

LEITFADEN – DSL4RAS

Systemarchitektur und modulares Design roboterartiger Systeme mittels multidimensionaler Kennfelder

Mai 2025

1	Allgemeines.....	2
2	Zielsetzung und Anwendungsbeispiel.....	3
3	Anwendung DSL4RAS	3
3.1	Ermittlung von Anforderungen	3
3.2	Erstellen des DSL4RAS-Modells	4
3.3	Erstellen der Simulationsmodelle.....	8
3.4	Berechnung der Lösungsräume.....	11
3.5	Anwendung der Ontologie.....	12
3.6	Bewertung des Konzepts	14
4	Möglichkeiten, Grenzen und Generalisierbarkeit.....	14
5	Zusammenfassung	15
6	References	15
7	Anhang	17
7.1	Symbole der DSL.....	17



LPL

1 Allgemeines

Dieser Leitfaden dient als modellbasierte Unterstützung bei der Komponentenauswahl und Auslegung von Robotern und roboterartigen Systemen (RAS). Er entstand während des DFG-Projekts „Systemarchitektur und modulares Design roboterartiger Systeme mittels multidimensionaler Kennfelder“ (DSL4RAS; DFG-Nr. 461993234). Zur gezielten Systemauslegung von roboterartigen Systemen mithilfe einer Beschreibungssprache (DSL) wird das Vorgehensmodell aus Abbildung 1 angewandt.

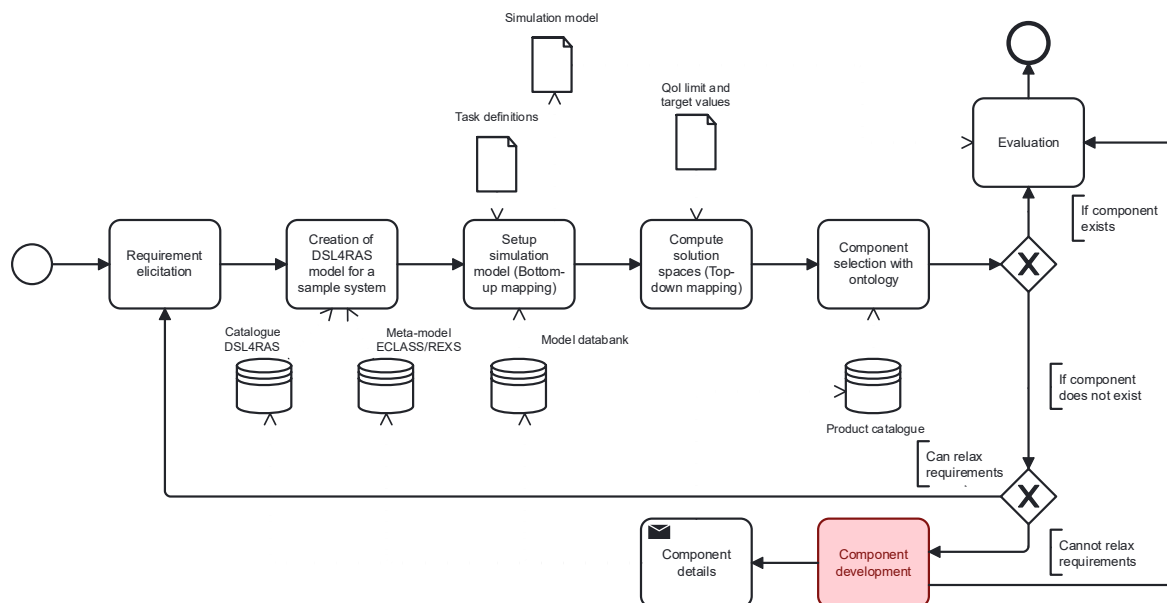


Abbildung 1: Vorgehensmodell DSL4RAS

Die Anwendung der DSL kann in insgesamt 6 Schritte unterteilt werden.

1. Anforderungen ermitteln
2. DSL4RAS-Modell erstellen
3. Simulationsmodell erstellen
4. Lösungsräume berechnen
5. Bauteile mithilfe der Ontologie auswählen
6. Bewertung des entstandenen Konzepts

Hierbei besteht DSL4RAS aus drei gekoppelten Konzepten:

- Modellierungssprache für den Anwendungsfall,
- Simulation durch zusammengesetzte Einzelmodelle,
- Komponentenauswahl durch Solution Space Engineering und Katalogabgleich.

Die einzelnen Schritte und der Durchlauf dieses Workflows werden in diesem Dokument beispielhaft erklärt.

2 Zielsetzung und Anwendungsbeispiel

Mit dem vorliegenden Leitfaden sollen Systementwickler bei der Entwicklung von roboterartigen Systemen unterstützt werden. Der Leitfaden soll zu Beginn der Systementwicklungsphase eingesetzt werden.

Als Beispiel soll ein SCARA (Selective Compliance Assembly Robot Arm) Roboter ausgelegt werden, wie schematisch in Abbildung 2 zu sehen ist. Im vorliegenden Beispiel besteht er aus 2 Armsegmenten, die jeweils von einer Einheit aus Motor, Getriebe, Sensor und Controller angetrieben werden. Zur Vereinfachung ist das linke Armsegment als unbeweglich angenommen und das rechte Armsegment beweglich. Beide Armsegmente deformieren nicht nennenswert und werden als starr angenommen. Der linke Arm bewegt sich auf einer Trajektorie von der Ausgangslage P_0 zu P_1 und zurück zu P_2 . An den Endpunkten P_1 und P_2 ist jeweils eine Haltezeit vorgesehen, in der der Roboter eine Aufgabe ausführen kann.

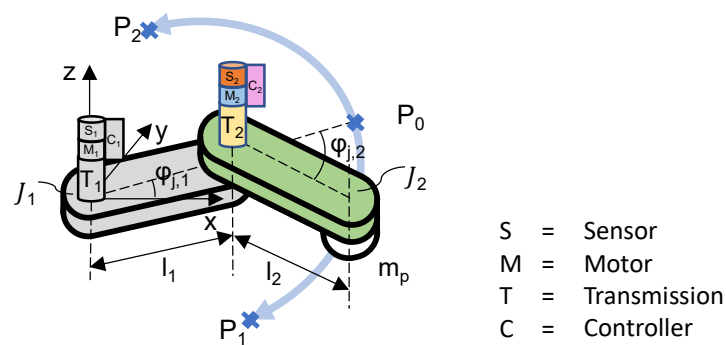


Abbildung 2: Elemente, Performance Measures und Trajektorie eines SCARA Roboters [1]

Zur Demonstrationszwecken wurde ein sehr herausforderndes Beispiel gewählt, welches einen SCARA an die Grenzen des machbaren bringt, bzw. diese überschreitet. Im Folgenden wird der Anwendungsfall und dessen Randbedingungen modelliert und genauer erklärt.

3 Anwendung DSL4RAS

In diesem Kapitel wird der beispielhafte Durchlauf des DSL4RAS-Workflows anhand des Anwendungsbeispiels beschrieben.

3.1 Ermittlung von Anforderungen

Bevor mit der Modellierung der DSL begonnen werden kann, müssen zuerst die Anforderungen an das roboterartige System angegeben werden. Typische Anforderungen an roboterartige Systeme sind beispielsweise die Anforderung an die Positionsgenauigkeit und die Zykluszeit (siehe Tabelle 1). Für die Anforderungen können die Wertebereiche vorgegeben werden. So kann z.B. ein SCARA mit einer kleineren Positionsgenauigkeit oder einer schnelleren Zykluszeit entwickelt werden.

Tabelle 1: Anforderungen an einen SCARA Roboter [2]

Robot	Name	Symbol	Lower	Upper	Units
Variant-1	Accuracy	Δx	0	4×10^{-4}	m
	Stab. time	t_{stab}	-	0.64	s
	Reach	L	0.55	-	m
	Robot mass	m_r	-	23	kg
Variant-2	Accuracy	Δx	0	1×10^{-3}	m
	Stab. time	t_{stab}	-	0.65	s
	Reach	L	0.90	-	m
	Robot mass	m_r	-	25	kg

3.2 Erstellen des DSL4RAS-Modells

Um eine gesamtheitliche Beschreibung eines RAS abbilden zu können muss eine Vielzahl an Disziplinen berücksichtigt werden. Hierunter zählen die Elektrotechnik, Softwareentwicklung, Mechanik und nicht zuletzt auch die Regelungstechnik. Zur Übersichtlichkeit wurden drei Sichten eingeführt, die zur Unterscheidung zwischen Elektronik und Software, Komponentenauslegung, und Mechanikauslegung dienen:

- **Steuerungs-Sicht [3]:** Elektronik, Software, Regelungstechnik, Signaltechnik,
- **Komponenten-Sicht:** Sensorik, Aktorik, Antriebe (Motoren), Getriebe,
- **Mechanik-Sicht [4]:** Struktur, Gelenke.

In Abbildung 3 ist der Anwendungsfall mittels der entwickelten DSL abgebildet. Die DSL besteht aus grafischen Symbolen (im Anhang) und einem hinterlegten Metamodel, das zur Beschreibung der Symbole und deren Zusammenhänge dient. Das Metamodel bietet ebenfalls die Schnittstelle zu Simulationen die eine Analyse zur Komponentenauswahl bieten. Die Grundsymbole der DSL sind in Tabelle 2 zu finden, alle weiteren Symbole sind im Anhang beschrieben.

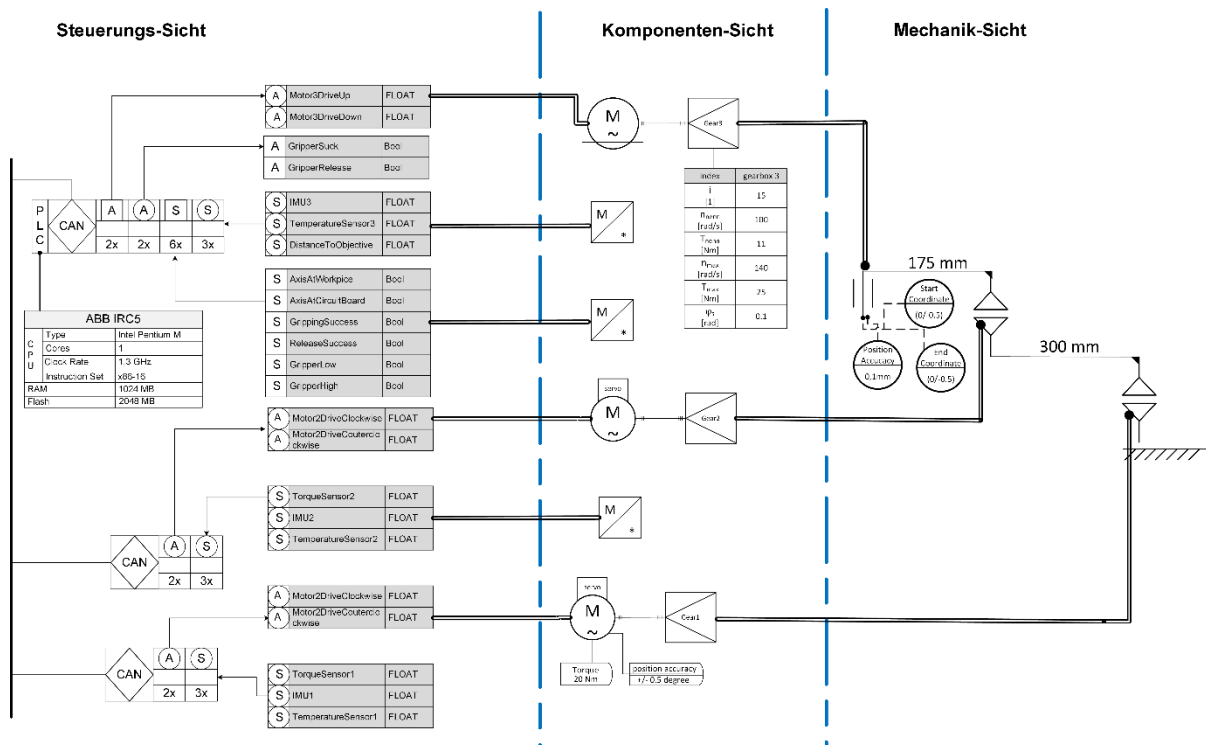


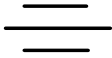
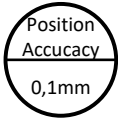



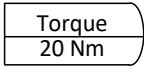
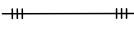
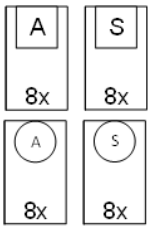




Abbildung 3: DSL4RAS Modell des Anwendungsfalls

Beginnend mit der rechten Seite von Abbildung 3, ist der mechanische Aufbau des SCARA Roboters zu sehen, bestehend aus 2 Drehgelenken und einem Translationsgelenk. Diese sind mittels nicht weiter detaillierten Verbindungselementen verbunden. In unserem Beispiel sind am Tool Center Point (TCP) Anwendungsanforderungen spezifiziert. Die Positionsgenauigkeit ist entsprechend des Beispiels mit der Toleranz aus Tabelle 1 angegeben. Die Trajektorie wird über den Start und Endpunkt des TCP angegeben und ist hier aufgrund des SCARA Anwendungsfalls auf 2 Dimensionen beschränkt. Allgemein ist die Anzahl der Dimensionen variabel.

Die Komponenten aus der Komponentensicht sind mittels einer Kante verbunden, die an die Position der jeweiligen Komponente zeigt. Ziel ist es aktuell vor allem die Motor-Getriebe-Kombinationen, also den Antriebsstrang, genauer zu beschreiben. Die Komponentensicht bietet die Möglichkeit eventuell vorausgesetzte Komponenten mit ihren Eigenschaften zu beschreiben. Dies ist beispielsweise der Fall bei „Gear 3“ bei dem die Parameter in der tabellenartigen Index Darstellung festgelegt wurden. Ebenfalls bei einem der verwendeten Motoren ist die Eigenschaft „Torque“ und „Positionsgenauigkeit“ angegeben. Insgesamt werden alle Komponenten aufgeführt, die dazu beitragen, Genauigkeiten, Prozesszeiten und Latenzzeiten des RAS zu beeinflussen.

Tabelle 2: Grundsymbole




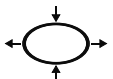
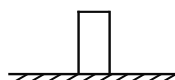
Sicht	Symbol	Spezifikation
Mechanik-Sicht		Verbindungselement. Es ist ein roboterbezogenes Element und es kann sich z.B. um einen Arm, eine Schwinge oder ein Karussell handeln.
Mechanik-Sicht		Drehgelenk mit fluchtender Rotationsachse. Drehbewegung der Rotationsachse erfolgt um sich selbst.
Mechanik-Sicht		Translationsgelenk (Schlitten). Drehung um die Translationsachse ist möglich.
Mechanik-Sicht		Anforderungen an den verbundenen Punkt im mechanischen System. Diese können Zeitanforderungen, Positionsgenauigkeiten, oder Trajektorien sein.
Komponenten-Sicht		Getriebe. Kann expandiert werden in der genauen Beschreibung des Getriebes wie im Folgenden. Die spitze Seite des Dreiecks zeigt in Richtung a Seite, die breite in Richtung b Seite.
Komponenten-Sicht		Servo Motor. Kann mittels Komponentenschnittstelle an die Mechanik und Steuerung verbunden werden. Enthält Eigenschaften wie Nennmoment, Schnittstellenbeschreibung und mehr.
Komponenten-Sicht		Wechselstrom Motor. Kann mittels Komponentenschnittstelle an die Mechanik und Steuerung verbunden werden. Enthält Eigenschaften wie Nennmoment, Schnittstellenbeschreibung und mehr.
Komponenten-Sicht		Eigenschaftsbeschreibung für Komponenten. Visualisiert die Eigenschaften, die einer Komponente zugeordnet sind.
Komponenten-Sicht		Komponenten Schnittstelle. Verbindet die Komponenten der Einzelnen Sichten.
Steuerungs-Sicht		Ein- und Ausgangsklemmen (E/A). Der Signaltyp wird durch die Form und den Buchstaben oben angegeben <ul style="list-style-type: none"> • Digitaler Aktoranschluss (Quadrat A) • Digitaler Sensoranschluss (Quadrat S) • Analoger Aktoranschluss (Kreis A) • Analoger Sensoranschluss (Kreis S) Die Zahl auf der Unterseite gibt die Anzahl der verfügbaren E/A-Kanäle an.
Steuerungs-Sicht		Die speicherprogrammierbare Steuerung (SPS) führt eine Steuerungsaufgabe aus.
Steuerungs-Sicht		Buskoppler mit Ethernet Interface. Die Interfaces werden über den Namen in der Mitte des Symbols bestimmt.

Die Komponenten sind ebenfalls mittels Verbinder mit den Signalen der Steuerungssicht verbunden. In der Steuerungssicht wird die Netzwerkarchitektur und Informationen zur Softwareverteilung angegeben. Alle drei Sichten kombiniert können so die auftretenden Latenzen im Gesamtsystem betrachten, welche maßgeblich für die Funktionalität des RAS notwendig sind.

Die Darstellung mittels der DSL dient zur Parametrierung und Konfiguration für die weiteren Analysen und Berechnungen die notwendig sind, um geeignete Komponenten für den entsprechenden Anwendungsfall auszuwählen. Die Verbindung der DSL zu den entsprechenden Simulationsmodellen ist über das hinterlegte Metamodell gewährleistet, dass die Zusammenhänge zwischen den Symbolen und die hinterlegten Informationen hinter den Symbolen definiert.

Zur genaueren Beschreibung der Getriebestruktur wurde ein systematisches Vorgehen durch Landler et al. [5] entwickelt. Dazu werden im ersten Schritt unterschiedliche Symbole für die einzelnen Kontaktarten definiert:

Tabelle 3: Modelle zur Getriebemodellierung

Kontaktart	Symbol	Spezifikation
Rollender Kontakt		Rollende Bewegung (z.B. bei einem Rillenkugellager)
Wälzkontakt		Kontakt von zwei Zahnflanken (Kombination aus Rollen und Gleiten)
Gleitender Kontakt		Reines Gleiten zwischen zwei Kontaktpartner
Flexibler Kontakt		Große Verformung von zwei Kontaktpartner
Fester Kontakt		Keine Relativbewegung zwischen zwei Kontaktpartnern

Der Getriebeaufbau kann anschließend durch die Kombination der einzelnen Elemente beschrieben werden. Als beispielhaftes Getriebe soll hier ein Planetengetriebe genauer analysiert werden, welches über das Sonnenrad (*Sun gear*) angetrieben wird und den Planetenträger (*Carrier*) als Abtrieb nutzt. Das Hohlrad (*Ring gear*) ist in diesem Fall fest mit dem Gehäuse verbunden.

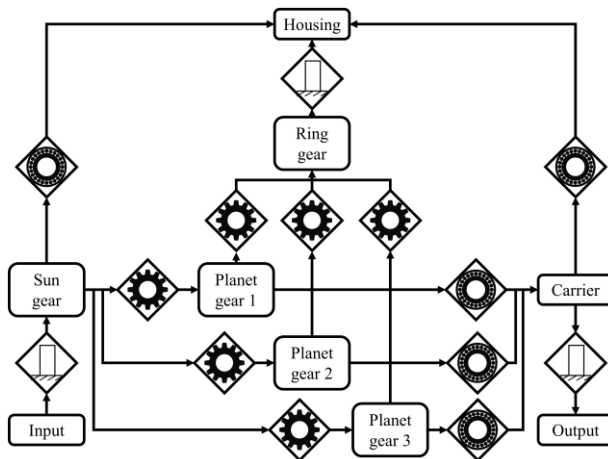


Abbildung 4: Beispiel des Vorgehens anhand eines Planetengetriebes

Das gezeigte Modell weist die Kontaktarten „Fester Kontakt“ (z.B. Verbindung Planetenträger mit dem Gehäuse), „Wälzkontakt“ (z.B. Zahneingriff Sonne mit Planetenrad (Planet gear 1)) und „Rollender Kontakt“ (z.B. Lagerung zwischen Planetenrad und Planetenträger) auf.

3.3 Erstellen der Simulationsmodelle

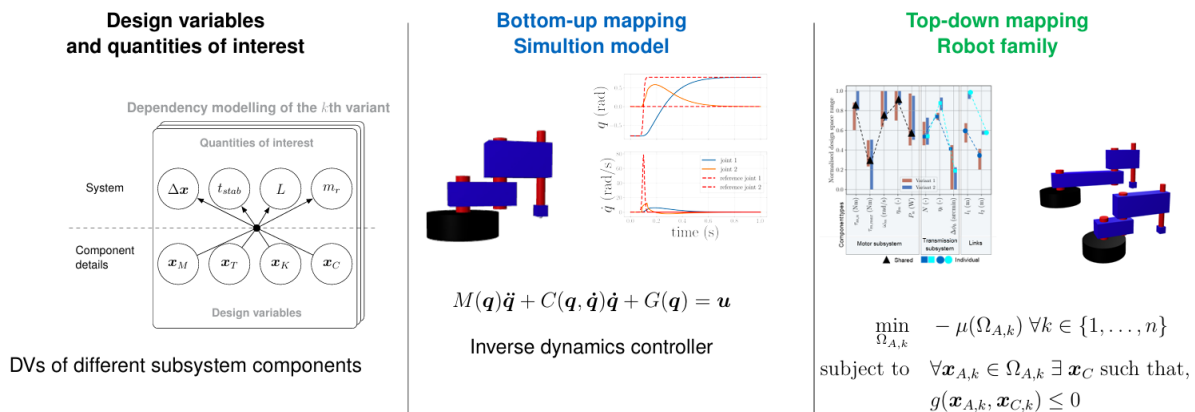


Abbildung 5: Überblick über das Vorgehen

In den vorhandenen Modellen sind sowohl die Eigenschaften in Eigenschaftsmodellen als auch das zeitliche Verhalten in Verhaltensmodellen der einzelnen Komponenten und des Gesamtsystems abgebildet. Die Eigenschaften für das Getriebe sind beispielsweise das Übersetzungsverhältnis i und der Wirkungsgrad $\eta(T_M, \dot{\phi}_M)$ (siehe Abbildung 6). Sie beeinflussen die Eigenschaften des Gesamtsystems wie die Zykluszeit t_c und die Positionsgenauigkeit AP_P .

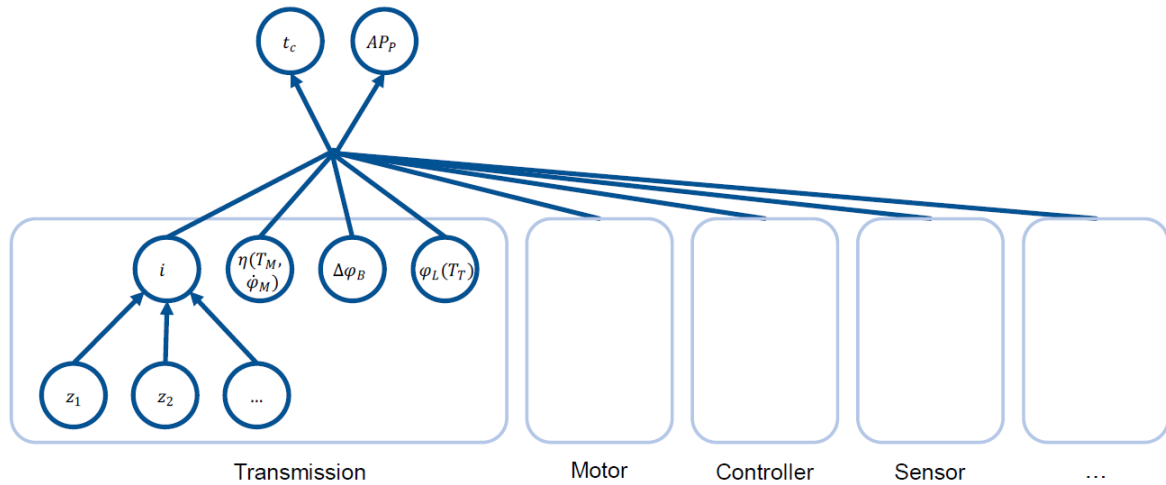


Abbildung 6: Eigenschaften des Getriebes dargestellt in einem Attribute Dependency Graph [1]

Die Eigenschaftensmodelle werden in den Komponentenverhaltensmodellen aufgerufen. Die Komponentenverhaltensmodelle, wie das Modell in Abbildung 7, bilden die zeitabhängigen Größen an den Schnittstellen aufeinander ab.

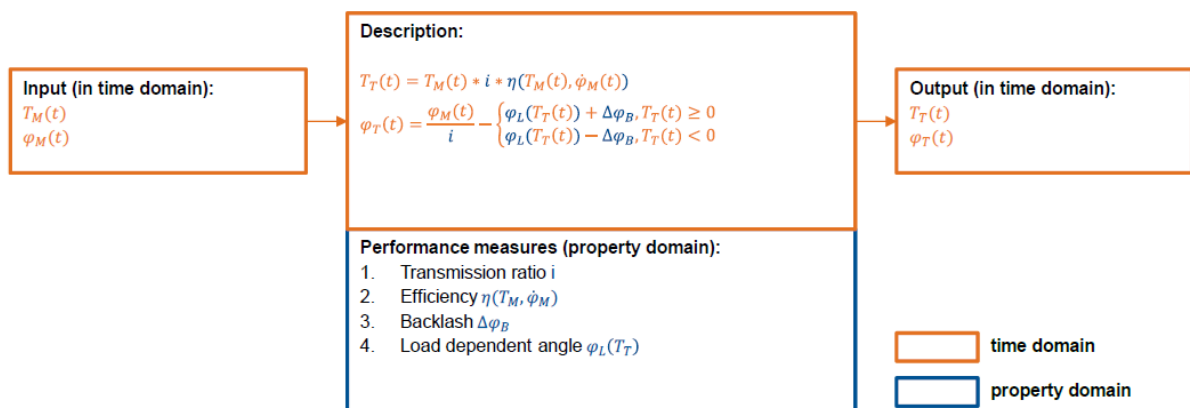


Abbildung 7: Komponentenverhaltensmodell des Getriebes T10 [1]

Das Gesamtsystem bildet die zeitabhängigen Wechselwirkungen eines roboterartigen Systems ab (siehe Abbildung 8). Die Werte, die zwischen den einzelnen Komponentenverhaltensmodellen übergeben werden, können der Abbildung entnommen werden.

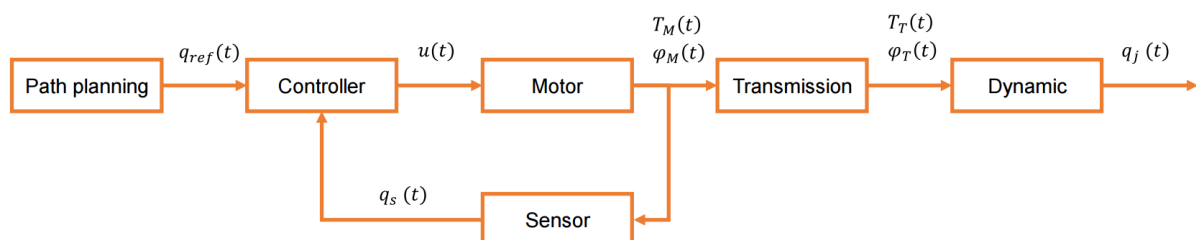


Abbildung 8: Verhaltensmodell des Gesamtsystems [1]

Für die Simulationsmodelle gibt es eine Grundausswahl an Modellen. Diese umfassen Modelle für Motoren, Getriebe, Sensoren und Buse. Die Modelle sind in einer Bibliothek in Simulink hinterlegt.

Die Eigenschaften des zu verwendenden Getriebes können durch Modelle bestimmt werden, welche das Last-Verformungs- und das Wirkungsgradverhalten widerspiegeln.

Zur Erstellung des Simulationsmodells müssen folgende Schritte ausgeführt werden:

1. Definition des Anwendungsbeispiels: In diesem Schritt wird die Roboterkonfiguration festgelegt und beispielsweise die Länge des Armsegments definiert. Außerdem wird die Trajektorie definiert, indem die Koordinatenpunkte und die Zeit vorgegeben werden.
2. Auswahl der Modelle: In diesem Schritt wird festgelegt, welche Modelltypen verwendet werden sollen. Es kann beispielsweise ausgewählt werden, ob ein Sensormodell verwendet werden soll oder ob beispielsweise das Lastverformungsverhalten mit Hysterese-Effekt berücksichtigt werden soll. Die entsprechenden Modelle können aus der Bibliothek in Simulink abgerufen werden.
3. Bedatung der Modelle: Für jedes Modell in der Bibliothek sind Design Variablen definiert. Um einzelne Simulationen durchzuführen, werden Werte für die Design Variablen vorgegeben. Beispielsweise können Getriebe mit unterschiedlichen Übersetzungsverhältnissen simuliert werden.
4. Interpretation der Ergebnisse: Nach einem Simulationsdurchlauf kann das Verhalten des Roboters anhand von Diagrammen bewertet werden. Beispielsweise ist zu sehen, ob die Positionsgenauigkeit mit dem simulierten Roboter erreicht wird.

Je nach Zusammensetzung der Modelle verhält sich das Gesamtmodell unterschiedlich. In Abbildung 9 ist ein beispielhafter Ergebnisverlauf der Simulation gezeigt.

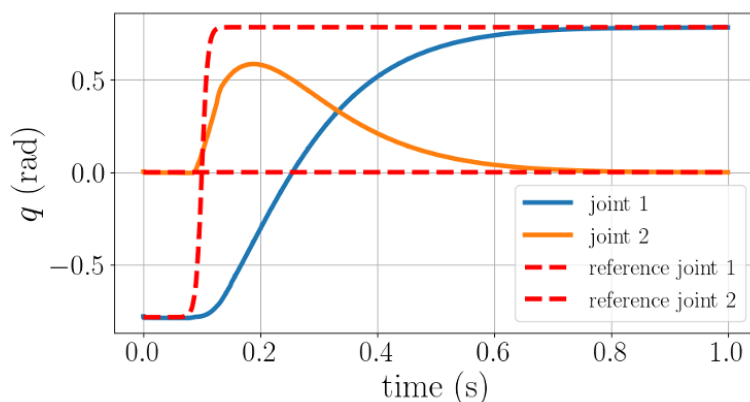


Abbildung 9: Ein beispielhaftes Simulationsergebnis für die SCARA-Roboterbewegung

3.4 Berechnung der Lösungsräume

Das Simulationsmodell kann zur Berechnung von Lösungsräumen verwendet werden. Aus den Lösungsräumen können zulässige Intervalle für die Design Variablen abgelesen werden. Designs, die in den zulässigen Intervallen liegen, erfüllen die zuvor definierten Anforderungen. [6, 7]

Folgende Schritte sind zur Berechnung der Lösungsräume im Rahmen dieses Projekts notwendig:

1. Definition des Design Space: Für jede Design Variable wird der Bereich festgelegt, in dem nach guten Lösungen gesucht werden soll.
2. Lösungsraumberechnung starten
3. Ergebnisinterpretation

Aus den Diagrammen lassen sich die zulässigen Bereiche für jede Design Variable ablesen. In Abbildung 10 sind beispielhafte Lösungsboxen zu sehen.

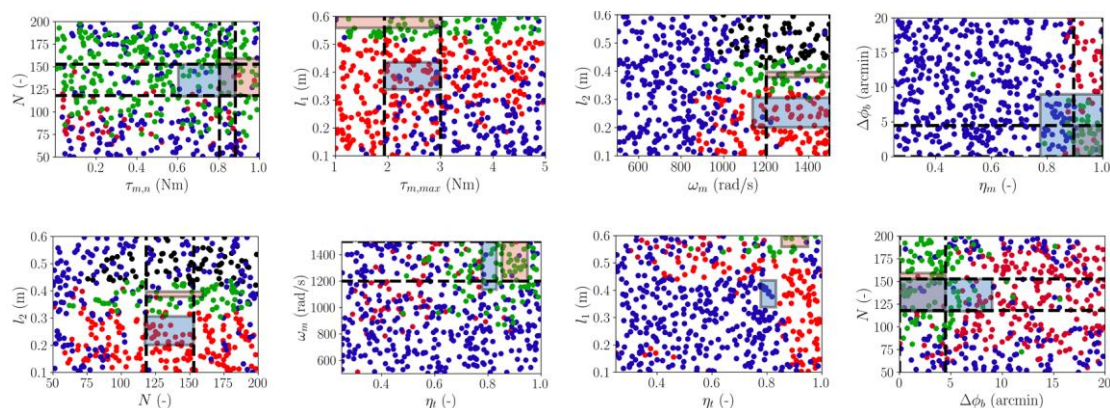


Abbildung 10: Lösungsräume für einen SCARA Roboter

Nachdem die Lösungsräume für jede Variante berechnet wurden, können überlappende Bereiche identifiziert werden. In diesen überlappenden Bereichen können Komponenten für mehrere Varianten gemeinsam genutzt werden, in den anderen Bereichen sind eigene Komponenten erforderlich. In Abbildung 11 ist die Ausführung von zwei beispielhaften Produktausführungen gezeigt. Im linken Teil ist dazu die Zusammenfassung der Eigenschaften der Lösungsräume gezeigt.

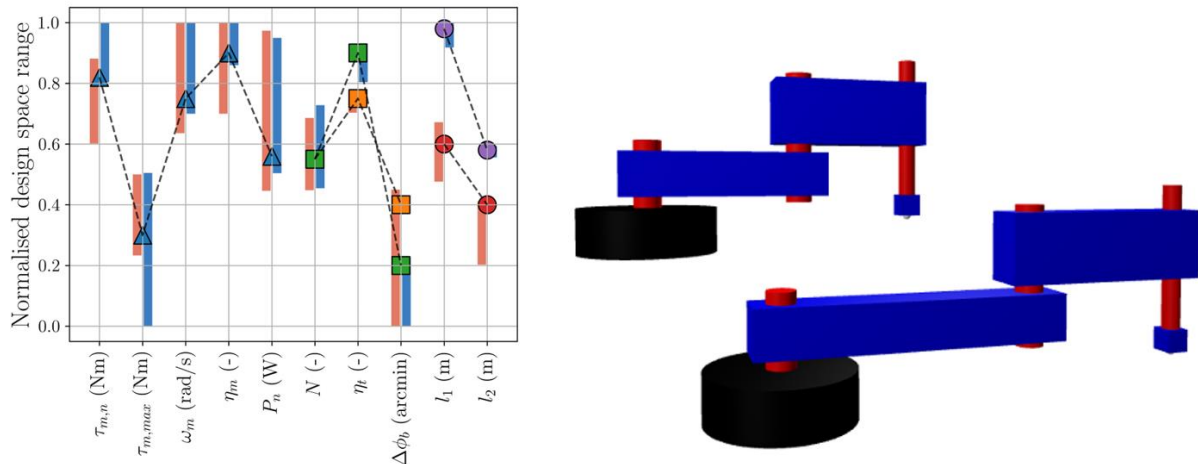


Abbildung 11: Vergleich von zwei Ausführungsvarianten

Zusammenfassend lässt sich sagen, dass das Verfahren zur Durchführung der Produktfamilienauslegung folgende Bestandteile enthält (siehe Abbildung 12):

1. die Identifizierung der Komponentenattribute, die während der Getriebemodellierung im vorherigen Arbeitspaket generiert wurden,
2. die Erstellung von Komponentenverhaltensmodelle und Komponenteneigenschaftsmodellen mit der DSL, die in eine Simulationsumgebung übersetzt werden können und
3. die Abtastung des Konstruktionsraums, um Bereiche zu identifizieren, die in Bezug auf die Anforderungen realisierbar sind, d.h. die Identifizierung von Lösungsräumen, um die physischen Roboter anhand ihrer überlappenden Bereiche zu realisieren.

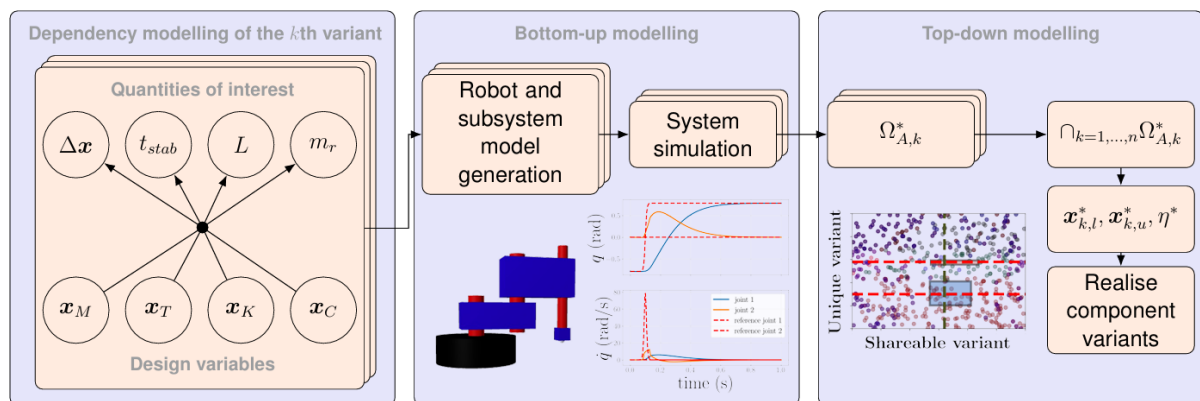


Abbildung 12: Zusammenfassung des Verfahrens zur Generierung von Produktfamilien

Die zulässigen Bereiche werden an die Ontologie übergeben.

3.5 Anwendung der Ontologie

Im letzten Schritt werden mittels Ontologie valide reale Komponentenkombinationen ermittelt. Dies geschieht auf Basis des Inputs aus dem SSE in Form von zulässigen

Lösungsräumen. Hierbei werden nicht nur die Lösungsräume abgefragt, sondern auch weitere Merkmale, die zur Schnittstellenabstimmung notwendig sind, zum Abgleich herangezogen. Grundlage für den Komponentenabgleich ist eine Komponentendatenbank. Ein Beispiel hierfür ist in Abbildung 13 zu finden. Hier wird die Passung der Wellen und die dazugehörige Passfeder und Nut zwischen Motorwelle und Getriebewelle abgeglichen. Weitere binäre und nicht binäre Merkmale werden ebenfalls abgeglichen. Bei der Berechnung wird auf einen effizienten Ansatz gesetzt, der Rechenkapazitäten und Speicherplatz gegeneinander abwägt.

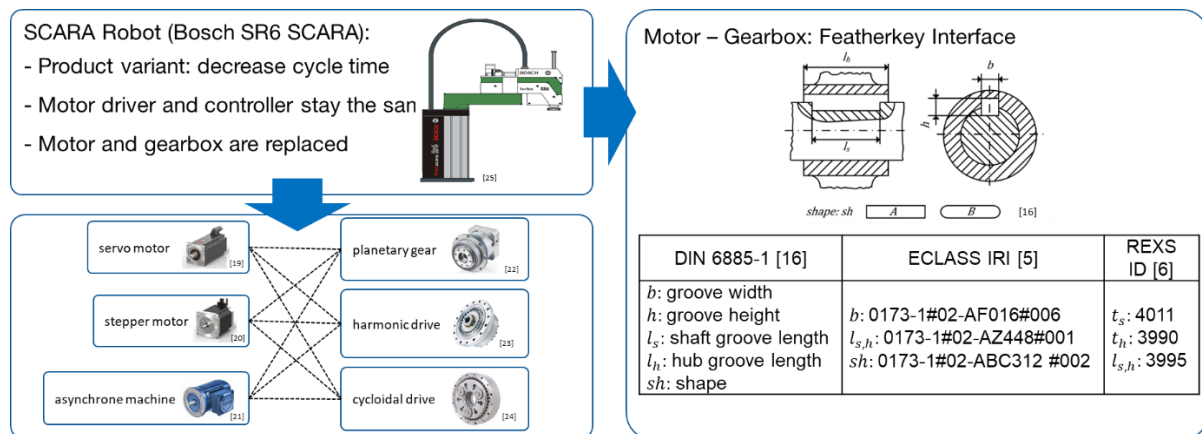


Abbildung 13: Schnittstellenbeschreibung zwischen Motor und Getriebe [8]

Der Ansatz beruht auf verschiedenen Standards aus unterschiedlichen Domänen. Im Detail wurden hier Komponentendefinitionen von REXS (für Getriebe) und ECLASS (für Elektrokomponenten) übernommen.

Die Verwendung einer Ontologie zur Abfrage der real existierenden Komponenten bietet einige Vorteile. Zum einen basiert die Ontologie auf einer Open World Assumption und kann somit, anders als ein einfaches Modell, auch nicht existente oder vorher modelliert Eigenschaften handhaben. Das ist insbesondere interessant, wenn man über neuartige Komponenten nachdenkt, die zum Zeitpunkt des Entwurfs dieses Ansatzes noch nicht existent waren. Auszüge der Wissensbasis sowie die Ergebnisse des Abgleichprozesses sind in Abbildung 14 dargestellt.

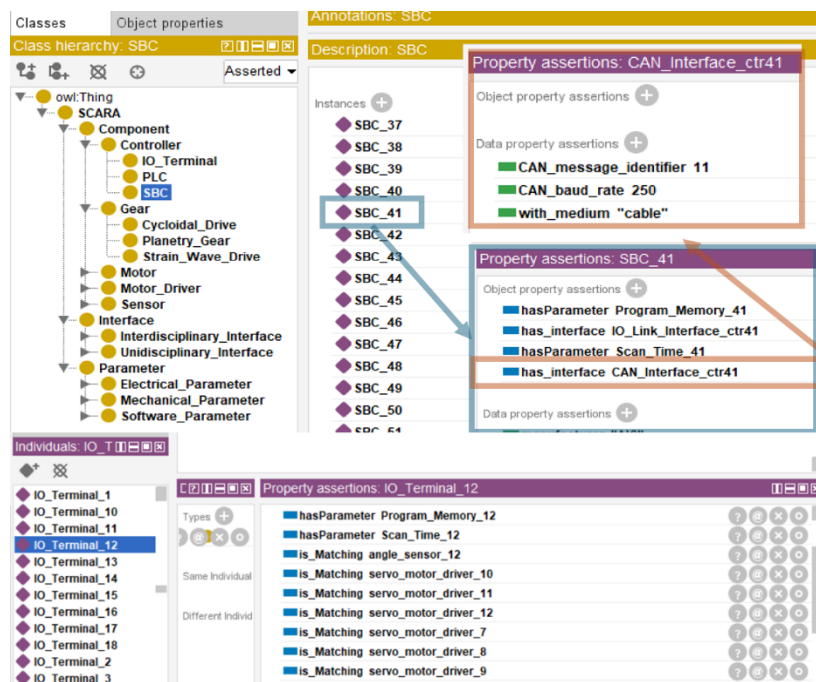


Abbildung 14: SCARA-Antriebselemente und Abgleichergebnisse in der Ontologie

Darüber hinaus werden auch Umwelteinflüsse auf die Komponenten berücksichtigt. Diese sind durch die Modellierung von typischen Einsatzbedingungen in der Wissensdatenbank mit abgebildet [9]. Mit besonderem Fokus auf die Sensorik erlaubt dieser Ansatz die gezielte Analyse von Umweltfaktoren wie Temperatur und Luftfeuchtigkeit und unterstützt fundierte Entscheidungsprozesse bei der Komponentenauswahl. Zusammenfassend zeigt sich, dass der ontologiebasierte Ansatz sowohl die interdisziplinäre Modellierung und Abstimmung von Schnittstelleneigenschaften als auch die Wiederverwendung etablierter Kompatibilitätsbeziehungen ermöglicht. Dadurch werden Effizienz und Skalierbarkeit bei der Auswahl von Antriebselementen für RAS maßgeblich verbessert.

3.6 Bewertung des Konzepts

Der abschließende Schritt der Ergebnisevaluation umfasst die Bewertung des entworfenen Konzepts anhand der berechneten Lösungsräume. Dabei kann die Performance der Lösung mit den zuvor festgelegten Anforderungen (siehe Tabelle 1) an die Anwendung verglichen werden.

4 Möglichkeiten, Grenzen und Generalisierbarkeit

Die Anwendung einer disziplinübergreifenden Beschreibungssprache eröffnet neue Wege in der frühzeitigen Abstimmung des Zusammenspiels einzelner Komponenten im Gesamtsystem. Durch den holistischen Ansatz wird gewährleistet, dass Wechselwirkungen von Beginn an berücksichtigt werden und dadurch ein optimales Komponentendesign sichergestellt werden kann.

Die Vereinigung verschiedenster Disziplinen in einer Modellierungssprache geht allerdings auch mit vielen Herausforderungen einher. So ist es beispielsweise von großer Wichtigkeit, dass die Modelle auf gleichem Abstraktionslevel entwickelt sind. Bei der Modellerstellung wurde genau auf diesen grundlegenden Modellierungsgrundsatz geachtet. Dazu wurde während der Projektlaufzeit eine Metrik eingeführt, die die entwickelten Modelle beispielsweise nach Modellgüte, Komplexität und Kompatibilität bewertet. Anhand dieser Maßnahme konnte sichergestellt werden, dass die Modellierungstiefe modellübergreifend auf gleichem Niveau gehalten wurde.

Die Modelle wurden zudem so entwickelt, dass sie möglichst universell auf verschiedene Anwendungsfälle übertragbar sind. Durch die Basis von allgemeinen Grundelementen können verschiedenste roboterartige Systeme mit der Modellierungssprache beschrieben werden.

5 Zusammenfassung

In diesem Dokument ist die Anwendung der Modellierungssprache für Roboter und roboterartige Systeme zusammengefasst. Anhand der Beispielanwendung „SCARA-Roboter“ wird der Modellierungs- und Auslegungsprozess mit der Beschreibungssprache auf Basis des Vorgehensmodells (siehe Abbildung 1) systematisch vorgezeigt.

Alle einzelnen Elemente der DSL sind zudem in tabellarischer Form erklärt und beschrieben (siehe Abschnitt 7.1).

6 References

- [1] K. Ziegler, M. Volpert, M. Amm, B. Vogel-Heuser, K. Stahl, and M. Zimmermann, "MBSE INCORPORATING TIME-DEPENDENT BEHAVIOR FOR THE DESIGN OF ROBOT-LIKE SYSTEMS," *Proc. Des. Soc.*, vol. 3, pp. 2585–2594, 2023, doi: 10.1017/pds.2023.259.
- [2] K. Ziegler, K. Stahl, B. Vogel-Heuser, and M. Zimmermann, "Multidisciplinary Design for robot-like Systems," 2023.
- [3] B. Vogel-Heuser, E. Trunzer, D. Hujo, and M. Sollfrank, "(Re)deployment of Smart Algorithms in Cyber–Physical Production Systems Using DSL4hDNCS," *Proc. IEEE*, vol. 109, no. 4, pp. 542–555, 2021, doi: 10.1109/JPROC.2021.3050860.
- [4] K. J. Waldron, G. L. Kinzel, and S. K. Agrawal, *Kinematics, dynamics, and design of machinery*. Chichester, West Sussex: Wiley, 2016.

- [5] S. Landler, R. M. Blanco, M. Otto, B. Vogel-Heuser, M. Zimmermann, and K. Stahl, "Determination of the Characteristics of Gears of Robot-Like Systems by Analytical Description of their Structure," pp. 439–445, doi: 10.1109/IROS55552.2023.10342105.
- [6] M. Zimmermann and J. E. von Hoessle, "Computing solution spaces for robust design," *Numerical Meth Engineering*, vol. 94, no. 3, pp. 290–307, 2013, doi: 10.1002/nme.4450.
- [7] M. Zimmermann *et al.*, "On the design of large systems subject to uncertainty," *Journal of Engineering Design*, vol. 28, no. 4, pp. 233–254, 2017, doi: 10.1080/09544828.2017.1303664.
- [8] M. Volpert, B. Vogel-Heuser, D. Hujo, K. Stahl, and M. Zimmermann, "Compatibility Assessment for Interfaces in Drivetrains of Robot-Like Systems," *Procedia Computer Science*, vol. 232, pp. 2994–3002, 2024, doi: 10.1016/j.procs.2024.02.115.
- [9] B. Vogel-Heuser *et al.*, "A lightweight sensor ontology for supporting sensor selection, deployment, and data processing in forming processes," *Prod. Eng. Res. Devel.*, vol. 18, no. 6, pp. 1007–1021, 2024, doi: 10.1007/s11740-024-01290-2.

7 Anhang

7.1 Symbole der DSL

Tabelle 4 Symbole der Mechanik-Sicht (Adaptiert aus: [4])



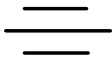
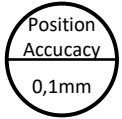



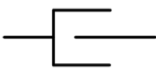
Sicht	Symbol	Spezifikation
Mechanik-Sicht		Verbindungselement. Es ist ein roboterbezogenes Element und es kann sich z.B. um einen Arm, eine Schwinge oder ein Karussell handeln.
Mechanik-Sicht		Drehgelenk mit fluchtender Rotationsachse. Drehbewegung der Rotationsachse erfolgt um sich selbst.
Mechanik-Sicht		Translationsgelenk (Schlitten). Drehung um die Translationsachse ist möglich.
Mechanik-Sicht		Anforderungen an den verbundenen Punkt im mechanischen System. Diese können Zeitanforderungen, Positionsgenauigkeiten, oder Trajektorien sein.
Mechanik-Sicht		Drehgelenk mit fluchtender Rotationsachse. Drehbewegung der Rotationsachse erfolgt um sich selbst.
Mechanik-Sicht		Fixiertes Gelenk. Dies gibt es so üblicherweise nicht, wird aber für Modellierungszwecke eingesetzt. Es ist z.B. ein virtuelles zwischen Grundgestell des Roboters und Fundament, üblicherweise mit 0 nummeriert.
Mechanik-Sicht		Festes Gelenk.
Mechanik-Sicht		Translationsgelenk (Zylinder). Drehung um die Translationsachse ist möglich.

Tabelle 5 Symbole der Komponenten-Sicht




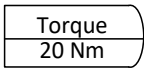
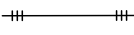


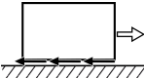
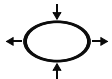
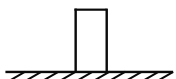
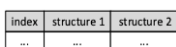

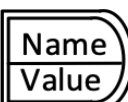
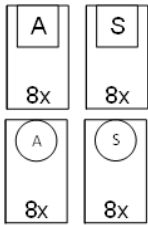


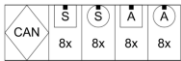




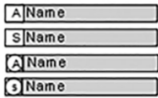
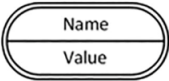
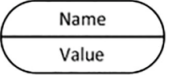

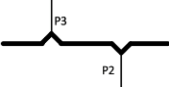

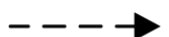
Sicht	Symbol	Spezifikation
Komponenten-Sicht		Getriebe. Kann expandiert werden in der genauen Beschreibung des Getriebes wie im Folgenden. Die spitze Seite des Dreiecks zeigt in Richtung a Seite, die breite in Richtung b Seite.
Komponenten-Sicht		Servo Motor. Kann mittels Komponentenschnittstelle an Mechanik und Steuerung verbunden werden. Enthält Eigenschaften wie Nennmoment, Schnittstellenbeschreibung und mehr.
Komponenten-Sicht		Wechselstrom Motor. Kann mittels Komponentenschnittstelle an Mechanik und Steuerung verbunden werden. Enthält Eigenschaften wie Nennmoment, Schnittstellenbeschreibung und mehr.
Komponenten-Sicht		Eigenschaftsbeschreibung für Komponenten. Visualisiert die Eigenschaften, die einer Komponente zugeordnet sind.
Komponenten-Sicht		Komponenten Schnittstelle. Verbindet die Komponenten der Einzelnen Sichten.
Komponenten-Sicht		Rollende Bewegung (z.B. bei einem Rillenkugellager).
Komponenten-Sicht		Kontakt von zwei Zahnflanken (Kombination aus Rollen und Gleiten).
Komponenten-Sicht		Reines Gleiten zwischen zwei Kontaktpartner.
Komponenten-Sicht		Große Verformung von zwei Kontaktpartner.
Komponenten-Sicht		Keine Relativbewegung zwischen zwei Kontaktpartnern.
Komponenten-Sicht		Vergleichstabelle von Eigenschaften verschiedener Getriebealternativen.
Komponenten-Sicht		Multiplexer stellt ein digitales Schaltelement dar, das mehrere Eingangssignale zu einem einzelnen zusammenfasst.
Komponenten-Sicht		Anforderungsbeschreibung für Komponenten. Visualisiert die Anforderungen, die einer Komponente zugeordnet sind.

Tabelle 6 Symbole der Steuerungs-Sicht (Aus: [3])

Sicht	Symbol	Spezifikation
Steuerungs-Sicht		Ein- und Ausgangsklemmen (E/A). Der Signaltyp wird durch die Form und den Buchstaben oben angegeben <ul style="list-style-type: none"> • Digitaler Aktoranschluss (Quadrat A) • Digitaler Sensoranschluss (Quadrat S) • Analoger Aktoranschluss (Kreis A) • Analoger Sensoranschluss (Kreis S) Die Zahl auf der Unterseite gibt die Anzahl der verfügbaren E/A-Kanäle an.
Steuerungs-Sicht		Die speicherprogrammierbare Steuerung (SPS) führt eine Steuerungsaufgabe aus.
Steuerungs-Sicht		Buskoppler mit Ethernet Interface. Die Interfaces werden über den Namen in der Mitte des Symbols bestimmt.
Steuerungs-Sicht		Bus Adapter mit I/O-modules ohne Steuerungsaufgaben.
Steuerungs-Sicht		Analogsensor
Steuerungs-Sicht		Digitalsensor
Steuerungs-Sicht		Digitalaktor
Steuerungs-Sicht		Analogaktor
Steuerungs-Sicht		Definition von Signalen
Steuerungs-Sicht		Communication / Application Time Requirement
Steuerungs-Sicht		Communication / Application Time Property
Steuerungs-Sicht		Controller Description Connection

Sicht	Symbol	Spezifikation														
Steuerungs-Sicht	<table><tr><th colspan="2">Beckhoff CX2040</th></tr><tr><td>Type</td><td>Intel i7 2715QE</td></tr><tr><td>Cores</td><td>4</td></tr><tr><td>Clock Rate</td><td>2.1 GHz</td></tr><tr><td>Instruction Set</td><td>x64</td></tr><tr><td>RAM</td><td>4 GB</td></tr><tr><td>Flash</td><td>16 GB</td></tr></table>	Beckhoff CX2040		Type	Intel i7 2715QE	Cores	4	Clock Rate	2.1 GHz	Instruction Set	x64	RAM	4 GB	Flash	16 GB	Controller Hardware Description
Beckhoff CX2040																
Type	Intel i7 2715QE															
Cores	4															
Clock Rate	2.1 GHz															
Instruction Set	x64															
RAM	4 GB															
Flash	16 GB															
Steuerungs-Sicht		Fieldbus														
Steuerungs-Sicht		Sensor / Actuator Description Connector														
Steuerungs-Sicht		Logical Connection														