000	OSG	=	5050.000	
			T	=	100.000				
			Bias	=	B <sub>i</sub>	=		2.14968	
			Bias	=	% B <sub>i</sub>	=		2.15%	
			Korrelationskoeffizient	=	Korrelationskoeffizient	=		0.078%	
			Li (GMPT EMS)	=	L <sub>i</sub>	=		0.00069857	
			Li (GMPT EMS)	=	% L <sub>i</sub>	=		0.00%	
			Prüfsystem fähig (Bi)						
			TK Presta Global 2018 v1: Linearity						

# Messsystemanalyse



Bericht Werk	TK - Presta Steering AG	Bearb.Name Prüfername	malgeo malgeo	Prüfbeg. Prüfende	07.06.2021 07.06.2021	
Prfm.Nr. Prfm.Bez. Prüfort	ABB Force Control Sensor M1661	Prfm.Aufl. Prüfgmd. Masch. Nr.	0.11 ibn 1661	Teilnr. Teilebez. Modul	1	
Merkm.Nr. Merkm.Bez.	1	Nennm. Einheit	100.000 N	OSG USG	108.000 92.000	$\hat{n} \geq 1$ 8.000 -8.000
Prfm. MSA Prfm. MSA Prfm. MSA	Kistler 10kN		SNr.: SNr.: SNr.:	SN.638331		

Bemerkung

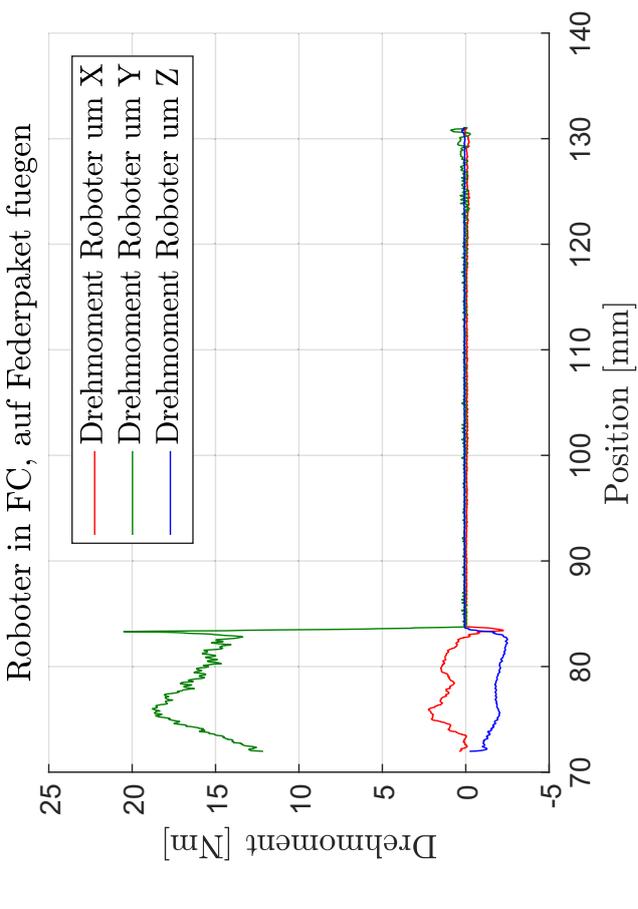
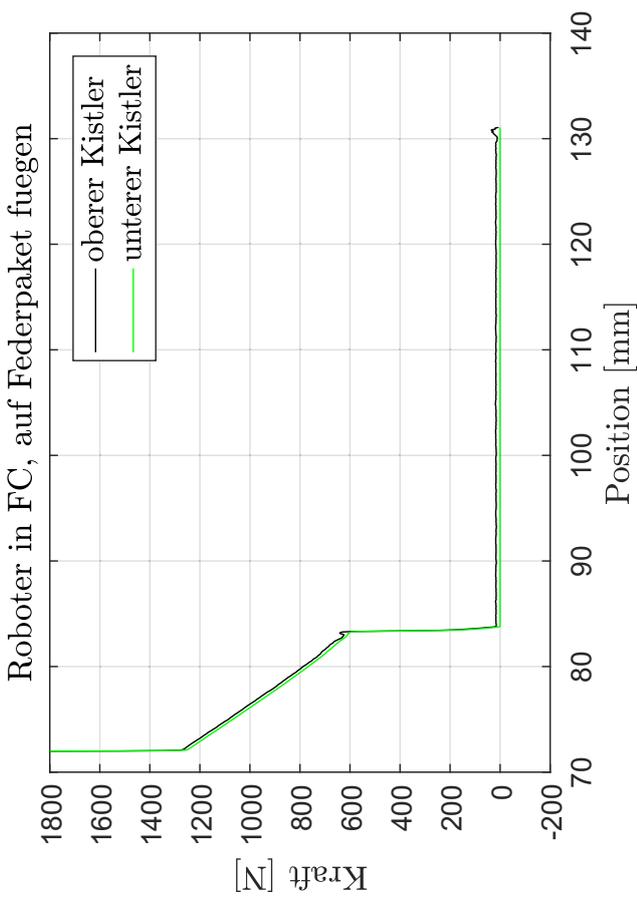
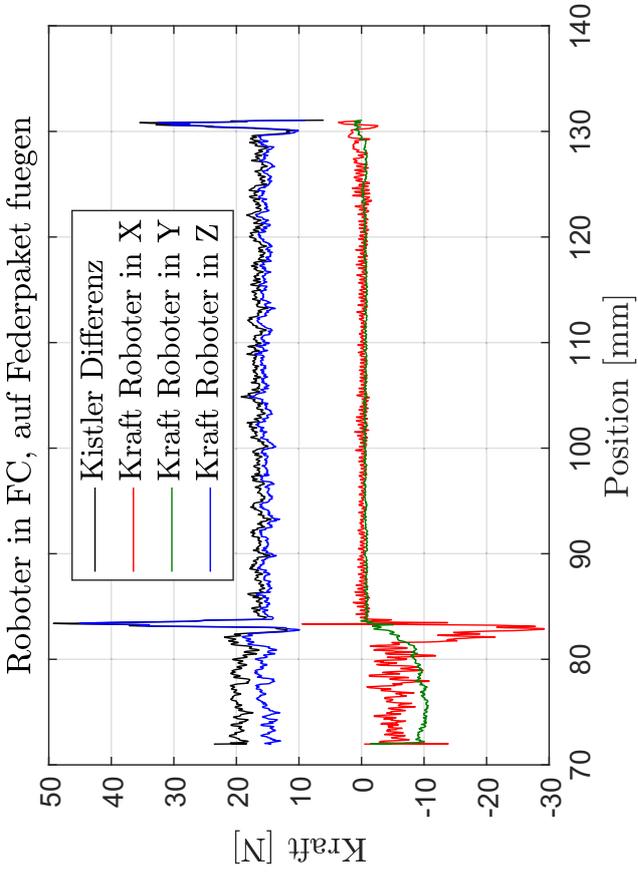
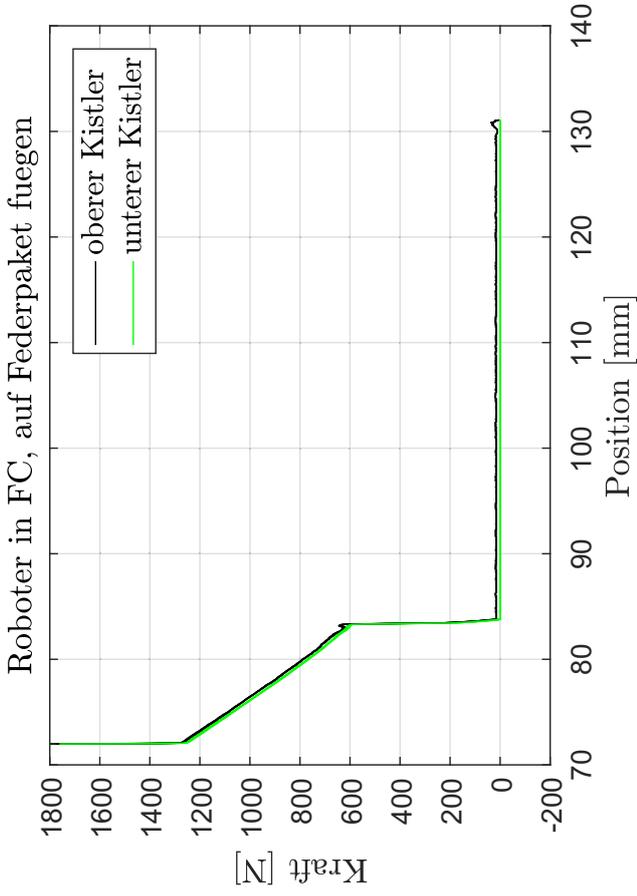


n	$\bar{x}_g$ Ref.	$x_{A,1}$	$x_{A,2}$	$x_{A,3}$	$x_{A,4}$	$x_{A,5}$	$x_{A,6}$	$x_{A,7}$	$x_{A,8}$	$x_{A,9}$	$x_{A,10}$	$x_{A,11}$	$x_{A,12}$	$\bar{x}_{gj}$	$s_j$
1	9.50798	9.584	9.899	10.029	9.630	9.760	9.717	9.716	9.653	9.624	9.382	9.296	9.394	9.6403	0.2120
2	14.61620	14.769	15.079	15.544	15.052	15.213	15.020	15.237	14.970	15.205	15.327	14.941	15.113	15.1225	0.2014
3	22.52186	23.040	23.285	23.408	23.168	23.091	22.909	23.665	23.633	22.987	23.689	23.356	23.320	23.2958	0.2679
4	45.09356	45.413	45.182	45.290	45.523	45.596	45.592	45.383	45.253	45.467	45.396	45.030	44.798	45.3269	0.2360
5	69.80446	70.175	70.942	71.029	71.130	70.755	70.781	69.698	70.474	70.007	70.796	70.096	70.897	70.5650	0.4638

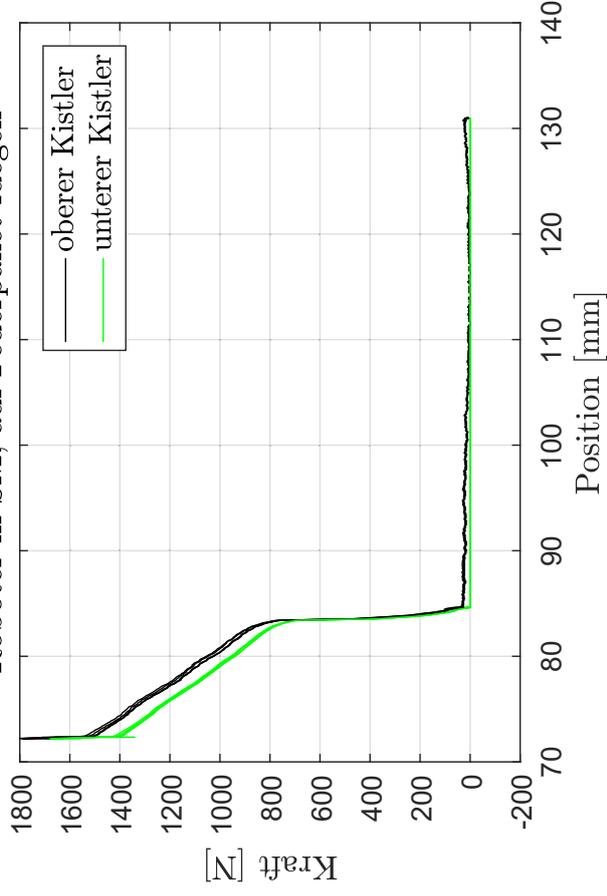
Teilnr.	1			Teilebez.				
Merkm.Nr.	1			Merkm.Bez.				
USG	=	92.000	$T_m$	=	100.000	OSG	=	108.000
			T	=	16.000			
			Bias	=	Bi	=		0.77392
			Bias	=	% Bi	=		4.84%
			Korrelationskoeffizient	=	Korrelationskoeffizient	=		7.730%
			Li (GMPT EMS)	=	Li	=		0.0047850
			Li (GMPT EMS)	=	% Li	=		0.03%
Prüfsystem fähig (Bi)								
TK Presta Global 2018 v1: Linearity								

# D Diagramme der Fügeversuche

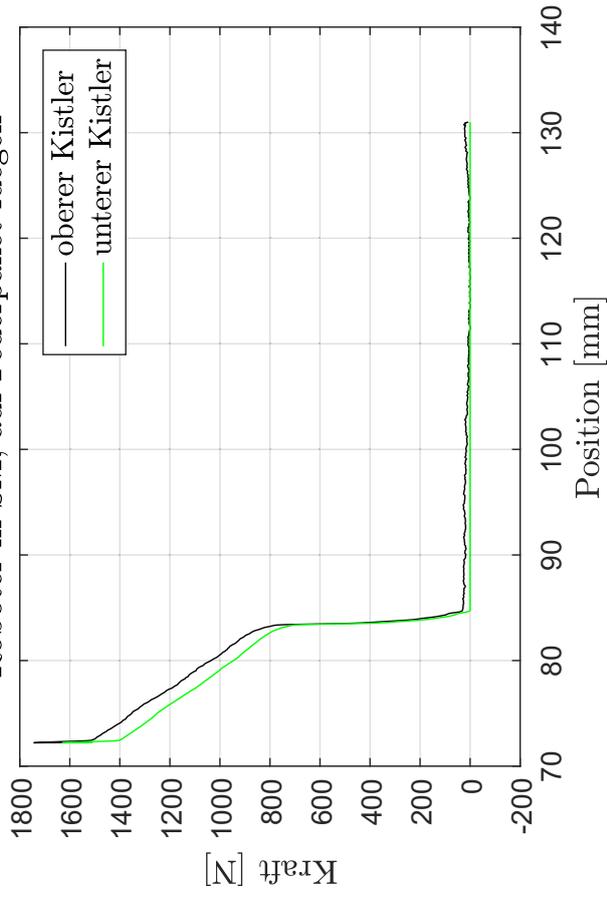
1. Force Control Federpaket (1 Seite)
2. Softmove Federpaket (1 Seite)
3. Force Control Lager einpressen (1 Seite)
4. Softmove Lager einpressen (1 Seite)
5. Force Control Getriebewelle durchschieben, Deckel fügen (1 Seite)
6. Softmove Getriebewelle durchschieben, Deckel fügen (1 Seite)



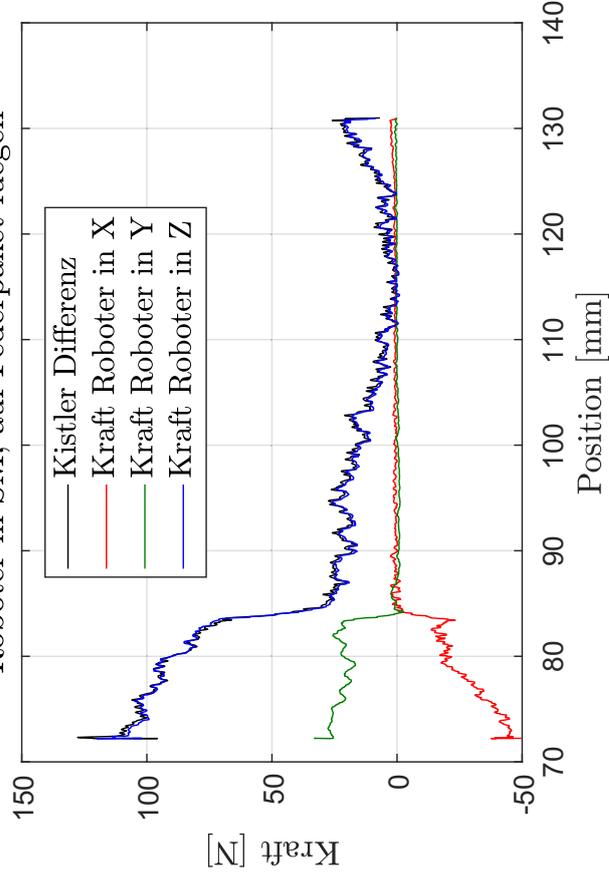
Roboter in SM, auf Federpaket fuegen



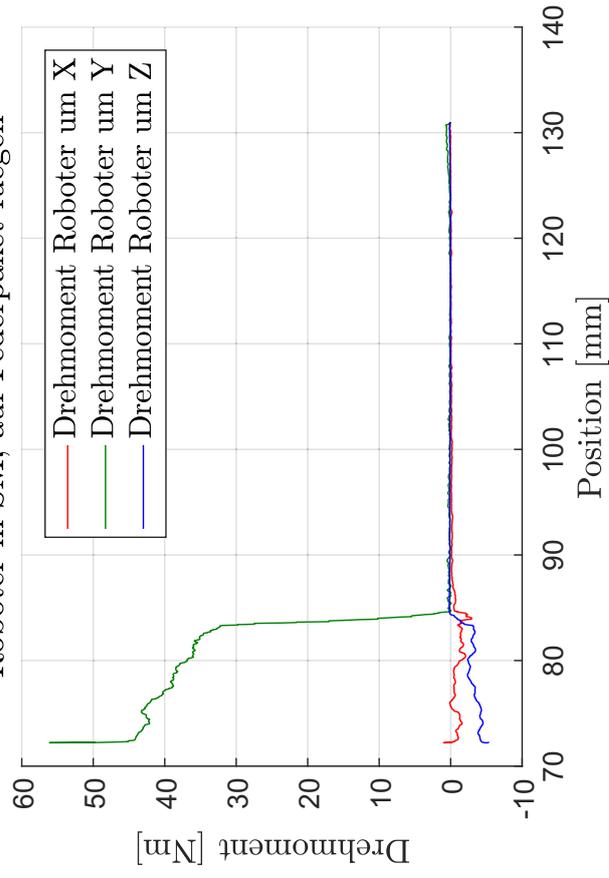
Roboter in SM, auf Federpaket fuegen

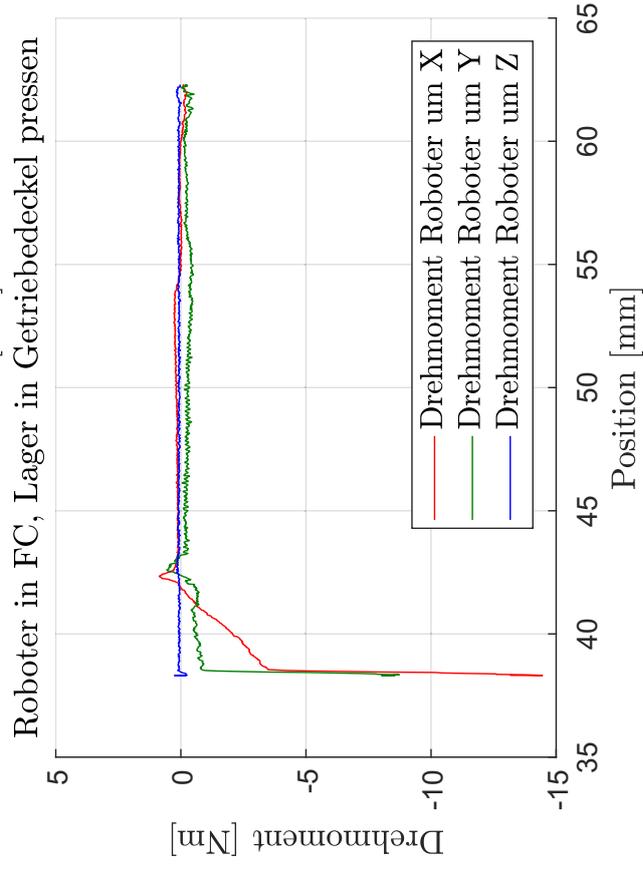
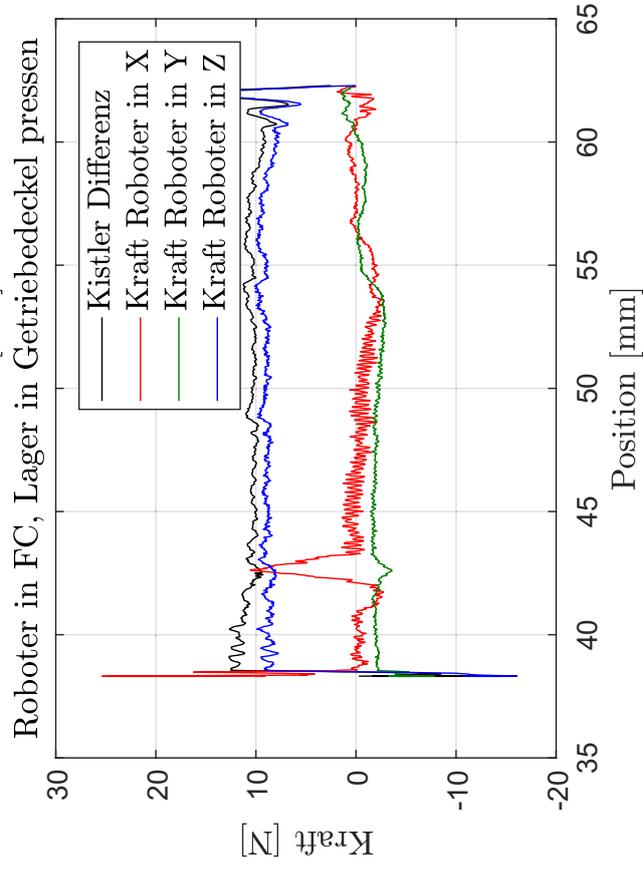
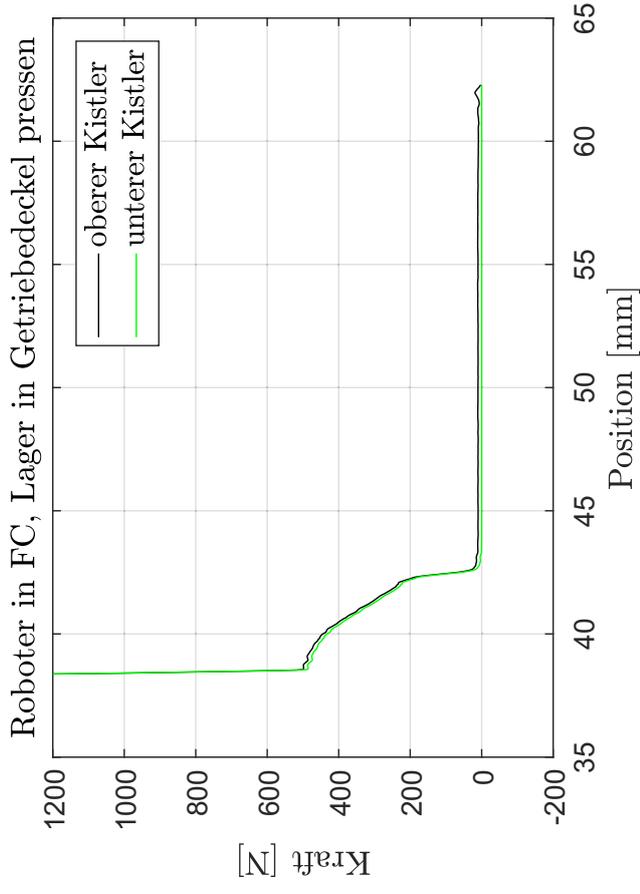
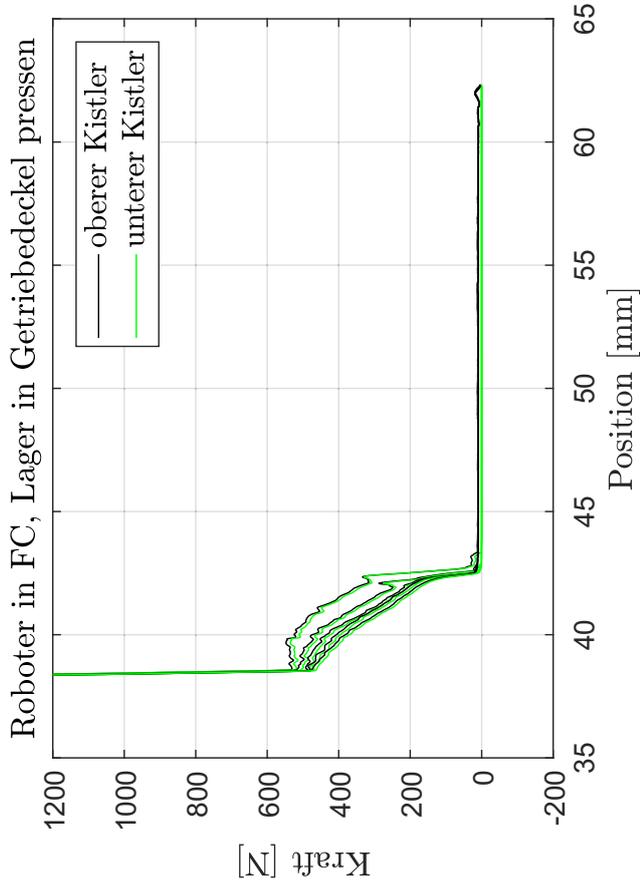


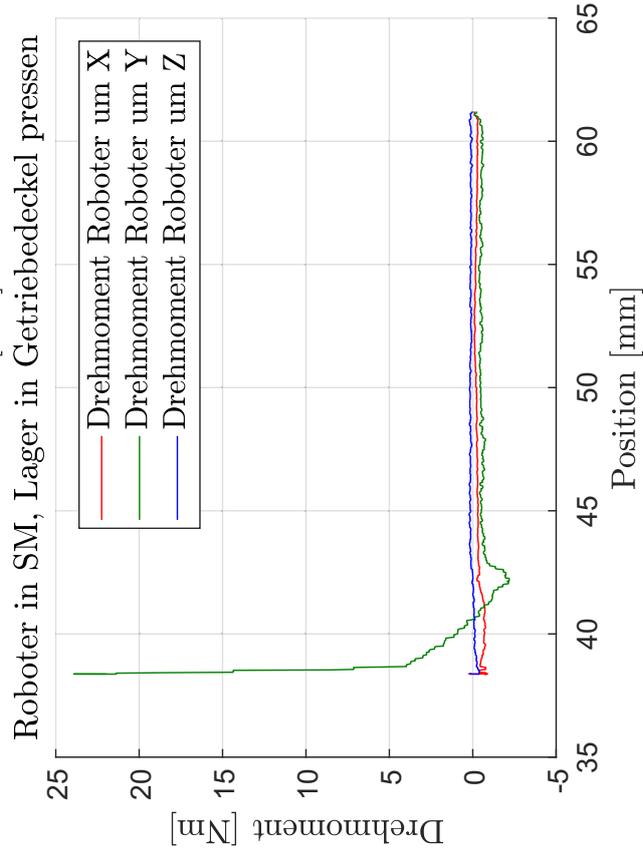
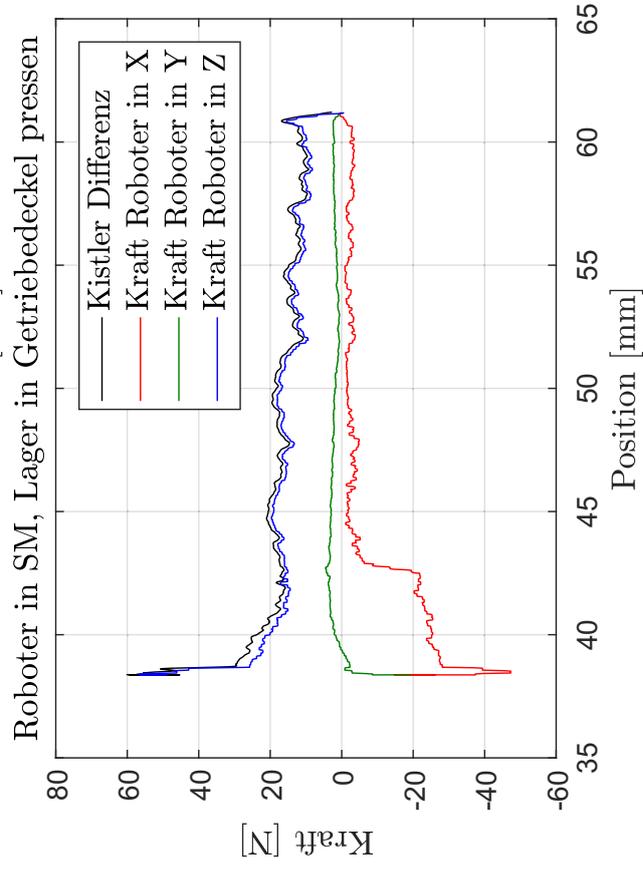
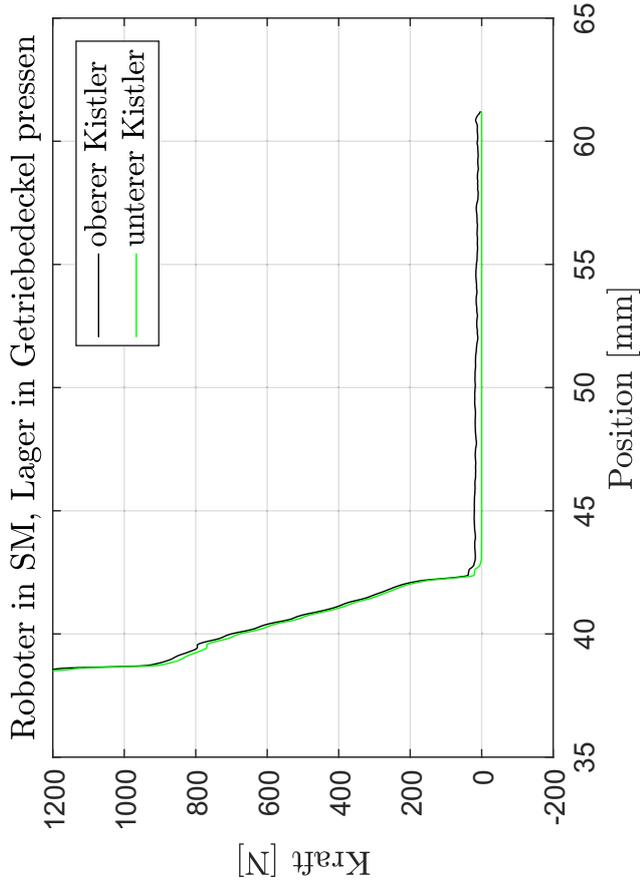
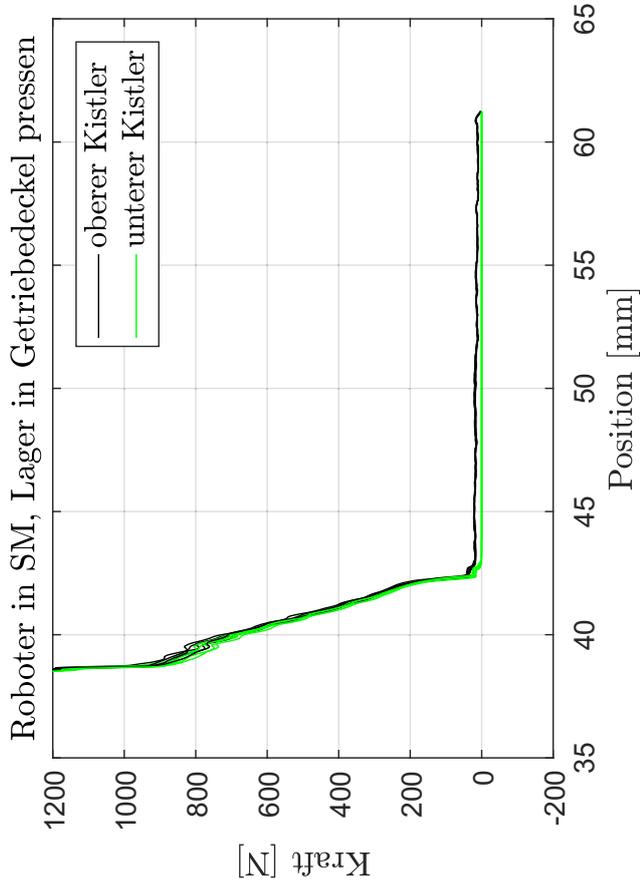
Roboter in SM, auf Federpaket fuegen



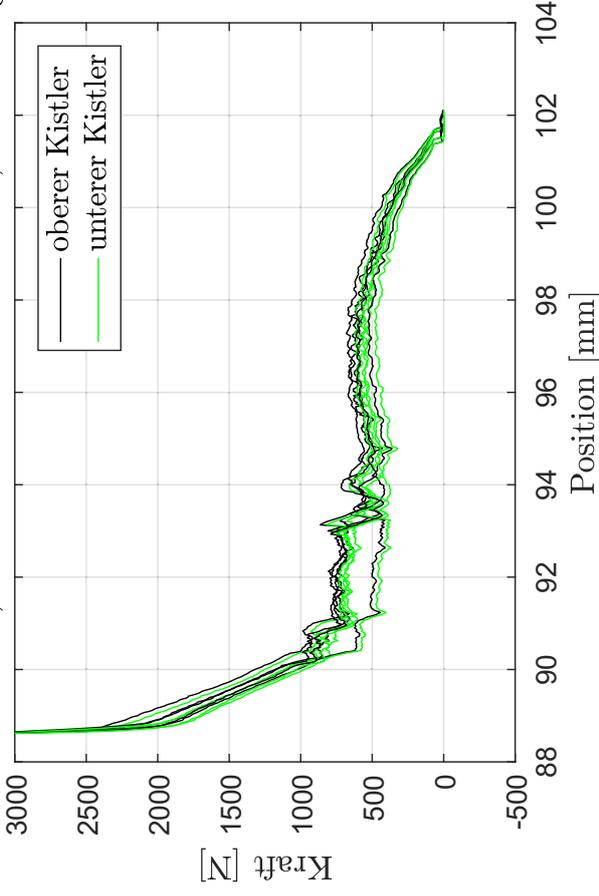
Roboter in SM, auf Federpaket fuegen



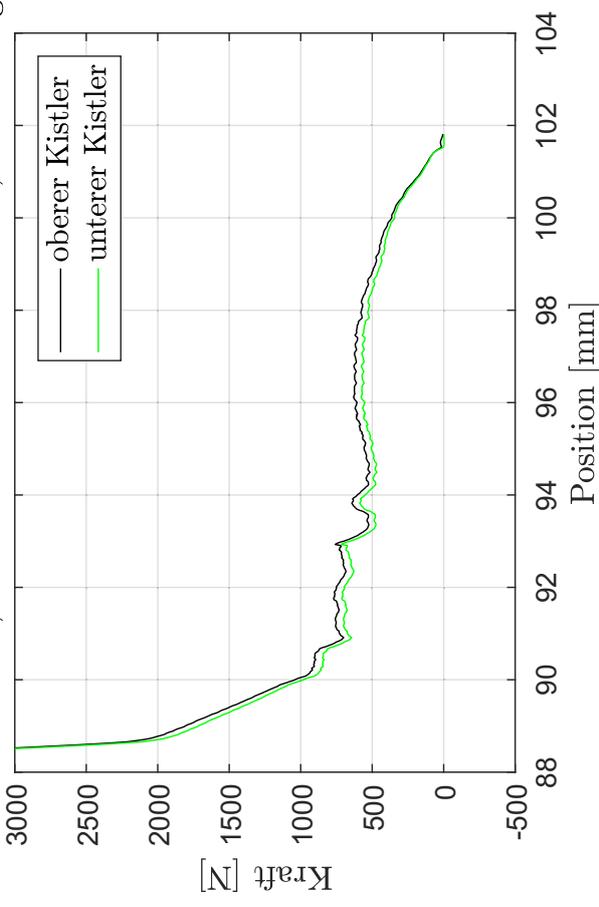




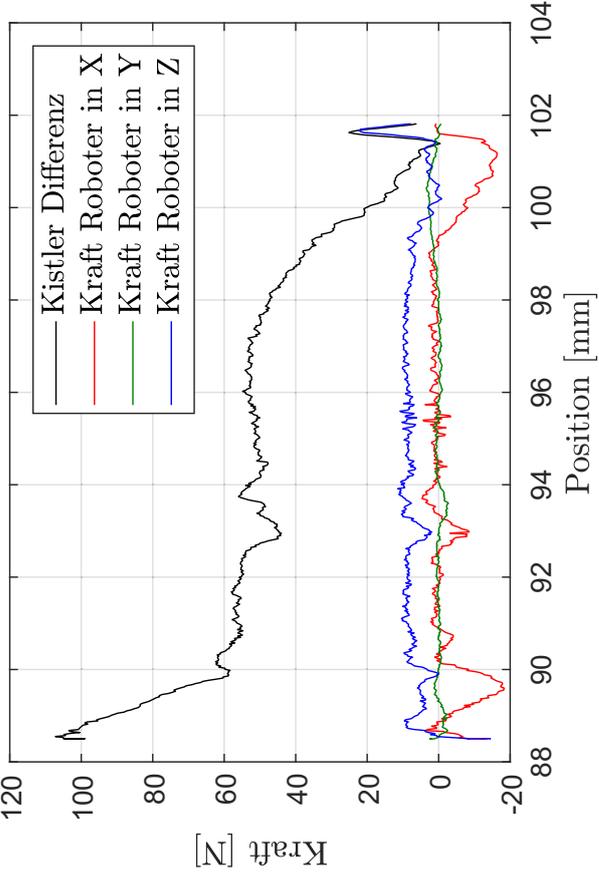
Roboter in FC, Getriebewelle durchschieben, Deckel fügen



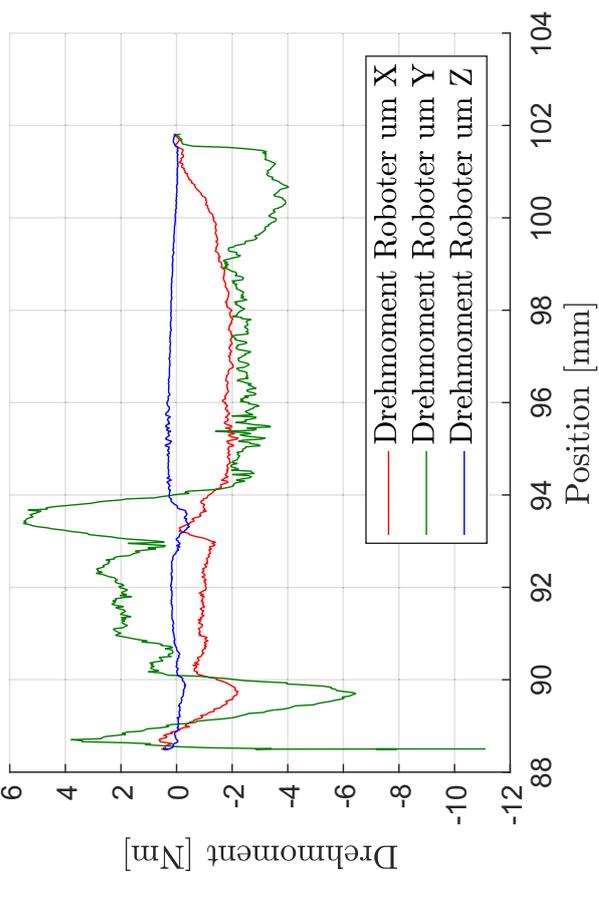
Roboter in FC, Getriebewelle durchschieben, Deckel fügen



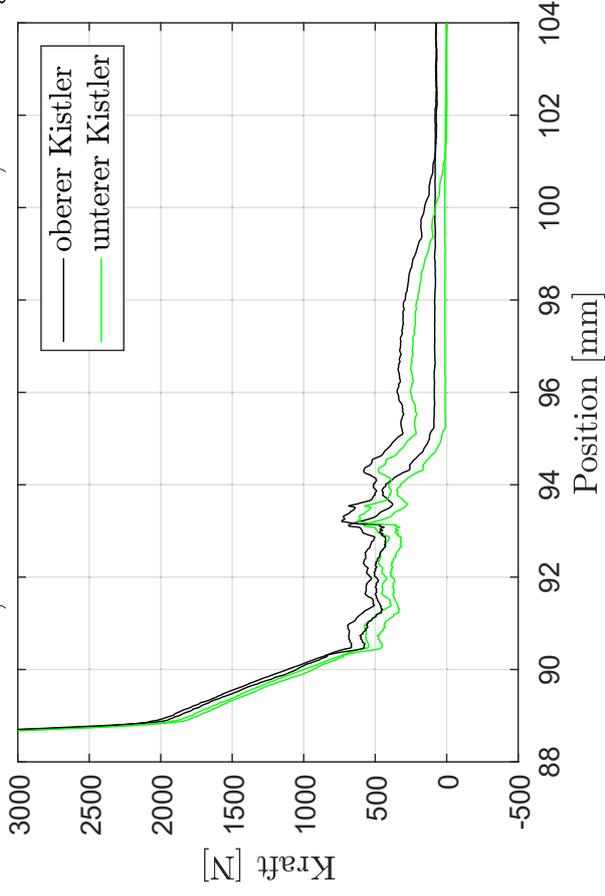
Roboter in FC, Getriebewelle durchschieben, Deckel fügen



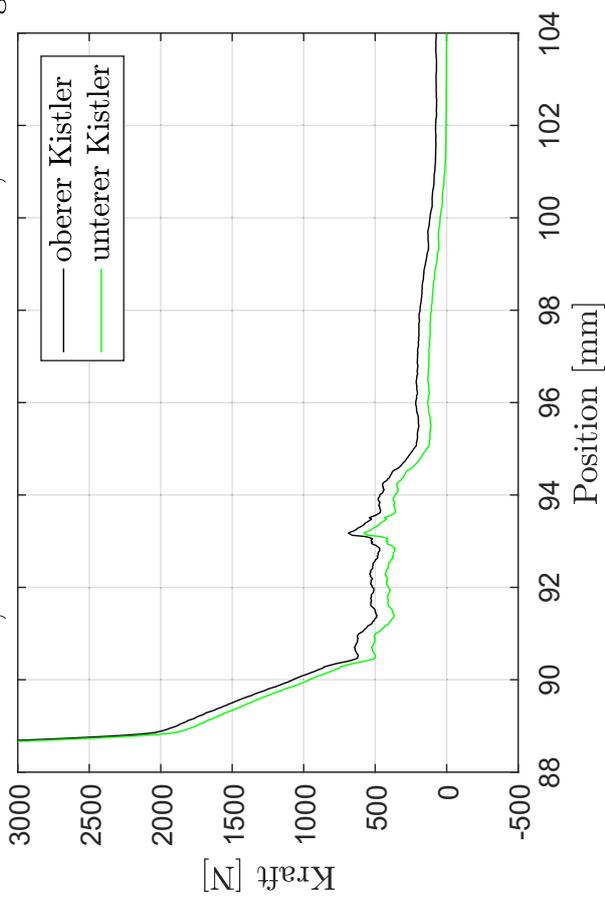
Roboter in FC, Getriebewelle durchschieben, Deckel fügen



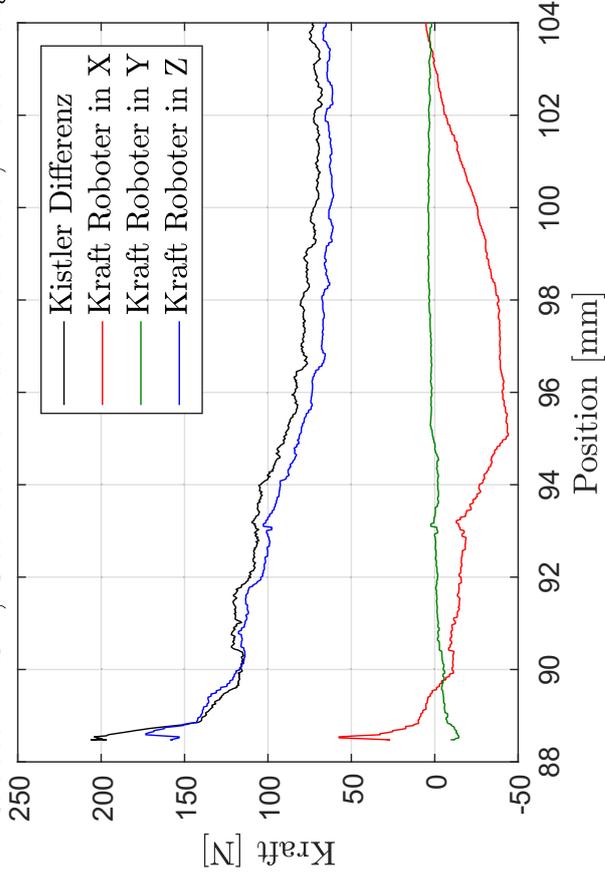
Roboter in SM, Getriebewelle durchgeschoben, Deckel fügen



Roboter in SM, Getriebewelle durchgeschoben, Deckel fügen



Roboter in SM, Getriebewelle durchgeschoben, Deckel fügen



Roboter in SM, Getriebewelle durchgeschoben, Deckel fügen

