

METHODE ZUR EVOLUTIONÄREN OPTIMIERUNG DER WIRKSTRUKTUR MECHATRONISCHER SYSTEME

Rafael Radkowski, J. Gausemeier

Zusammenfassung

Der Beitrag beschreibt eine neuartige Methode für den automatisierten Entwurf mechatronischer Systeme mit Hilfe evolutionärer Algorithmen. Der verwendete Algorithmus basiert auf der Genetischen Programmierung und wird detailliert beschrieben. Für die Anwendung ist ein Vorgehensmodell entwickelt worden, das den automatisierten Entwurf in einen Produktentwicklungsprozess integriert. Die Funktionsfähigkeit wurde mit einem Softwarewerkzeug nachgewiesen.

1 Einleitung

Mechatronik fördert im hohen Maße Produktinnovationen im Maschinenbau und verwandten Branchen. Durch das enge Zusammenspiel von Mechanik, Elektronik und Informationsverarbeitung sind neue Funktionen realisierbar und vorhandene einfacher, flexibler und kostengünstiger zu verwirklichen [3]. Allerdings bewegen sich Entwickler mechatronischer Systeme beim Entwurf neuer Systeme in einem Spannungsfeld: Sie sind auf der einen Seite dazu gezwungen zuverlässige Lösungen zu erstellen. Auf der anderen Seite zwingt der Wettbewerb zu Innovationen. In Zeiten steigenden Kostendrucks und immer kürzer werdender Entwicklungszeiten gilt die Devise „Keine Experimente“. Es wird in der Regel eine bewährte Lösung als Basis verwendet, die dann modifiziert wird. [1], [7]

Technische Systeme bestehen aus einer Komposition von so genannten Lösungselementen/ Systemelementen [3]. Dies gilt auch für Neukonstruktionen. Das Neue ergibt sich zum Teil auch durch eine geschickte Kombination von Systemelementen. Allerdings ist die Anzahl der heute zur Verfügung stehenden Systemelemente extrem hoch. Mit einem diskursiven Entwicklungsansatz können kaum alle prinzipiell in Frage kommenden Lösungen betrachtet werden. Der Lösungsraum ist zu groß: Man stelle sich einen morphologischen Kasten mit 50 Funktionen und 25 mögliche Systemelementen pro Funktion vor. Es wäre kaum möglich alle in Frage kommenden Elemente und deren Kombinationsmöglichkeiten in einem angemessenen Zeitrahmen zu analysieren.

In diesem Aufsatz wird ein neuartiger Ansatz für den automatisierten Entwurf der Wirkstruktur mechatronischer Systeme mit Hilfe evolutionärer Algorithmen¹ (EA) vorgestellt. Der Ansatz beruht auf der Hypothese, dass mechatronische Systeme grundsätzlich aus einer Komposition bestehender Systemelemente bestehen. Der realisierte evolutionäre Algorithmus basiert auf der von John Koza entwickelten Genetischen Programmierung, die die Prinzipien der biologischen Evolution zur Optimierung von Computerprogrammen verwendet [5].

Durch den automatisierten Entwurfsprozess kann rechnerunterstützt eine erheblich größere Anzahl von Systemelementen und deren Kombinationsmöglichkeiten in kürzerer Zeit

¹ Evolutionäre Algorithmen bilden eine Klasse von Verfahren, die sich die Prinzipien der biologischen Evolution zum Vorbild nehmen, um primär Optimierungsprobleme zu lösen [4].

untersucht werden, als beim konventionellen Entwurf. Dem Entwickler werden nur die prinzipiell besten Lösungen präsentiert, wodurch der Lösungsraum stark eingeschränkt wird. Diese vorgeschlagenen Lösungen können dann von den Entwicklern analysiert werden und als Grundlage für die weitere Konstruktion dienen.

In diesem Beitrag wird der Algorithmus zur Optimierung der Wirkstruktur beschrieben. Kapitel 2 beschreibt die Repräsentation mechatronischer Systeme, die zur evolutionären Optimierung benötigt wird. Kapitel 3 stellt die evolutionäre Optimierungsmethode dar. Kapitel 4 fasst die Ergebnisse zusammen und gibt einen kurzen Ausblick auf die nächsten Entwicklungsschritte.

2 Repräsentation mechatronischer Systeme

Die Erstellung einer geeigneten Repräsentationsform – in Anlehnung an die Evolutionstheorie im folgenden auch als Chromosom bezeichnet – ist eine der Hauptaufgaben beim Entwurf eines evolutionären Algorithmus. Wird an dieser Stelle kein vernünftiger Ansatz gewählt, werden auch keine zufrieden stellenden Ergebnisse erzielt [10]. Zur Repräsentation des mechatronischen Systems – im Folgenden auch als Individuum bezeichnet – wird deshalb die Wirkstruktur nach Frank/Kallmeyer verwendet ([2]), dessen Semantik für den evolutionären Entwurfsprozess angepasst wurde.

2.1 Systemelemente und Verbindungen

Die Systemelemente des mechatronischen Systems werden durch die Knoten eines Graphen repräsentiert. Zur Beschreibung der Wirkungsweise werden für jedes Systemelement Eigenschaften angegeben. Die Eigenschaften sind in zwei Kategorien aufgeteilt: In Eigenschaften zur Beschreibung der Gestalt und in Eigenschaften zur Beschreibung des Verhaltens:

- Die Grobgestalt wird vereinfacht durch ein Hüllvolumen beschrieben, das das Systemelement umschließt. Zusätzlich werden Positionen definiert an denen kinematische Verbindungen zu anderen Systemelemente möglich sind.
- Das Verhalten des Systemelements wird mit Hilfe von speziellen Funktionen beschrieben. Dabei handelt es sich um eine Beschreibung der Funktionalität mit Hilfe von technischen Verben nach Roth u.a. wie z.B. „transportieren“ oder „umwandeln“ und einem Substantiv. Als Substantive werden lediglich die Größen technischer Einheiten verwendet, z.B. „Moment“ (Nm) oder „Geschwindigkeit“ (v). Für die Analyse des Systems wird jede Funktion durch eine mathematische Funktion und ein oder mehrere Randwerte/ Parameter konkretisiert (Anzahl ist abhängig von der mathematischen Funktion). Die Funktionen werden noch in Haupt-, Neben- und Störfunktionen eingeteilt. Ein Beispiel ist in Bild 1 dargestellt. [8], [9]

Neben diesen Daten sind noch weitere allgemeine Angaben wie z.B. eine Bezeichnung, eine schematische Abbildung und das Gewicht des Systemelements enthalten. Zur Verbindung mit anderen Systemelementen besitzt jedes Element Ein- und Ausgänge über die Stoff-, Energie- oder Signalflüsse übertragen werden kann. Die Anzahl der Ein- bzw. Ausgänge ist von der Anzahl der Variablen der math. Funktion bzw. von der Anzahl der Funktionen selbst abhängig. Der Asynchronmotor aus Bild 1 besitzt z.B. eine Funktion zur Berechnung eines Drehmoments. Das Systemelement hat demzufolge nur einen Ausgang.

Die Verbindungen zwischen den Systemelementen, das können z.B. Verbindungskabel, Rohrleitungen oder Gelenke sein, verknüpfen die Systemelemente zu einer Wirkstruktur. Zur genaueren Beschreibung der Wirkungsweise werden die drei Verbindungsarten Stoff-,

Energie- und Signalfloss verwendet und durch Randbedingungen konkretisiert. Jede Kante kann dabei nur eine Flussart beschreiben. Was übertragen wird ist abhängig vom Aus- bzw. Eingang der verbundenen Knoten (Der Asynchronmotor aus Bild 1 würde ein Moment übertragen/ Energie leiten). Für jede Verbindung kann mit Hilfe eines Vektors zusätzlich eine Wirkungsrichtung definiert werden. Dadurch werden z.B. Freiheitsgrade mechanischer Gelenkverbindungen beschrieben. [9]

Funktion	Typ	Math. Beschreibung	Parameter	Wert
Moment erzeugen	H	$T = 9550 \cdot U \cdot I / n$	Leistung P	15 kW
			Stromaufnahme	20A
			Spannung	230V
			Max. Drehzahl	2500 U

Bild 1: Symbol und Beispiel einer Funktionsbeschreibung für einen Asynchron-Elektromotor

2.2 Repräsentation eines mechatronischen Systems

Die Wirkstruktur eines mechatronischen Gesamtsystems (ein Individuum) ist beispielhaft in Bild 2 dargestellt.

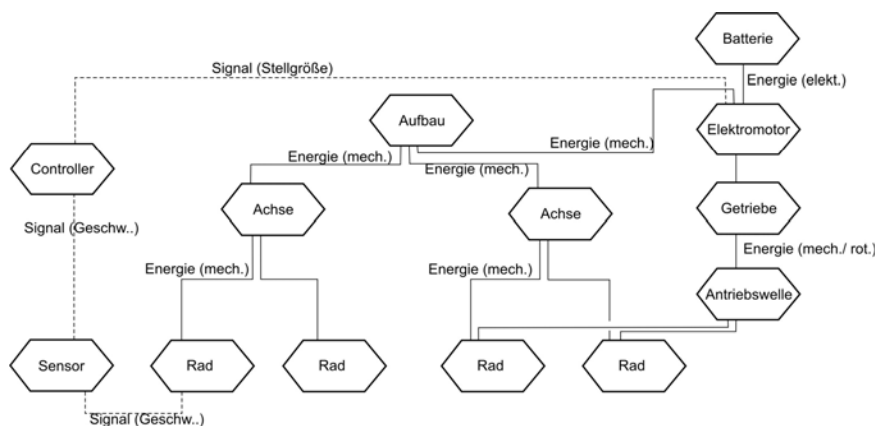


Bild 2: Schematische Darstellung der Wirkstruktur eines mechatronischen Systems.

Es handelt sich hier um ein einfaches Fahrzeug, das über einen Elektromotor betrieben wird. Die Geschwindigkeit des Fahrzeugs wird von einem Controller geregelt. Das Gesamtsystem besteht aus 13 Systemelementen die über 15 Kanten miteinander verbunden sind. Mehrfachverbindungen zwischen den Systemelementen sind ebenfalls zulässig. Anhand der entstandenen Struktur und der Daten die in den Systemelementen abgebildet sind, kann der evolutionäre Algorithmus die Wirkungsweise des mechatronischen Systems ermitteln.

3 Ablauf des Entwurfsprozesses

Um den Entwurfsprozess nach den Prinzipien der biologischen Evolution zu automatisieren sind sechs Schritte notwendig. Diese sind im Bild 3 als Phasen-Meilenstein Diagramm dargestellt.

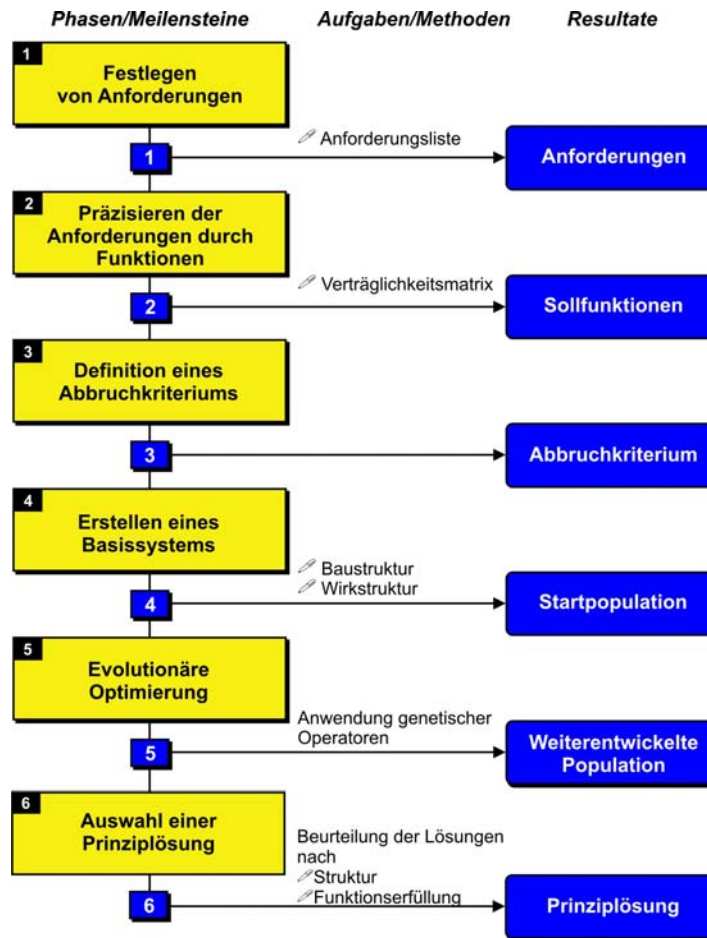


Bild 3: Vorgehen beim automatisierten Entwurf der Wirkstruktur mechatronischer Systeme

Die ersten vier Schritte sind zur Vorbereitung des evolutionären Optimierungsprozesses notwendig: Die Festlegung von Anforderungen, die Präzisierung von Anforderungen durch Funktionen, die Definition eines Abbruchkriteriums und die Erstellung eines Basissystems. Die darauf folgende evolutionäre Optimierung ist in Abschnitt 4 beschrieben. Dessen Aufbau basiert auf der von John Koza entwickelten genetischen Programmierung [5], [6]. Resultat des Optimierungsprozesses ist eine weiterentwickelte Population von Individuen. In der Regel ist mehr als eine Lösung für den Anwender brauchbar. Durch Beurteilung der Struktur des Systems und dem Erfüllungsgrad der Anforderungen kann die Auswahl weiter eingeschränkt werden.

- 1) Festlegen der Anforderungen: Die Rahmenbedingungen für die evolutionäre Optimierung werden durch Anforderungen, die an das mechatronische System gestellt werden, definiert. Die Anforderungen werden in einer Anforderungsliste festgehalten. Ziele und Bedingungen an das mechatronische System müssen durch Anforderungen in Form von Forderungen (F) und Wünschen (W) klar herausgearbeitet werden. Um ausreichend Informationen für die evolutionäre Optimierung zu erhalten sollten die Anforderungen so weit wie möglich durch qualitative oder quantitative Aspekte konkretisiert werden. [8]
- 2) Präzisierung der Anforderungen: Die Anforderungen werden über Funktionen konkretisiert. Die Funktionen werden durch ein Substantiv, das einer Größe einer technischen Einheit entspricht und einem technischen Verb beschrieben. Eine Anforderung kann durch eine oder mehrere Funktionen detailliert werden.

- 3) Definition eines Abbruchkriteriums: Das Abbruchkriterium legt fest, wann die – theoretisch unendliche – evolutionäre Optimierung abgebrochen wird und das bis dahin gefundene Individuum als „optimal“ interpretiert wird. Die Bestimmung eines Abbruchkriteriums erweist sich jedoch oft als schwierig: Aufgabe eines evolutionären Algorithmus ist es, eine optimale Lösung zu einem vorgegebenen Problem zu finden. Jedoch kann nicht sichergestellt werden, dass die potentielle Lösung die Optimale ist. Oftmals existiert überhaupt keine optimale Lösung. Die ermittelte Lösung stellt meistens lediglich einen Kompromiss zwischen den unterschiedlichen Anforderungen dar. Das Finden eines Optimums kann deshalb nicht als Abbruchkriterium für die evolutionäre Optimierung verwendet werden.
- 4) Erstellen eines Basissystems: Ein Programm, das mit Hilfe eines evolutionären Algorithmus versucht eine Lösung für ein Problem zu erstellen, benötigt zu Beginn ein Basissystem, das den Startpunkt einer Optimierung darstellt. Das Startsystem kann entweder vom Anwender erstellt oder per Zufall vom Programm zusammengebaut werden. Der Anwender kann beispielsweise eine ihm bekannte Lösung vorgeben, die durch den evolutionären Algorithmus verbessert werden soll. Zur Erstellung des Basissystems können Systemelemente aus einer Systemelemente-Bibliothek² ausgewählt werden. Aus dem Basissystem wird in einem weiteren Schritt eine Startpopulation³ für die evolutionäre Optimierung erstellt.
- 5) Evolutionäre Optimierung: Die Wirkstruktur wird unter Verwendung eines evolutionären Algorithmus optimiert. Der Algorithmus wird genauer in Abschnitt 4 beschrieben.
- 6) Auswahl einer Prinzipallösung: Resultat des evolutionären Optimierungsprozesses sind eine oder mehrere Wirkstrukturen, die die Anforderungen erfüllen. Der Anwender kann nach Beendigung des Prozesses durch Ändern von Optimierungsparametern die Auswahl noch weiter einschränken und ein oder mehrere Wirkstrukturen auswählen.

4 Evolutionärer Optimierung

Mit Hilfe des Evolutionären Algorithmus werden die Individuen in einem iterativen Prozess optimiert (Bild 4).

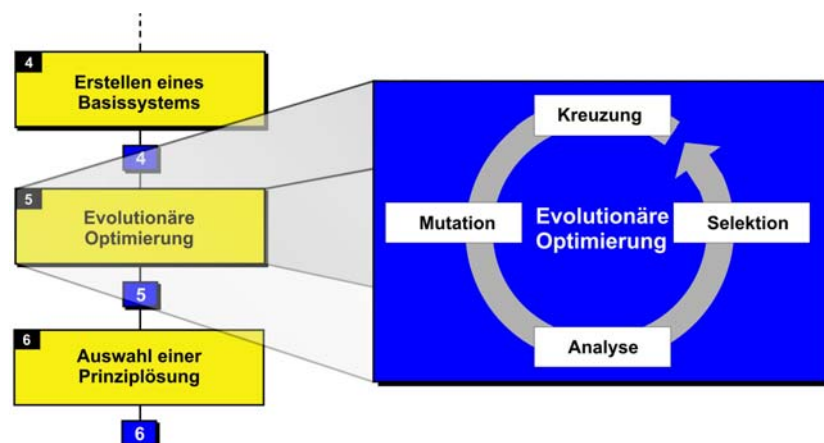


Bild 4: Ablauf der evolutionären Optimierung der Wirkstruktur mechatronischer Systeme.

Um während des Prozesses Veränderungen an den Individuen vornehmen zu können werden genetische Operatoren verwendet. In Anlehnung an den biologischen Ursprung werden diese Kreuzung, Mutation und Selektion bezeichnet. Alle drei Operatoren sind für

² Zur Verwaltung der Systemelemente wurde eine SQL-Datenbank implementiert.

³ Population: Die Gesamtheit aller Individuen wird als Population bezeichnet.

diesen Anwendungsfall angepasst worden. Während der Kreuzungs- und der Mutations-Operator Veränderungen am Aufbau der Wirkstruktur vornehmen, sorgt der Selektions-Operator dafür, dass nur die besten Individuen in einen nachfolgenden Iterationsschritt übernommen werden. Die Verbesserung der Individuen und damit die Optimierung der Wirkstruktur erfolgt mit fortschreiten des Prozesses durch das ständige Wechselspiel zwischen Kreuzung/Mutation und Selektion. Da durch die Selektion nur die besten Individuen in einen nachfolgenden Iterationsschritt übernommen werden, erhöht sich die Wahrscheinlichkeit von Schritt zu Schritt, dass durch die Kreuzung noch bessere Individuen entstehen. Welche Individuen die Besten sind wird in einem Analyseschritt ermittelt. Die Eigenschaften der entstandenen mechatronischen Systeme werden untersucht und mit den Anforderungen verglichen. Aus den Ergebnissen wird ein Fitnesswert berechnet der über die weitere Verwendung des Individuums entscheidet.

Im Weiteren Verlauf werden die genetischen Operatoren und die Analyse erläutert.

4.1 Kreuzungs-Operator

Der Kreuzungs-Operator bietet die Möglichkeit, die Systemelemente zweier Individuen miteinander zu kombinieren. Dabei werden aus jeweils zwei Individuen der Population durch Austausch von mehreren Systemelementen zwei neue Individuen erzeugt (Bild 5).

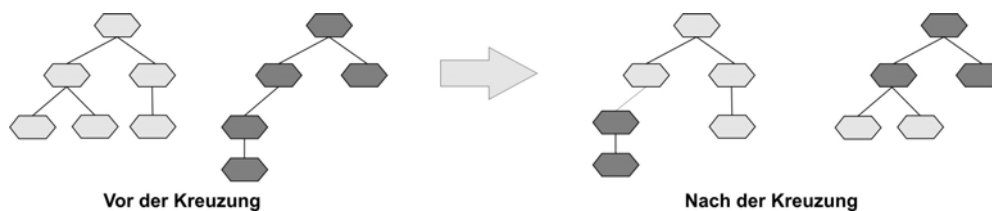


Bild 5: Beispiel für eine Kreuzungs-Operation zweier Wirkstrukturen

Der implementierte Algorithmus arbeitet in vier Schritten: Zuerst werden in zwei für den Kreuzungsvorgang ausgewählten Individuen mehrere zusammenhängende Systemelemente ausgewählt. Dazu wird per Zufall ein Systemelement selektiert, von dem ausgehend eine Teilwirkstruktur erstellt wird. Die Teilwirkstrukturen beider Individuen werden kombiniert wodurch zwei neue Individuen entstehen. Welche Systemelemente miteinander kombiniert werden bleibt dabei dem Zufall überlassen. Die Anzahl der Individuen-Paare die in einer Population pro Zyklus kombiniert werden sollen wird per Zufall bestimmt. Die Anzahl kann allerdings die Hälfte der Populationsgröße nicht überschreiten.

4.2 Mutations-Operator

Bei der Mutation wird ein Systemelement durch einen neues, zufällig gewähltes Systemelement ersetzt (Bild 6). Auf diese Weise werden neue Systemelemente in den Entwurfsprozesse eingefügt. Dadurch werden neue Eigenschaften in die Population integriert und es erhöht sich die Wahrscheinlichkeit, ein an die Anforderungen optimal angepasstes Individuum zu erhalten. Dabei wird wie folgt vorgegangen: Mutationen treten in der Biologie zufällig auf. Darum wird für jedes Individuum mittels Wahrscheinlichkeitsrechnung bestimmt, ob eine Mutation stattfinden soll. Findet eine Mutation statt, wird festgelegt (per Zufallszahl) wie viele Systemelemente ausgetauscht werden. Anschließend erfolgt der Austausch. Welche neuen Verbindungen entstehen wird dabei ebenfalls nicht vorgegeben. Falls Verbindungen offen bleiben, da sie nicht mehr an den neuen Knoten angeschlossen werden können, werden sie gelöscht.

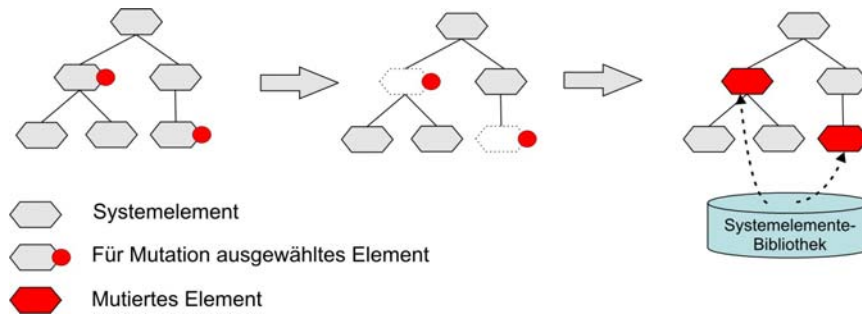


Bild 6: Mutation eines Individuums

Die Systemelemente für die Mutation werden einer Systemelemente-Bibliothek entnommen.

4.3 Analyse der Wirkstruktur

Während die genetischen Operatoren lediglich den Aufbau der Wirkstruktur verändert haben, ohne die Bedeutung der Elemente zu berücksichtigen, werden bei der Analyse die Eigenschaften des entstandenen mechatronischen Systems berücksichtigt. Ergebnis der Analyse ist der Fitnesswert des Individuums. Die Analyse ist in zwei Teilanalysen aufgeteilt: In die Analyse der einzelnen Systemelemente und in die Analyse der Verbindungen. Jede Analyse liefert eine Teilfitness welche am Ende zu einem Gesamtfitnesswert zusammengefasst werden.

4.3.1 Fitnessbestimmung der einzelnen Systemelemente

Der Algorithmus überprüft, ob die vom Systemelement erfüllten Funktionen zur Anforderungserfüllung beitragen (Bild 7). Dazu werden die Funktionen des Systemelements und die Anforderungen über eine Verträglichkeitsmatrix verglichen.

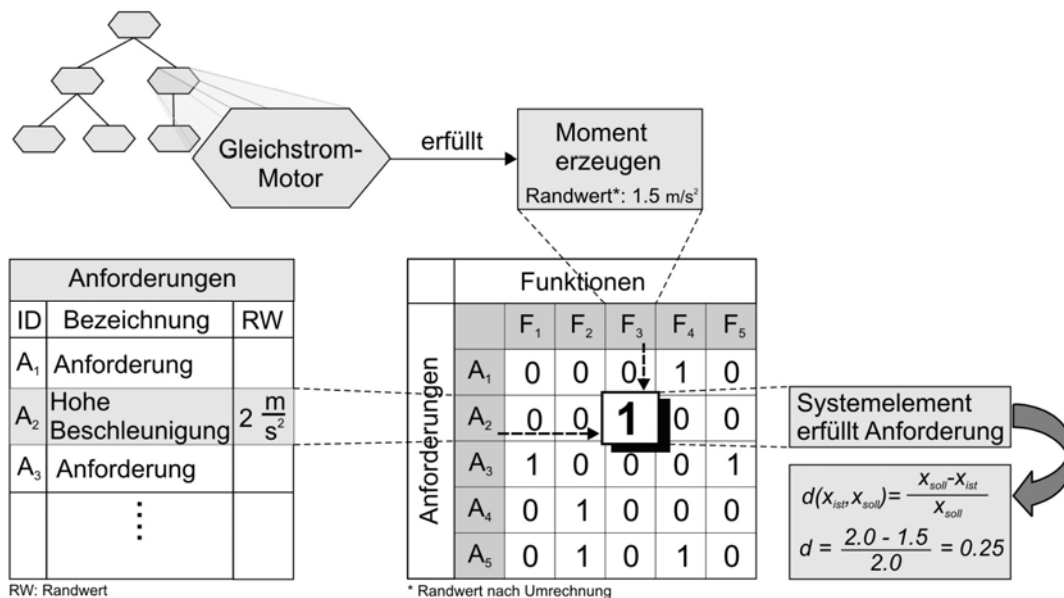


Bild 7: Vorgehen bei der Analyse einzelner Systemelemente

Erfüllt das Systemelement eine Anforderung werden die Randbedingungen verglichen: Dazu wird die Differenzwert *diff* zwischen den Randwerten des Systemelements und den Randwerten aus den Anforderungen berechnet:

$$d(x_{ist}, x_{soll}) = \frac{x_{soll} - x_{ist}}{x_{soll}}$$

Der Wert x_{soll} entspricht dabei der Randbedingung, x_{ist} ist der Grenzwert des Systemelements. Sind mehrere Randbedingungen enthalten (z.B. ein min- und ein max-Wert) werden mehrere Teilfitnesswerte berechnet.

Die Berechnung erfolgt in beiden Fällen mittels linearer statischer Fitness:

$$fitness_{SE,i}(diff) = a * diff + b$$

Da es sich um eine Mehrkriterienoptimierung handelt, werden die einzelnen Teilfitnesswerte unter Berücksichtigung der Gewichtung der jeweiligen Anforderung summiert:

$$fitness_{SE,j} = \sum_{i=0..k} fitness_{SE,i} * gew$$

Der Parameter *gew* entspricht der Gewichtung der jeweiligen Anforderung. Störfunktionen werden auf die gleiche Weise bearbeitet, jedoch führen sie zu einer Abnahme des Fitnesswertes. Hauptgebrauchsfunktionen und Nebenfunktionen die nicht in den Anforderungen enthalten sind, werden nicht berücksichtigt. [4]

4.3.2 Fitnessbestimmung der Struktur

Hierbei wird überprüft ob die Struktur des Individuums eine sinnvolle Lösung ergibt, d.h. ob alle Verbindungen qualitativ und quantitativ korrekt sind. Dazu werden die Verbindungen und die Wirkungsrichtungen Verbindungen zwischen den Systemelementen überprüft. Für die Überprüfung der Wirkungsrichtung ist vorher der Aufbau der Struktur des Individuums notwendig, was jedoch nicht Thema dieser Veröffentlichung ist.

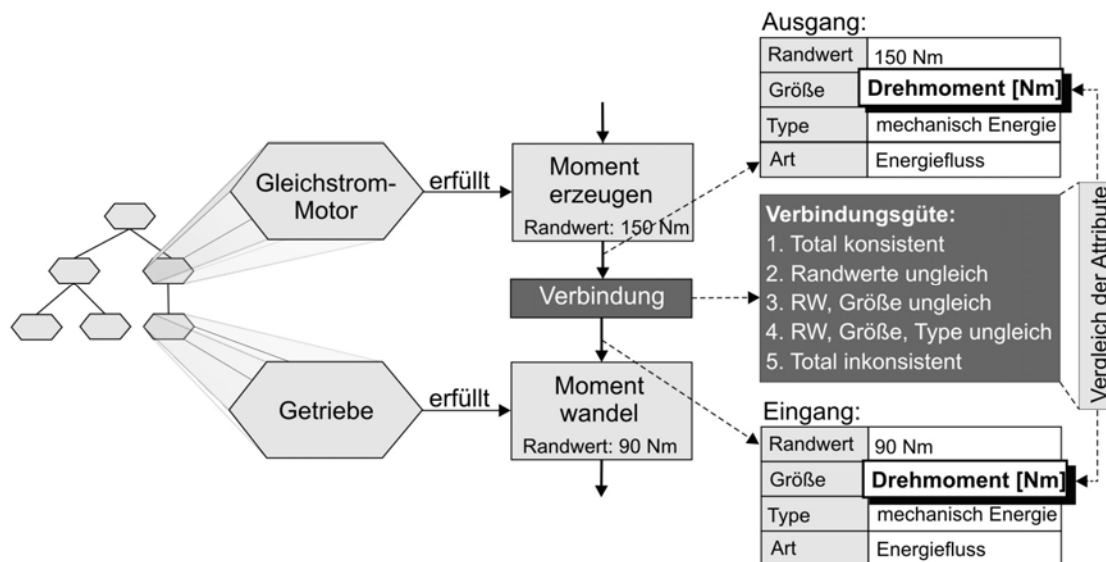


Bild 8: Analyse der Verbindungen zwischen den Funktionen der Systemelemente

Es wird überprüft inwieweit Verbindungsart, Type der übertragenen Größe, die Größe und die Randbedingungen von zwei verbundenen Systemelementen konsistent sind (Bild 8). Je nach Konsistenz der Verbindung wird eine Verbindungsgüte ermittelt. Die Berechnung des Fitnesswertes ist abhängig von der ermittelten Güte. Sie wird mittels exponentieller Skalierung berechnet

$$fitness_{VE,i}(k, diff) = (a * diff + b)^k$$

wobei der Exponent k (-1...1) und der Parameter a von der Güte abhängig sind.

4.4 Selektions-Operator

Während der Analyse ist ein Fitnesswert für jedes Individuum ermittelt worden: Individuen, die die Anforderungen erfüllen und sinnvolle Verbindungen haben, erhalten einen hohen Fitnesswert. Der Selektions-Operator berechnet für jedes Individuum anhand des Fitnesswertes eine Wahrscheinlichkeit, ob es in eine nachfolgende Generation übernommen wird. Dazu wird das Verfahren des Linearen Rankings verwendet: Die Individuen werden auf Basis ihrer Fitness in einer Rangliste geordnet. Die Wahrscheinlichkeit w , dass ein Individuum in die nächste Generation übernommen wird, wird über eine lineare, von seinem Rang abhängige Funktion berechnet:

$$w(pos) = \frac{1}{size} - \left(E_{max} - (E_{max} - E_{min}) \right) \frac{(pos - 1)}{(size - 1)}$$

Die Variable pos entspricht der Position in der Rangliste, $size$ entspricht der Gesamtgröße der Population. E_{max} und E_{min} sind Erwartungswerte über die die Größe der neuen Population gesteuert werden kann. Anschließend wird über eine Wahrscheinlichkeitsrechnung bestimmt welche Individuen in die nächste Generation übernommen werden. [5], [4]

5 Prototypische Umsetzung

Zur Überprüfung der Algorithmen ist ein Softwarewerkzeug entwickelt worden, mit dem die Funktionsfähigkeit des Algorithmus überprüft werden soll. Bild 9 zeigt die Oberfläche dieses Werkzeuges.

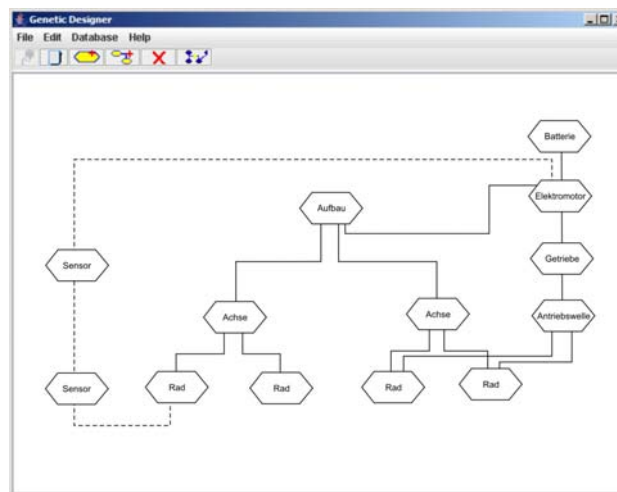


Bild 9: Implementierung der evolutionären Algorithmen in dem Software-Werkzeug „Genetic Designer“

Bei den ersten Tests wurde der Algorithmus allein, d.h. ohne Berücksichtigung der Geometrie des mechatronischen Systems, getestet. Die ersten Problemstellungen waren einfach und dienten dazu, die Funktionsweise einzelner Teilschritte des Algorithmus zu verstehen. Bild 10 stellt die Ausgangssituation und das Ergebnis eines Tests dar: Der Algorithmus sollte vier Räder mit einem Aufbau verbinden. Eine direkte Verbindung würde aufgrund der Ausgänge (Kraft) am Aufbau und Eingänge (Drehmoment) am Rad zu einer ungünstigen Lösung führen.

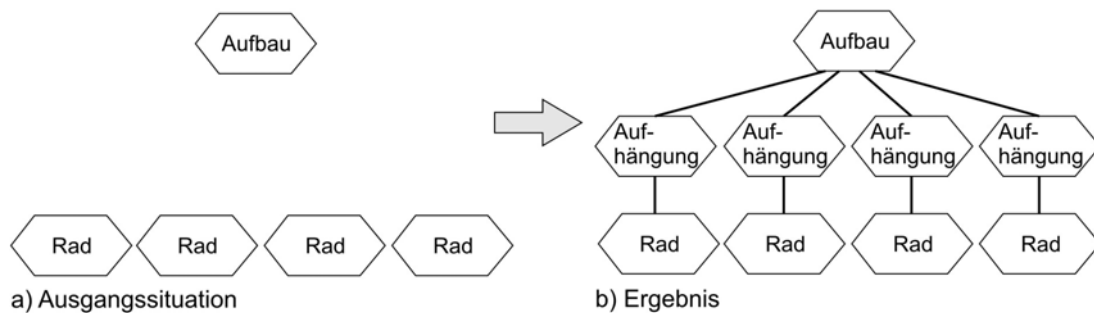


Bild 10: a) Ausgangssituation und b) Ergebnis eines Tests des Algorithmus. Ziel war es eine Verbindung zwischen Rad und Aufbau zu erstellen.

Nach 500 Iterationszyklen ist es dem Algorithmus gelungen ein brauchbares Ergebnis zu erstellen. Anzumerken ist hierbei, dass die Auswahl von Lösungselementen beschränkt ist und das die dargestellte Lösung lediglich eine der zehn besten Lösungen war, wobei andere, die durchaus einen hohen Fitnesswert hatten nicht nutzbar sind. Andere Tests führten zu ähnlichen Ergebnissen.

Zusammenfassend kann gesagt werden, dass der Algorithmus die an ihn gestellten Anforderungen erfüllt. Allerdings ließen sich über eine reine Funktions- und Verbindungsüberprüfung wie sie in Abschnitt 4 beschrieben worden ist lediglich kleine Systeme erstellen (bis zu 15 Elemente). Der Algorithmus neigt dazu, Systemelemente zu bevorzugen, zwischen denen eine direkte Verbindung möglich ist. Mechatronische Regelkreise werden zurzeit noch nicht berücksichtigt.

6 Ausblick

Die Umsetzung und die Tests des Algorithmus haben gezeigt, dass ein automatisierter Entwurf mechatronischer Systeme möglich ist. Es besteht jedoch noch Handlungsbedarf: Beispielsweise werden keinen mechatronischen Regelkreise berücksichtigt und wenn sie durch das Startsystem vorgegeben werden, gehen sie im Laufe des Prozesses verloren. Systemelemente werden teilweise mehrfach integriert, obwohl dies aus Sicht des Anwenders nicht nötig ist. Auf der anderen Seite wurden Systemelemente die mehrfach benötigt worden wären, nicht integriert. Ein Fahrzeug benötigt statt einem Rad vier Räder. Für diese Probleme wird momentan eine Lösung erarbeitet, die diese Fälle berücksichtigt. Des Weiteren wird der Algorithmus um Methoden zur Überprüfung der Geometrie des mechatronischen Systems ergänzt. Dadurch wird die Struktur des Systems viel stärker bei der Analyse berücksichtigt, wodurch auch größere Systeme erstellt werden können.

7 Literatur

- [1] Albers, A.; Ohmer, M.; Eckert, C.: Engineering Desig in a different way: Cognitive Perspective on the Contact & Channel Model Approach. In: Proc. Third International Conference Visual & Spatial Reasoning in Design, MIT, cambridge, 22.-23. Juli 2004
- [2] Frank, U.; Giese, H.; Klein, F.; Oberschelp, O.; Schmidt, A.; Schulz, B.; Vöcking, H.; Witting, K.: Selbstoptimierende Systeme des Maschinenbaus, Definitionen und Konzepte, HNI-Verlagsschriftreihe, Band 155, Paderborn, 2004

- [3] Gausemeier, J.; Ebbesmeyer, P; Kallmeyer, F.: Produktinnovation - Strategische Planung und Entwicklung der Produkte von morgen. Carl Hanser Verlag; München, 2001
- [4] Gerdes, I.; Klawonn, F.; Kruse, R.: Evolutionäre Algorithmen, Vieweg & Sohn Verlag Wiesbaden, 2004
- [5] Koza, J. R.: Genetic Programming, MIT Press, Cambridge, 1992
- [6] Koza, J. R.: Genetic Programming II, MIT Press, Cambridge, 1994
- [7] Oxman, R.: Prior knowledge in design: a dynamic knowledge-based model of design and creativity. Design Studies, 11, S.17.-28., 1990
- [8] Pahl, G.; Beitz, W.: Konstruktionslehre – Methoden und Anwendungen, 5. Auflage, Springer Verlag, Berlin, 2003
- [9] Roth, K.: Konstruieren mit Konstruktionskatalogen, Band 1, 2. Auflage, Springer Verlag, Berlin, 1994
- [10] Schöneburg, E.; Heinzmann, F.; Feddersen, S.: Genetische Algorithmen und Evolutionsstrategien - Eine Einführung in die Theorie und Praxis, Addison-Wesley, London, 1994

Dipl.-Ing. Rafael Radkowski
Heinz Nixdorf Institut, Virtual Reality Group
University of Paderborn
Fuerstenallee 11, D- 33102 Paderborn
Tel: +49-5251-60-6228
Fax: +49-5251-60-6268
Email: rafael.radkowski@hni.upb.de
URL: <http://www.hni.upb.de/vr>

