

# **Zum Einsatz von KI-Methoden zur Lösung von Problemen der Regelungstechnik**

Hanns Sommer, Andreas Kroll

Fachgebiet Mess- und Regelungstechnik, Universität Kassel

34128 Kassel, Mönchebergstr.7

E-Mail: {hans.sommer, andreas.kroll}@mrt.uni-kassel.de

## **Kurzfassung**

Es wird eine allgemeine Entwurfsmethode für Heuristiken zur Lösung von Problemen der Regelungstechnik vorgestellt. Diese Methode ist mit nur sehr geringen Kenntnissen der Regelungstechnik benutzbar. Mittels diesem Beispiel werden die Vor- und Nachteile einer Verwendung von KI-Methoden in den Ingenieurwissenschaften erörtert.

## **1 Einführung**

Wir betrachten wohlgestellte Probleme, bei denen alle Daten zur Bestimmung einer Lösung mit der Aufgabenstellung gegeben sind und fragen:

„Was macht ein Problem schwierig?“ Man findet Versuche in der Literatur, die Schwierigkeit von Problemen zu definieren [1], aber diese Versuche erscheinen in der Praxis wenig sinnvoll, da das Lösen von Problemen verschiedenartige Fähigkeiten erfordert. Im Bild 1 sind diese Fähigkeiten dargestellt:

1. Das Problem muss aus seiner umgangssprachlichen, praktischen Formulierung in eine formale Sichtweise überführt werden.
2. Die Berechnung des Gesuchten erfolgt aus der formalen Sichtweise.

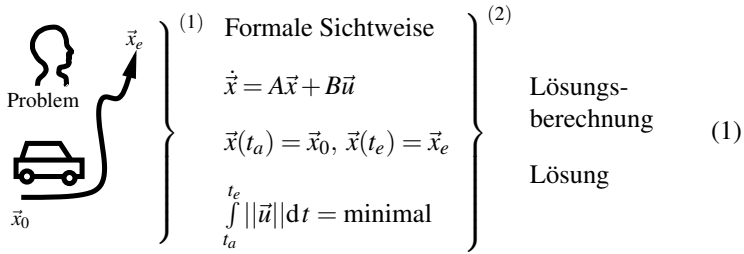


Bild 1: Schritte beim Problemlösen

Die formale Sichtweise stellt einen Verstehenshorizont dar, innerhalb dem dann Berechnungen möglich werden. Das Problem der Suche nach einer Steuerungsfunktion, die  $\vec{x}_0$  in  $\vec{x}_e$  überführt, wird so in ein mathematisches Problem überführt, das im Rahmen der Matrizenrechnung gelöst werden kann. Es ist klar, dass Fähigkeit (2) durch Rechnerunterstützung erheblich erhöht wird und heute vielfach ganz dem Computer übertragen werden kann (vgl. [2]), aber die Ziele der Künstlichen Intelligenz (KI) sind wesentlich weiter, es soll auch Fähigkeit (1) erweitert werden. Da der Mensch immer derjenige bleibt, der dem Computer, in einer diesem verständlichen Weise, sagen muss, was zu tun ist, wird er niemals vollständig vom Computer ersetzt werden, die Kommunikation mit dem Computer ist dagegen vereinfachbar. Mittels einer **Bereitstellung von Berechnungsleistungen für sehr allgemeine Sichtweisen** macht es die KI möglich, dass der Mensch spezielle Einzelheiten seiner Fragestellung nicht mehr erfassen muss und diese, in einer für ihn sehr einfachen Form, an den Computer übergibt. Dadurch soll es dem Menschen ermöglicht werden, ohne tiefere Fachkenntnisse auch kompliziertere Aufgabenstellungen zu lösen.

In unserem Beitrag soll dieses „Versprechen der KI“ an Aufgabenstellungen aus der Regelungstechnik erprobt werden. Dafür wird in Kapitel 2 eine Überführung einer großen Klasse von Aufgaben der Regelungstechnik in Suchprobleme vorgestellt. Nach einer Diskussion der Schwierigkeiten beim Lösen von Suchproblemen in Kapitel 3, werden dann in Kapitel 4 Listen von Prinzipien zum schnellen Lösen von Suchproblemen und zur Koordinierung der Prozessoren bei Multit-Prozessor-Computern vorgestellt. Mittels diesen Listen und ei-

nem in Kapitel 5 dargestellten Schema zur Konzipierung eines Suchverfahrens ist es dann Studierenden, ohne tiefere Kenntnisse aus der Regelungstechnik möglich, Aufgaben dieses Fachgebiets zu lösen. Die Vor- und Nachteile dieser Methode gegenüber den klassischen Lösungsverfahren können damit erkannt werden. Die KI-Methode hat insbesondere das Problem der Validierung. Dieses Problem wird in Kapitel 6 diskutiert, mittels eines Vergleichs der dargelegten Methode mit ChatGPT. Kapitel 7 resümiert die Ideen zur Konzipierung eines allgemeinen Problemlösers.

## 2 Umformung eines allgemeinen Problems in ein Suchproblem

Mit der Methode der Generalisierung von Fragestellungen, lassen sich die Aufgabenstellungen der Regelungstechnik vereinheitlichen. Da diese Aufgabenstellungen mathematisch wohlgestellte Probleme sein müssen, erfüllen sie die folgende **Definition**:

**Definition:** Ein Problem heißt **mathematisch wohlgestellt**, wenn folgende Fragen beantwortet werden können:

- 1. **Wo wird gesucht?**  $\longrightarrow S$  Suchraum
- 2. **Was wird gesucht?**  $\longrightarrow P(s)$  Beschreibung des Gesuchten

Da für Probleme der Regelungstechnik der Suchraum  $S$  und die Beschreibung des Gesuchten, das Prädikat  $P(s)$ , bekannt sind, ist eine Überführung in ein Suchproblem möglich.

Im Bild 2 sind in der ersten Spalte die Problemtypen und in der zweiten Spalte die zugehörigen Spezifikationen von  $S$  und  $P(s)$  angegeben. Die dritte Spalte zeigt die vereinheitlichende Schreibweise, die alle diese Probleme erfasst. Insbesondere bei Mehrgrößensystemen wird die regelungstechnische Lösung sehr komplex.

Das Suchproblem soll mit einem **einheitlichen Lösungsprozess**, ohne spezielle Fachkenntnis aus der Regelungstechnik, gelöst werden. Im nächsten Abschnitt, wird zunächst die Schwierigkeit dieses Problems untersucht.

Regelungstechnik: Steuerungsproblem	$S = \text{Funktionen } \vec{u}(t)$ mit $t \in [ta, te]$ z.B. Treppenfunktionen, Polynome, Splines, ... $P : \vec{u}(t)$ überführt $\vec{x}_0$ in $\vec{x}_e$ und Nebenbedingungen	} Suchproblem $\Pi : s \in S? \quad P(s)$
Polvorgabe- Regler Berechnung	$S = \{F \in \mathbb{R}^{p \times n}\}$ $P: p(\lambda) = \det(\lambda I - (A + BF))$ hat Nullstellen $\lambda_1, \lambda_2, \dots, \lambda_n$ und Nebenbedingungen	
Beobachter- Konstruktion	.....	
$\vdots$		

Bild 2: Überführung von Aufgaben der Regelungstechniker in ein Suchproblem

### 3 Über die Schwierigkeit von Suchproblemen

Eine Problemlösung wird schrittweise erreicht. Bild 3, stellt das Problem der „Suche nach dem Wirtshaus“ dar.

Zunächst könnte das Wirtshaus irgendwo sein, und es müsste der gesamte Suchraum danach abgesucht werden. Gibt es aber wegweisende Schilder, so müssen wir nur die, in unserem lokalen Sichtbereich entdecken und können somit das Problem einfach lösen. Wir definieren:

**Definition:** Probleme, die mit einer **lokalen Sichtweise** gelöst werden können, heißen **einfache Probleme**, Probleme zu denen keine lokale Sichtweise existiert, heißen **schwierige Probleme**.

Um diese Definition zu rechtfertigen, überlegen wir und kurz den Rechenaufwand zur Lösung eines Optimierungsproblems. Ohne jede Problemvereinfachung müssten wir jedes Element aus  $S$  mit jedem anderen Element verglei-

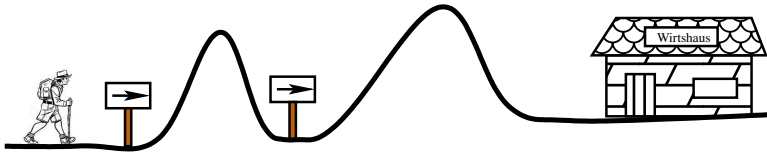


Bild 3: Die Suche nach dem Wirtshaus

chen, was bei  $N$  Elementen in  $S$  eine Größenordnung von  $\exp(N)$  Vergleichen ergibt. Kann die Anzahl der Vergleiche aber auf kleine Untermengen von  $S$  mit  $u$  Elementen eingeschränkt werden, so sind hierfür nur größenordnungsmäßig  $\exp(u)$  Vergleichsoperationen notwendig, was eine Rechenzeitvereinfachung um den Faktor  $\frac{\exp(n)}{\exp(u)}$  ergibt.

Wir erkennen somit, dass unsere Definition konzeptionell der Unterscheidung zwischen P-schwierig und NP-schwierig entspricht. Der Vorteil unserer Definition ist jedoch, dass diese den folgenden Satz beweisbar macht:

**Satz:** Es gibt schwierige Probleme.

*Beweis.* Die Unlösbarkeit des **Consensus Problems** (vgl. [3]) zeigt, dass nicht jede globale Zwangsbedingung durch lokale Zwangsbedingungen ersetzt werden kann. Daher ist es nicht immer möglich, eine globale Kennzeichnung des Gesuchten in eine lokale Kennzeichnung umzuformen, die bereits in einem lokalen Bereich erkennbar wäre.  $\square$

Dass ein Problem schwierig ist, erkennen wir daraus, dass es widersprüchliche, lokal unvereinbare Lösungsstrategien gibt. Im Bild 4 ist **Verpackungs-Problem** dargestellt. Für das Verpackungs-Problem gibt es zwei Strategien, die sich gegenseitig widersprechen und die nicht in einer einzigen lokal formulierbaren Anweisung vereinbar sind:

Strategie (1): Fülle jedes Paket möglichst vollständig!

Strategie (2): Verteile die großen Schachteln zuerst!

In analoger Weise haben wir für das **Suchproblem** die unvereinbaren Strategien:

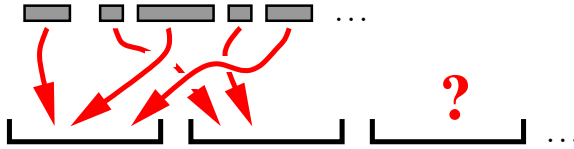


Bild 4: Verpacke die oberen Schachteln in möglichst wenigen der unten gezeigten Standard-Pakete

**(1) Exploitation:** Nutze Dein Wissen, um nicht alles absuchen zu müssen!

**(2) Exploration:** Berücksichtige alle Teile des Suchraums!

Für schwierige Probleme gilt das

**„No Free Lunch“ Theorem:** Zu schwierigen Problemen existiert nicht ein bestes Lösungsverfahren für die gesamte Problem-Klasse. Ein jeweils bestes Verfahren muss an das spezielle Problem angepasst sein.

## 4 Effiziente Suchverfahren

### 4.1 Schnelle Ein-Agenten-Suchverfahren

Um effiziente Heuristiken zu erstellen, benötigen wir Prinzipien, mittels denen wir die Suche schneller machen können. Dazu müssen wir einen Grund in der Problemstellung finden, der ein Prinzip zur Suchbeschleunigung möglich macht. Alle diese Prinzipien können wir in einer, im folgenden angegebenen, Liste zusammenstellen. Die Vollständigkeit dieser Liste ist natürlich nicht mathematisch beweisbar, sondern nur argumentativ begründbar.

- Eine Untersuchung aller uns bekannten Heuristiken ergab keine weiteren Prinzipien.
- Das Argument „Was sonst?“. Man kann überlegen, welche Möglichkeiten zur Beschleunigung einer Suche prinzipiell bestehen und daraus erkennen, dass keine weiteren Prinzipien denkbar sind.

In der Liste sind angegeben:

- der Grund, der beim Problem erfüllt sein muss, damit das Prinzip anwendbar ist,
- das **Lösungsprinzip**
- und Beispiele von Verfahren, in denen das Prinzip realisiert ist.

Um die Liste zur Konzipierung von **Ein-Agenten-Suchverfahren** zu verwenden, muss der Anwender zunächst prüfen, ob der Grund für die Anwendung eines Prinzips in seinem speziellen Such-Problem erfüllt ist. Falls dies bejaht wird, so kann er das entsprechende Prinzip mittels einer Unteroutine, die dieses realisiert, in seine Heuristik aufnehmen. Am einfachsten ist es, ein in den Beispielen angegebenes, realisiertes Verfahren zu kopieren und in die eigene Heuristik einzubauen. Eine Übersicht über alle Methoden, die in der Literatur angegeben werden, ist hier natürlich nicht möglich. Die Anzahl der Beispiele könnte beliebig ergänzt werden. Weiter ist zu beachten, dass in vielen Verfahren oft auch mehrere Prinzipien realisiert sind, d.h. eventuell auch solche, die beim vorliegenden Problem keine Verringerung der Suchzeit bewirken.

## Prinzipien schneller Lösungsverfahren:

- Es gibt eine Nachbarschaftsstruktur  $N(s)$  für  $s \in S$  auf dem Suchraum, die zum Problem passt.  
 $\Rightarrow$  Prinzip: „**Suche das Bessere nahe dem Guten!**“  
 Beispiele: Gierige Heuristik (Gradientenverfahren), Nachbarschaftssuche, variable Nachbarschaftssuche.
- Es gibt eine Bewertung der Elemente  $s \in S$  bezüglich Ihrer Lösungsqualität, ohne dass  $P(s)$  explizit berechnet werden muss.  
 $\Rightarrow$  Prinzip: „**The most promising first!**“  
 Beispiele:  $A^*$ -Algorithmus, Fred Glovers Methode: Ändere den bisherigen Lösungsweg an der Stelle der vielversprechendsten Alternative!
- Es gibt Teile des Lösungsvorschlags  $s \in S$ , die besser bestätigt sind als andere.  
 $\Rightarrow$  Prinzip: „**Lass Sicheres ungeändert und ändere Unsicheres.**“

Beispiele: Backbone-Methode, No-Goods-Methode, Ameisenverfahren, Kern Search, Auswahlverfahren der Nachbarschaftsstruktur  $N_k(s)$  bei der variablen Nachbarschaftssuche.

- Vermeidung von Doppelt-Durchsuchungen.  
 $\implies$  Prinzip: "**Ordnung hat, wer weiß, wo er erst gar nicht suchen muss!**"

Beispiele: TABU-Search, Branch & Bound-Methode.

- $\Pi : s \in S? \quad P(s) \quad \text{mit } s = (s_1, s_2, \dots, s_K) \in S_1 \times S_2 \times \dots \times S_K \equiv S$   
 ist darstellbar durch die Teilprobleme:  
 $\Pi_k : s_k \in S_k? \quad P_k(s_k) \quad \text{mit } k = 1, 2, \dots, K$   
 $\implies$  Prinzip: „**Zerlege das Problem in Teilprobleme**“

Beispiele: Gauß-Algorithmus, Crossover-Operator beim Genetischen Algorithmus.

- $\Pi : s \in S? \quad P(s)$  ist äquivalent zur Folge der Probleme:  
 $\Pi_1 : s_1 \in S_1? \quad P_1(s_1),$   
 $\Pi_2 : s_2 \in S_2? \quad P_2(s_2|s_1), \dots,$   
 $\Pi_K : s_K \in S_K? \quad P_K(s_K|s_1, s_2, \dots, s_{K-1})$   
 $\implies$  Prinzip: „**Verfolgung von Zwischenzielen!**“

Beispiel: Schrittweise Approximationsmethoden.

- $\Pi_k : s_k \in S? \quad P_k(s)$ , mit  $k = 1, \dots, K$  sei eine Folge von Problemen mit:  
 ( $\alpha$ ) Alle  $\Pi_k$  haben den selben Suchraum  $S$ .  
 ( $\beta$ )  $P_{i_1}$  ist einfach lösbar.  
 ( $\gamma$ )  $P_{i_k}$  ist ähnlich zu  $P_{i_{k+1}}$ .

$\implies$  Prinzip: „**Homotopie-Suchverfahren**“

Suche Lösung  $s_1$  zu  $\Pi_1$ ,

Suche Lösung  $s_2$  zu  $\Pi_2$  in Umgebung  $N(s_1)$ ,

Suche Lösung  $s_3$  zu  $\Pi_3$  in Umgebung  $N(s_2)$ ,

.....

Suche Lösung  $s_K$  zu  $\Pi_K$  in Umgebung  $N(s_{K-1})$ .



Tabelle 1: Koordinierungsstrategien

Intensivierung	Diversifikation
Agenten lernen vom Verhalten anderer Agenten.	Die Agenten teilen sich die Aufgabe auf, und verhalten sich möglichst unterschiedlich.

- Die Menge der möglichen Lösungsverfahren ist klein.  
 $\Rightarrow$  Prinzip: „Überführung eines Problems in sein Meta-Problem“:

$$\Pi : s \in S? \quad P(s) \quad \Longleftrightarrow$$

$\Pi_{\Pi} : \text{Lösungsverfahren} \in \text{Menge der Lösungsverfahren zu } \Pi?$   
mit Lösungsverfahren löst  $\Pi$ .

## 4.2 Koordinierungsmethoden bei Multi-Agenten-Suchverfahren

Multi-Agenten-Verfahren sind dann sinnvoll, wenn die Arbeit mehrerer Agenten koordiniert werden kann. Da bei Multi-Prozessor-Rechnern die Agenten auf die Prozessoren verteilt werden können, ergeben sich in diesem Falle große Erhöhungen der Rechengeschwindigkeit. Das **Ziel der Koordinierung** sind die beiden Strategien in Tabelle 1. Diese Strategien können, falls die entsprechenden Voraussetzungen zu den Regeln auf dem betrachteten, speziellen Problem erfüllt sind, in der Heuristik zur Koordinierung der Agenten eingesetzt werden (Tabelle 2).

## 5 Ein generelles Lösungsverfahren für die Probleme der Regelungstechnik

In diesem Kapitel wird zunächst ein Ansatz vorgestellt, mittels dem die im vorherigen Kapitel bereitgestellten Prinzipien und Regeln zur Konzipierung eines

Tabelle 2: Strategien der Koordinierung

Koordinierung von Agenten	Verfahren die diese realisieren
„Lerne vom Besten!“	Partikelschwarm-Optimierung
„Lerne beim Test jedes Agenten zur Verbesserung der gesamten Agenten-Gruppe“	Nelder-Mead-Verfahren
„Bilde das Bessere aus guten Teilen“	Genetische Algorithmen
„Lass viele Gruppen versuchen ein Problem zu lösen und wähle die effizienteste Gruppe aus“	Neuronale Netze
„Eliminiere ineffiziente Agenten und füge neue Agenten in effizienter Weise ein“	Neustart-Methode, Identifikation von vielversprechenden Gebieten in $S$ .
Kombination vieler Regeln	Fred Glovers „Multi-Agenten-Suchverfahren“

Lösungsverfahren für Suchprobleme realisiert werden können. Mit diesem Ansatz konnten Studierende Erfahrungen sammeln und regelungstechnische Probleme lösen. Dies ermöglicht einen Vergleich der verwendeten KI-Methode mit den klassischen Lösungsansätzen.

Als Ausgangspunkt wurde angenommen, dass die Studierenden praxisnah, ein einzelnes anfallendes Problem möglichst schnell lösen sollten, d.h. ohne tiefere Einarbeitungszeit in die zugehörige Mathematik. Zur Lösung wird die hier eingeführte Methode, mittels der Anleitung von Abschnitt 5.1, verfügbar gemacht. Die Aufgabe gilt als erfolgreich gelöst, wenn eine geeignete Lösung zum Problem gefunden wurde. Eine darüber hinausgehende Bewertung soll aus den folgenden Gründen nicht erfolgen:

1. In der industriellen Praxis ist es in der Regel nicht von Interesse, nachzuweisen, dass eine bereits gefundene Lösung auch einfacher auffindbar gewesen wäre.
2. Eine Bewertung des Lösungsverfahrens hängt, wegen des „No Free Lunch“-Theorems, immer auch von dem oder den betrachteten Problemen ab. Die in einer solchen Bewertungsaussage vorkommenden Begriffe: Schnelligkeit, Zugehörigkeit zur Klasse des speziellen Problems, Erfolgssicherheit des Lösungsansatzes und einige weitere, sind nur schwer fassbar. Es ist, zum Beispiel für SAT-Probleme, einer Untermenge der Klasse der Suchprobleme, bekannt, dass die besten Löser für schwierige Probleme nicht optimal für industrielle Probleme sind.
3. Da die Validierung ein eigenes Gebiet ist, das spezielle Betrachtungsweisen erfordert, soll darauf erst im Kapitel 6 eingegangen werden. In Kapitel 5 beschränken wir uns bei der Validierung auf den Nachweis der Korrektheit der erhaltenen Lösung.

## 5.1 Konzipierung eines Suchverfahrens

Gegeben ist ein Problem  $\Pi : s \in S? \quad P(s)$  und Zusatzwissen.

(1) Prüfe, ob Voraussetzungen für schnelle Suchverfahren erfüllt sind, indem die folgenden Fragen beantwortet werden:

- Gibt es Umgebungsstrukturen auf  $S$ , die zum Problem passen?
- Gibt es „vielversprechende Elemente“ in  $S$ ?
- Gibt es Kriterien, um Gebiete von  $S$  auszuschließen?
- Gibt es Wichtiges und Unwichtiges bez.  $S$  oder bez. seiner Elemente?
- Bestehen die Elemente  $s \in S$  aus mehreren Teilen?
- Sind „sichere oder gute Teile“ in Elementen  $s \in S$  erkennbar?
- Ist das Problem in mehrere Teilprobleme zerlegbar?
- Gibt es Zwischenziele, die angestrebt werden können?

- Kann das Problem aus der „Verbiegung“ eines einfachen Problems erhalten werden?
- Sollten die Agenten eine Gruppe bilden um größere Bereiche zu überblicken?

(2) Wähle einige der Methoden zu den Prinzipien aus, deren zugehörige Testfrage in (1) mit „ja“ beantwortet wurde.

(3) Entscheide, ob mehrere Agenten zur Realisierung der Methoden von (1) koordiniert werden können. Falls ein Multi-Agenten-Verfahren sinnvoll erscheint, wähle die Regeln für die Koordinierung.

(4) Realisiere die ausgewählten Methoden in einer Heuristik.

(5) Validiere die erhaltene Heuristik.

**Bemerkung:** In (1) bis (3) werden die Voraussetzungen der Prinzipien geprüft, (4) erfordert softwaretechnische Realisierungsschritte, die in (5) überprüft werden. Die Prinzipien ergeben eine „grobe“ Charakterisierung der verwendeten Verfahren. Die Effizienz der Heuristik hängt auch stark davon ab, wie gut die einzelnen Verfahren in der Heuristik realisiert sind.

## 5.2 Anwendung und Test der Methode zur Konzipierung von Heuristiken

Um dem Leser einen Eindruck zu vermitteln, welche Aufgabenstellungen durch die Studierenden gelöst wurden, werden diese im Folgenden kurz vorgestellt. Alle Aufgaben konnten innerhalb der Zeit, die für eine Masterarbeit verfügbar ist, gelöst werden. Auf eine weitere Bewertung des Lösungsverfahrens wurde, wegen der oben angegebenen Schwierigkeiten, verzichtet.

### 5.2.1 Polvorgaberegler für ein lineares Multi-Input-System

**Aufgabe:** Gesucht ist eine Matrix aus dem Suchraum  $S = \{F \in \mathbb{R}^{n \times p}\}$  mit:

$$p_A(\lambda) = \det(\lambda I - (A + BF)) \text{ hat die Nullstellen: } \lambda_1, \lambda_2 \dots \lambda_n \quad (2)$$

und der **Nebenbedingung**  $\|F\|_2$  minimal.

**Lösungsansatz:** Das Problem ist sehr komplex, da der Suchraum sehr viele lokale Optima hat. Es wurden Tests mit verschiedenen Topologien über dem Suchraum durchgeführt. Schließlich konnten für Systeme bis zur Dimension 6 mit dem **Nelder Mead-Verfahren** befriedigende Ergebnisse erzielt werden. Da das Nelder Mead-Verfahren eine ganze „Wolke von Agenten“ über dem Suchraum koordiniert, wobei die Bewertung jedes einzelnen Agenten die gesamte Wolke beeinflusst, erschien diese Methode passend zu sein, zum Problem der sehr vielen lokalen Optima.

**Lösungsumsetzung:** Mit den Bezeichnungen:

$$\sum_{k=0}^n a_k(F) \lambda^k := \det(\lambda I - (A + BF)) \text{ und } \sum_{k=0}^n \bar{a}_k \lambda^k := \prod_{k=1}^n (\lambda - \lambda_k)$$

wird der Wert  $\alpha_0 = \sum_{k=0}^{n-1} (a_k(F) - \bar{a}_k)^2$  definiert.

Die Heuristik minimiert  $\alpha_0 + \rho \|F\|_2$  wobei  $\rho$  mit ansteigender Suchzeit verkleinert wird. Die Suche gilt als erfolgreich, falls im Rahmen der Suchzeit  $\alpha_0 < \varepsilon = \frac{1}{10}$  erreicht werden konnte.

## 5.2.2 Multi-Input-Steuerungsproblem

**Aufgabe:** Gesucht wird eine zeitdiskrete Steuerungsfunktion  $(u_1(k), u_2(k))$ , die das im Bild 5 gezeigte Fahrzeug mit den in (3) angegebenen Systemgleichungen vom Zustand  $\vec{x}_0$  in den Zustand  $\vec{x}_e$  überführt.

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ x_5(k+1) \end{pmatrix} = \begin{pmatrix} x_1(k) + T_a(x_4(k) \cdot \cos(x_3(k))) \\ x_2(k) + T_a(x_4(k) \cdot \sin(x_3(k))) \\ x_3(k) + T_a \cdot x_5(k) \\ x_4(k) + T_a \cdot u_1(k) \\ x_5(k) + T_a \cdot u_2(k) \end{pmatrix} \quad (3)$$

**Lösungsansatz:** Eine Steuerfunktion besteht aus mehreren Abschnitten, daher wurde die Lösung mit dem Prinzip der Zerlegung in Teilprobleme gesucht und ein **Lösungsansatz mit einem genetischen Verfahren** gewählt.

**Lösungsumsetzung:** Da die Teilabschnitte von unterschiedlicher Wichtigkeit sind, wurden die Operatoren des Genetischen Algorithmus um Zusatzfunktionen erweitert, die weitere Prinzipien realisieren (Tabelle 3).

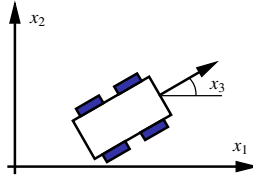


Bild 5: Zu steuerndes Fahrzeug

Wird den Routinen, welche die Prinzipien realisieren, eine gewisse Wirkungsstärke und ein gewisser Anteil an der Gesamtrechnenzeit der Heuristik zugeordnet, so kann aus dem Vergleich der Größe dieser Werte mit der verbleibenden Fehlergröße auf die Relevanz des Prinzips geschlossen werden.

### 5.2.3 Single-Input Steuerungsproblem

**Aufgabe:** Gesucht ist eine Steuerfunktion  $u(t)$ , die für das im Bild 6 dargestellten Dreitanksystem mit den Systemgleichungen (4)  $\vec{x}_0$  in  $\vec{x}_e$  überführt.

$$\begin{aligned}\dot{x}_1(t) &= -\rho\sqrt{x_1(t)} + \frac{1}{A_b}u(t) \\ \dot{x}_2(t) &= -\rho\sqrt{x_2(t)} + \rho\sqrt{x_1(t)} \\ \dot{x}_3(t) &= -\rho\sqrt{x_3(t)} + \rho\sqrt{x_2(t)}\end{aligned}\tag{4}$$

**Lösungsansatz:** Da eine Linearisierung  $\dot{\vec{x}}(t) = A\vec{x} + \vec{b}u$  in  $\vec{x} = \vec{0}$  bestimmt werden konnte und da das System, wie Gleichung (5) zeigt, stetig vom linearen System in das nichtlineare System überführt werden kann, wurde eine Lösung mit der **Homotopie-Suchmethode** ermittelt.

$$\dot{\vec{x}}(t) = \vec{x}(t) + \vec{b}u(t) + \lambda(f(\vec{x}(t), u(t)) - A\vec{x}(t) + \vec{b}u(t))\tag{5}$$

$$\text{mit } f(\vec{x}(t), u(t)) = \begin{pmatrix} -\rho\sqrt{x_1(t)} + \frac{1}{A_b}u(t) \\ -\rho\sqrt{x_2(t)} + \rho\sqrt{x_1(t)} \\ -\rho\sqrt{x_3(t)} + \rho\sqrt{x_2(t)} \end{pmatrix}\tag{6}$$

Tabelle 3: Zusatzfunktionen der Operatoren des Genetischen Algorithmus

Operatoren	Zusatzprinzipien
Crossover-Operator	Sichere Gene werden mit hoher Wahrscheinlichkeit erhalten. TABU-Liste (vermeidet zu große Ähnlichkeit der Elemente). Effiziente Agenten werden öfter berücksichtigt.
Mutation	Variable Mutationsstärke entsprechend der Effizienz des Agenten.
Agenten-Elimination	Löschen schlechter Agenten. Tendenz-Abschätzung (Agenten mit schlechter Prognose werden eliminiert.)
Neue Agenten	Diese werden in bisher unbesetzten Gebieten erzeugt.

**Lösungsumsetzung:** Zunächst wurde eine Steuerfunktion für das zugehörige lineare System bestimmt und dann die Homotopie-Suchmethode durchgeführt. Das Ergebnis für den erreichten Vektor  $\vec{x}(t_e)$  zeigt eine hinreichend gute Übereinstimmung mit dem Soll-Wert  $\vec{x}_e$ , siehe (7).

$$\vec{x}_e = \begin{pmatrix} 7,04223167 \\ 6,07614977 \\ 5,1288758 \end{pmatrix} \quad \vec{x}(t_e) = \begin{pmatrix} 7,21634809 \\ 6,13234234 \\ 5,07650415 \end{pmatrix} \quad (7)$$

### 5.3 Methoden-Vergleich

Tabelle 4 zeigt die Vor- und Nachteile der angewandten Methode gegenüber den klassischen Verfahren. Das größte offene Problem bei der Entwicklung eines generellen Problemlösers mit KI-Methoden ist die Durchführung einer

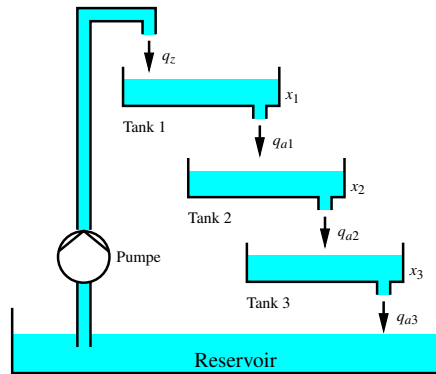


Bild 6: Nichtlineares Dreitanksystem

Validierung. Auf dieses Problem soll im Kapitel 6 genauer eingegangen werden.

## 6 Validierung

### 6.1 Anwendung des generativen Sprachmodells ChatGPT

Um das Problem der Validierung zu veranschaulichen, wird ein Vergleich des betrachteten Verfahrens mit dem Sprachmodell ChatGPT durchgeführt. ChatGPT wird zur Zeit in der KI-Gemeinde intensiv diskutiert und mancher KI-Forscher stellt sich die Frage, ob seine Forschung noch aktuell ist, oder ob er sich auf dieses neue Gebiet umstellen sollte. Auch Fachdidaktiker der Mathematik nehmen ChatGPT zur Kenntnis und stellen diesem die Frage [4]:

**Frage an ChatGPT:** Max ist 78 cm groß, Wenn er auf eine 20 cm hohe Kiste steigt, ist er genauso groß wie Klaus, Paul ist 15 cm kleiner als Klaus. Wie groß ist Paul?

Sie erhielten zunächst die falsche Antwort von ChatGPT: „Paul ist 43 cm groß.“



Tabelle 4: Vor- und Nachteile der Verfahren

Klassische Verfahren	KI-Verfahren
Wissen aus der Regelungstechnik ist notwendig.	Problemlösung ist bei geringem Wissen aus der Regelungstechnik möglich.
Jedes Problem erfordert eine spezifische Lösung.	Alle Probleme sind mit einer einzigen Lösungsanleitung lösbar.
Die entwickelten Regler erfordern geringe Rechenzeit und wenig Energie	Die entwickelten Regler erfordern hohen Rechenaufwand und viel Energie.
Der Datenaufwand ist gering	Der Datenaufwand kann hoch sein, bei einem großen Suchraum.
Die gefundene Lösung ist direkt einsetzbar und hat die gewünschten Eigenschaften.	Die gefundene Lösung erfordert eine ausgiebige <b>Validierung</b> .

Der **Zusatz zur Frage an ChatGPT**: „Lass uns Schritt für Schritt vorgehen.“  
Ergab jedoch die richtige Antwort: Paul ist 83 cm groß.

Die Frage ist nun: **Wie erhält man aus der Validierung eine Erklärung des Fehlers, die eine Verbesserung der Heuristik möglich macht?**

Als Nicht-Fachdidaktiker würde man nicht auf die Idee kommen, den obigen Zusatz zu geben, da angenommen würde, dass jemand, der überhaupt eine Antwort findet, diese immer nur schrittweise finden kann.

Das Problem von ChatGPT ist, dass es als Wissens-Hintergrund das Internet verwendet, das aus einer völlig unstrukturierten und auch häufig widersprüchlichen Informationsflut besteht. Bezüglich dieses Wissens-Chaos ist keine zielgerichtete Fehler-Grund-Suche möglich!

Dass die KI Erklärungen für ihre Ergebnisse liefern sollte, ist seit über 25 Jahren ein Thema und die Aufgabe einer „Explainable Artificial Intelligence“.

In der neueren KI-Literatur findet man viele Hinweise darauf, dass in diesem Zweig der KI Fortschritte wesentlich langsamer erhalten werden, als dies zunächst erhofft wurde. In speziellen Anwendungsfällen steht jedoch die Möglichkeit einer zielgerichteten Validierung zur Verfügung.

## 6.2 Zielgerichtete Validierung

Die Methode, mit der in unserem Fall eine Validierung durchgeführt werden kann, ist der „Means-end account of Explainable Artificial Intelligence“ [5]:

**Eine zielgerichtete Ermittlung der Nicht-Korrespondenzen zwischen dem Problem und der Heuristik.** Wie in Bild 7 dargestellt ist, können wir nicht nur Tests auf die üblichen Fehlerarten (Bias-Fehler, Overfitting und Overtraining usw.) durchführen, sondern auch gezielt die Effizienz der in der Heuristik realisierten Prinzipien und Koordinierungsregeln testen (analog, wie dies von Studierenden im Beispiel in Kapitel 5.2 durchgeführt wurde). Je nach Testergebnis kann dann die Wirkung eines Prinzips verstärkt oder abgeschwächt, bzw. ein Prinzip durch ein anderes ersetzt werden.

Da unsere Prinzipien- und Koordinierungslisten endlich sind, könnten diese (zur Zeit noch manuell ausgeführten) Tests automatisiert werden. Die Gültigkeit eines Prinzips oder einer Regel, (mit den zugehörigen Parametern,) kann als Hypothese aufgefasst werden, deren Gültigkeits-Wahrscheinlichkeit von einem Bayes-Schätzer mit den Ergebnissen aus der Validierung berechnet wird.

## 7 Schematische Darstellung eines allgemeinen Problemlösers

Wir gehen von einem Problem aus und machen uns zunächst allgemeine Überlegungen. In unserem Fall haben wir mit diesen Überlegungen einen Verstehenshorizont verfügbar gemacht, der von den Listen der Prinzipien und Koordinierungsregeln gebildet wird. Mit dem Verstehenshorizont können wir dann die Prinzipien zur Bildung eines Lösungsverfahrens zusammenstellen, dessen

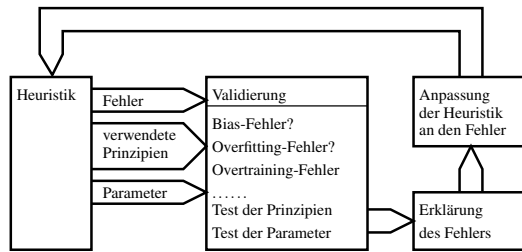


Bild 7: Validierung einer Heuristik, für den hier konzipierten Lösungsansatz.

Formalisierung einer Lösungs-Heuristik zur Berechnung einer Lösung liefert (vgl. [6] bezüglich einer analogen Vorgehensweise.)

Die Validierung der Lösung vor dem Hintergrund der verwendeten Prinzipien und Koordinierungsregeln ergibt eine Erklärung des Fehlers und damit die Möglichkeit zur Verbesserung der Heuristik. Bei allgemeinen KI-Systemen wie ChatGPT (oder auch z.B. Fahrerassistenz-Systemen) steht die Validierung zur Zeit vor dem Problem eines unerfassbaren und unstrukturierten Verstehenshorizonts, bezüglich dem der „Begriff der Erklärung“ nicht spezifizierbar ist. Der Attention-Mechanismus von ChatGPT erzeugt erst dessen Verstehenshorizont, so dass dieser dem Validierungs-Prozess nicht verfügbar ist. Die Validierungsaufgabe ist somit, im Gegensatz zu unserem Ansatz, mathematisch nicht vollständig spezifiziert. Analog zur „Strukturierten Programmierung“ bei der über die strikt mathematische Ebene der Programme eine argumentative Ebene gesetzt wird, welche die Einhaltung von Ordnungsprinzipien garantiert, wurde hier, oberhalb der Ebene der reinen Heuristiken, eine argumentative Ebene zur Konstruktion dieser Heuristiken eingefügt, um deren Auswahl und Organisation begründet durchführen zu können.

## Literatur

- [1] A. Arana und W. Stafford. „On the difficulty of discovering mathematical proofs“. Synthese, <http://doi.org/10.1007/s11229-023-04184-5>. 2023.

- [2] W. Flannery. „A revolution in physics education was forecast in 1989, why hasn't it happened? What will it take?“, Am. J. Phys., Vol.91, No4, 256-257, April 2023.
- [3] M. Lund und R.K. Belew, „No perfect two-state cellular automata for density classification exists“. Physical Review Letters 74 (25):5148-5150, 1995
- [4] S. Schorcht, L. Baumanns, N. Buchholtz, J. Huget F. Peters und M. Pohl. „Ask Smart to Get Smart: Mathematische Ausgaben generativer KI-Sprachmodelle verbessern durch gezieltes Prompt Engineering“. GMD-Mitteilungen 115, S.12-23, 2023.
- [5] O. Buchholz. „A Means-End Account of Explainable Artificial Intelligence“. Synthese, <http://doi.org/10.1007/s11229-023-04260-w>. 2023.
- [6] A. Marques Del Valle, R. Gomes Mantovani und R. Cerri. „A systematic literature review on AutoML for multi-target learning tasks“. *Artificial Intelligence Review*, <https://doi.org/10.1007/s10462-023-10569-2>. 2023.