

CI-Methoden zur Koordination und Entkoppelung komplexer Systeme

CI Methods for Coordination and Decoupling of Complex Systems

Axel Dürrbaum, Andreas Kroll, Hanns-Jakob Sommer, Universität Kassel

Zusammenfassung Der Beitrag beschreibt die Konzeptentwicklung und das Auffinden von Koordinationsregeln für komplexe verkoppelte Systeme. Am Beispiel dieser Aufgabenstellung wird die Bedeutung der Computational Intelligence für die Systemtheorie erläutert. ▶▶▶ **Sum-**

mary This paper presents the conceptual design and the derivation of coordination rules for complex coupled systems. Our solution approach demonstrates the eminent relevance of Computational Intelligence for a theory of complex systems.

Schlagwörter Komplexe Systeme, Koordination, Entkoppelung, CI-Methoden ▶▶▶ **Keywords** Complex systems, coordination, decoupling, CI-methods

1 Einleitung

Wegen des beständigen Anwachsens der Komplexität technischer Systeme wird es immer schwieriger, mit den Methoden der klassischen Regelungstechnik standardisierte Lösungsverfahren für große Systemklassen zu entwerfen. Schon bei der Entwicklung einer Regelungstheorie für nichtlineare Systeme mit den Methoden der Differentialgeometrie hat sich gezeigt, dass für jede einzelne Regelungsaufgabe bei einem nichtlinearen System, meist sehr komplexe Vorüberlegungen notwendig werden [1–3]. Für die Regelungstechnik wird es daher sinnvoll, neben der Ausrichtung auf Lösungsverfahren auch eine Theorie allgemein einsetzbarer Problemlösungssysteme zu entwerfen. Genau hierbei ist die Computational Intelligence (CI) das geeignete Hilfsmittel. Unter einem intelligenten Agenten wird im Wissenschaftsbereich der (CI) mehr verstanden, als nur ein Problemlösungsverfahren: Es wird ein adaptives, fehlertolerantes und schnelles Verfahren verlangt, das optimal seine Anforderungen erfüllt [4; 5]. Diese Anforderungen nach Flexibilität und damit Unbestimmtheit einerseits und rechentechnisch optimaler Datenrepräsentation, was absolute Bestimmtheit impliziert, andererseits, sind eigentlich unvereinbar. Daher muss zwischen der Konzeptentwicklung, die so allgemein wie möglich gehalten

werden muss, und der Realisierung, die optimal an die rechentechnischen Bedingungen angepasst sein sollte, unterschieden werden. In der vorliegenden Arbeit wird eine Entwurfsstrategie vorgestellt, die basierend auf sehr allgemeinen Lösungskonzepten, Problemlöser konzipiert, die auf einer großen Klasse von Problemen eingesetzt werden können und Parameter enthalten, mittels deren Justierung sie an ein jeweiliges spezielles Problem angepasst werden (vgl. Bild 1).

Es muss jedoch betont werden, dass der hier vorgestellte Ansatz nicht in Konkurrenz zu den klassischen Verfahren gesehen werden kann. Er liefert kein ‚bestes‘ Lösungsverfahren im klassischen Sinne und damit auch keinen vergleichbaren Grad an Problemeinsicht. Unser Ziel ist es dagegen, eine Vorgehensweise vorzustellen, die auch unter den einschränkenden Bedingungen zu einer Problemlösung führt, die üblicherweise bei der Projektabwicklung in der Industrie herrschen:

- Zu Beginn einer Projektplanungsphase sind die meisten Fragestellungen nicht vollständig mathematisch spezifiziert. Diese werden erst im Wechselspiel zwischen einer Auslotung der praktischen technischen Möglichkeiten und der theoretischen Planung fixiert.
- Kundenwünsche müssen auch im fortgeschrittenen Stadium der Projektabwicklung, z. B. nachdem dem

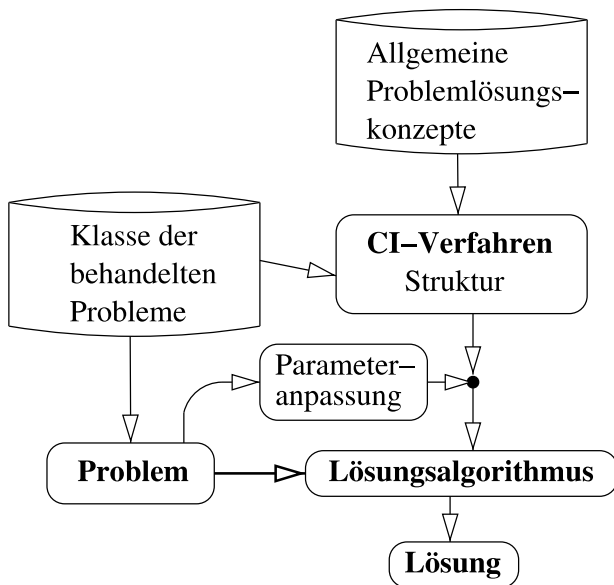


Bild 1 Konzeption eines CI-Lösungsalgorithmus.

Kunden einzelne Teile vorgestellt wurden, berücksichtigbar sein.

- Nach einer kurzen Projektplanungsphase muss die Machbarkeit des Projekts feststehen und seine Realisierungsdauer abschätzbar sein.
- Die Lösung muss vom verfügbaren Personal erbracht werden, und dieses Personal soll an der Aufgabenstellung ‚wachsen‘. D.h. es soll seine Fähigkeiten zur Bearbeitung einer möglichst breiten Vielfalt neuer Probleme verbessern. Dabei ist es ungünstig Mitarbeiter zu stark zu spezialisieren.

Kurz zusammengefasst, auch die Lösung der theoretischen Probleme eines Projekts muss in den Projektablaufplan passen und im Rahmen von diesem ‚planbar‘ sein. Theoretische Probleme müssen nicht nur lösbar sein, sondern ihre Lösung muss auch geplant werden können.

Die Fähigkeiten, die hier gefordert sind, sind sehr vielfältig und ein Intelligenzbegriff, der diesen Anforderungen entspricht, kann nicht mehr nur die Problemlösungskompetenz berücksichtigen. Es wird ein Schritt auch in der Technik notwendig, der im Bereich der Künstlichen Intelligenz mit dem Übergang zu einer Künstlichen Allgemeinen Intelligenz (Artificial General Intelligence (AGI) [6]) bereits vollzogen wurde. Allgemein intelligente technische Systeme sollen nicht mehr auf spezielle Probleme beschränkt bleiben, sie sollen ganze Klassen von Problemen bewältigen, im Voraus ihre Lösungskompetenz einschätzen können und in der Lage sein, eine Strategie anzugeben, die zur Lösung führt. Eine eingehende Untersuchung zeigt, dass bei einem ‚allgemein-intelligenten‘ System, im Gegensatz zur algorithmischen Intelligenz und analog zur natürlichen menschlichen Wissensverarbeitung, die Wissensverarbeitung zunächst mit möglichst allgemeinen, mathematisch nicht vollständig spezifizierten Begriffen abläuft. In der

vorliegenden Arbeit sollen diese Ergebnisse in praktische Verfahren umgesetzt werden, wobei wir bezüglich der theoretischen Begründung auf [7] verweisen.

Die wesentliche Idee eines konzept-orientierten Ansatzes erwächst aus der Beobachtung, dass es viel weniger prinzipielle Lösungskonzepte gibt, als konkrete Lösungsalgorithmen und deren Auffindung daher erheblich leichter planbar ist. Bei der praktischen Umsetzung dieser Idee sind zwei Fragen entscheidend:

- I Wie ist es möglich Lösungskonzepte zu einer Problemstellung zu erhalten?
- II Wie lassen sich Lösungskonzepte in Lösungsalgorithmen umsetzen?

Falls zu beiden Fragen effiziente Antworten gegeben werden können, so ergibt sich damit ein Ansatz zur Lösungsregelungstechnischer Fragen, der deren Kontext eines realen technischen Projekts berücksichtigt.

Die Vorgehensweise soll am Beispiel einer sehr allgemein formulierten Aufgabe der Systementkoppelung diskutiert werden. Im Kapitel 2 beschreiben wir in allgemeiner Form das Koordinierungsproblem bei großen technischen Systemen. In Kapitel 3 steht Frage (I) und in Kapitel 4 Frage (II) im Mittelpunkt. Die Arbeit schließt mit einer Betrachtung der Unterschiede des vorgeschlagenen Vorgehens zu klassischen Methoden, einem Ausblick auf künftige Anforderungen und einer wissenschaftlichen Einordnung von CI-Methoden.

2 Verkoppelte Systeme

Komplexe technische Systeme sind aus vielen Komponenten zusammengesetzt. Die Schwierigkeit, sie zu beherrschen und einem ungewollten Systemverhalten vorzubeugen, liegt gerade darin, dass diese Komponenten in sehr komplexer Weise miteinander verknüpft sind. Eines der gefährlichsten Phänomene, das zum totalen Zusammenbruch eines Systems führen kann, ist das sogenannte „Aufschaukeln“, bei dem jeweils eine Komponente auf eine andere reagiert und dadurch eine zyklische Reaktionskette erzeugt wird, die das System in einen immer kritischeren Zustand überführt. Ein solches Aufschaukeln führte auch zur Reaktorkatastrophe in Tschernobyl [8]. Die Überwachung dieses kritischen Phänomens führt häufig zu NP-schwierigen mathematischen Aufgaben und daher auf praktisch nicht realisierbare Anforderungen. Eine Systemüberwachung mit einem zentralen Supervisor ist dann nicht möglich und es müssen andere Kontrollmethoden gefunden werden.

Wir werden verkoppelte Systeme als Netzwerk aus Teilsystemen einführen ([9; 10]) und unser Ziel vorstellen: Das Auffinden von Verhaltensregeln für Teilsysteme, deren Einhaltung es jedem Teilsystem erlaubt, dennoch seine spezifischen Aufgaben zu erfüllen, aber dies in einer Weise, die andere nicht stört und diesen somit keine „Ausweich-“ oder „Korrekturreaktionen“ aufnötigt.

Durch diese Koordinierungsregeln können entweder gewisse Aktionen der Teilsysteme vollständig untersagt werden, oder es können den Teilsystemen Vorschriften

für die Art und Weise auferlegt werden, in der sie ihre Aktionen auszuführen haben. Die Befolgung der Koordinationsregeln kann dauerhaft gefordert werden, oder nur während sich das Gesamtsystem in gewissen Zuständen befindet.

Ein verkoppeltes System wird dargestellt durch ein Gleichungssystem aus den Systemgleichungen der K Teilsysteme:

$$\dot{\vec{X}}_k(t) = \vec{f}_k(\vec{X}(t), \vec{U}_k(t)), \quad (1)$$

oder bei zeitdiskreter Formulierung

$$\vec{X}_k(t+1) = \vec{f}_k(\vec{X}(t), \vec{U}_k(t))$$

mit $\vec{X}_k \in \mathcal{R}^{n_k}$, $\vec{X} = (\vec{X}_1^T, \dots, \vec{X}_K^T)^T$, $\vec{U}_k \in \mathcal{R}^{p_k}$, ($k = 1, \dots, K$) und Zwangsbedingungen:

$$\vec{g}_k(\vec{X}(t)) = \vec{h}_k(t) \quad (2)$$

mit $\vec{g}_k, \vec{h}_k \in \mathcal{R}^{q_k}$, ($k = 1, \dots, K$) und beliebigen Funktionen \vec{f}_k, \vec{g}_k .

Eine Zwangsbedingung der Form $g_k(\vec{X}(t)) < h_k(t)$ wird dargestellt durch $\text{sign}(g_k(\vec{X}(t)) - h_k(t)) = 0$.

Wir verbinden mit diesen Definitionen die folgenden Vorstellungen: Gleichung (1) beschreibt das dynamische Verhalten des Teilsystems k . Hängt die Funktion \vec{f}_k auch von Zustandsvektoren \vec{X}_j anderer Teilsysteme $j \neq k$ ab, so sagen wir, dass das Teilsystem k von den Variablen \vec{X}_j gestört bzw. beeinflusst wird. Wir sprechen in diesem Fall von einer *gegenseitigen Beeinflussung* der Teilsysteme. Die Zwangsbedingung (2) beschreibt die jeweiligen Anforderungen an die Teilsysteme. In jedem Teilsystem k muss \vec{U}_k so gewählt werden, dass Gleichung (2) gültig ist. (Falls es keine geeigneten \vec{U}_k gibt, so wäre das Eingreifen eines „Notsystems“ erforderlich.) Hängt die Anforderungsfunktion \vec{g}_k auch von den Zustandsvariablen \vec{X}_j eines anderen Teilsystems $j \neq k$ ab, so sagen wir, dass Teilsystem k sich mit Teilsystem j koordinieren muss. Es besteht zwischen beiden Teilsystemen ein Koordinierungsbedarf. Koordination bedeutet im Gegensatz zur Synchronisation [11] nicht, dass die Teilsysteme nach dem Abklingen des Übergangsverhaltens eine gemeinsame Trajektorie durchlaufen. Die Teilsysteme sollen durch die Koordinierung zwar von einem kritischen Systemverhalten ferngehalten werden, aber sonst so frei wie möglich in ihren Aktionsmöglichkeiten bleiben.

Bezeichnet $v(F)$ die Variablen, von denen die Funktion F abhängt und sind $\mathcal{X}_1 = \mathcal{R}^{n_1}, \dots, \mathcal{X}_K = \mathcal{R}^{n_K}$ die Zustandsräume der jeweiligen Teilsysteme, so können wir folgende Begriffe einführen:

- Das Teilsystem k heißt *unbeeinflusst*, falls: $v(\vec{f}_k) \subseteq \mathcal{X}_k$ und *unbeeinflusst* vom Teilsystem j , falls: $v(\vec{f}_k) \cap \mathcal{X}_j = \emptyset$.
- Die Zwangsbedingung $\vec{g}_k = \vec{h}_k$ heißt *streng lokal* falls: $v(\vec{g}_k) \subseteq \mathcal{X}_k$ und lokal, falls $v(\vec{g}_k)$, nur von wenigen \vec{X}_j abhängt.

- Eine Zwangsbedingung $\vec{g}_k = \vec{h}_k$ heißt *global*, falls für jede Bijektion $\sigma : \{1, \dots, K\} \rightarrow \{1, \dots, K\}$ gilt: $\vec{g}_k(\vec{X}_1, \dots, \vec{X}_K) = \vec{g}_k(\vec{X}_{\sigma(1)}, \dots, \vec{X}_{\sigma(K)})$.

Der *Abhängigkeitsgraph* eines verteilten Systems ist definiert durch $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ mit der Knotenmenge $\mathcal{V} = \{1, \dots, K\}$ und der Kantenmenge $\mathcal{E} = \{(k, j) \in \mathcal{V} \times \mathcal{V} \mid \mathcal{X}_k \cap v(\vec{g}_j) \neq \emptyset\}$.

Es gibt sehr viele Systeme, die mit den eingeführten Definitionen für verkoppelte Systeme beschrieben werden können. Wir denken hier in erster Linie an eine Anlage, die aus verschiedenen Stationen aufgebaut ist, aber der eingeführte Formalismus erfasst auch Multiagentensysteme zur Überwachung eines Gebiets, Verkehrssysteme und die Teilnehmer eines Marktes.

Um ihre Aufgaben erfüllen zu können, müssen die Aktionen der Teilsysteme oder Agenten k miteinander koordiniert werden. Dafür gibt es folgende Möglichkeiten:

- (I) Den *omnipotenten Koordinator*: Alle Agenten erhalten strikte Anweisungen bezüglich der Wahl ihrer Steuerfunktion \vec{U}_k von einem, die gesamte Anlage erfassenden und überwachenden, Koordinator.
- (II) Agenten einer lokalen Nachbarschaftsgruppe, z. B. die Agenten $N_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E} \text{ oder } (j, i) \in \mathcal{E}\}$, kommunizieren ihre jeweiligen Daten und führen ihre eigenen Aktionen in Absprache oder unter Beachtung der benachbarten Agenten aus.
- (III) Die Agenten bilden gemeinsam ein Wissen, auf das alle zugreifen können, mittels dem ihre Aktionen unter Beachtung zusätzlicher Regeln, (die wir hier auch *koordinierende Zwangsbedingungen* nennen,) so beschränkt werden, dass ihre Wahl, bezüglich der noch verbleibenden Freiheiten, für andere Systemteile keinen beschränkenden Einfluss mehr hat.

Die Koordinationsmethode I würde natürlich das beste Ergebnis erzielen, aber sie erfordert einen erheblichen Datenfluss im System und darüber hinaus stellt die Optimierungsaufgabe an den Koordinator im Allgemeinen ein nicht behandelbares (NP-schwieriges) Rechenproblem dar.

Alle Algorithmen der Methoden II und III sollen mit polynomialer Zeit ausführbar sein, d. h. ihre Rechenzeit ist durch eine Schranke $p(\sum n_k)$, $k = 1, \dots, K$ beschränkt, wobei p ein fest gewähltes Polynom ist. Während die Methode II bezüglich der Topologie des Abhängigkeitsgraphen nur lokale Abhängigkeiten berücksichtigt, hängt Methode III von der global im System verfügbaren Information ab. Im Unterschied zu Methode I muss dieses „globale Wissen“ bei Methode III jedoch in einem zentralen Speicher so aufbereitet sein, dass es von den Agenten jeweils polynomial-schnell ausgewertet werden kann. Die Methode III entspricht dem von Max Weber in den Sozialwissenschaften eingeführten *Idealverhalten* [12]. Sie soll hier insbesondere untersucht werden. Die Bestimmung der „Wohlverhaltensinformation“ und der zugehörigen koordinierenden Zwangsbedingungen ist das zentrale Problem des vorliegenden Aufsatzes. Eine

spezielle Form der koordinierenden Zwangsbedingungen besteht darin, Regeln an die Teilsysteme für eine *angepasste Aktionsausführung* vorzugeben. Agent k muss dann seine Steuerfunktion $\vec{U}_k(t)$ mit $t \in [T_0, T_{\text{Ende}}]$ so wählen, dass nach deren Eingabe die Zustandswerte für Zeiten $t > T_{\text{Ende}}$ in allen beeinflussten Teilsystemen j die selben Werte haben, die sie auch bei der Wahl $\vec{U}_k(t) \equiv 0$ angenommen hätten. Eine solche angepasste Aktionsausführung ist auch die Bedingung, die häufig an ein technisches System in einem ökologischen Umfeld gestellt ist. Nach dem Aktionsablauf, der vom technischen System verursacht wird, soll das ökologische Umfeld wieder in dem Zustand sein, den es auch angenommen hätte, wenn es das technische System nicht geben würde. An einem einfachen Beispiel wird diese Methode der angepassten Aktionsausführung im Abschnitt 4 genauer erläutert.

3 Lösungsansatz zum Problem der Systemkoordination

Wir betrachten in diesem Abschnitt Systeme unbeeinflusstester Teile, die über Zwangsbedingungen verkoppelt sind. Die folgenden Aussagen zeigen, dass die Koordinierungsmethoden I, II und III nicht gegeneinander austauschbar sind.

Satz 1. (vgl. [13])

- (1) *Es gibt verkoppelte Systeme, die mit Methode I entkoppelbar sind, bei denen jedoch selbst eine gemeinsame Anwendung der Methoden II und III keine Entkopplung ermöglicht.*
- (2) *Es gibt verkoppelte Systeme, zu denen eine P-schnell (mit polynomialer Zeit) berechenbare Entkopplung mittels der Methode III möglich ist, aber nicht mit Methode II.*
- (3) *Einige Klassen verkoppelter Systeme mit P-schnell berechenbaren Entkopplungsalgorithmen können durch algebraische Bedingungen charakterisiert werden.*
- (4) *Die P-schnell berechenbare Entkoppelbarkeit verkoppelter Systeme wird verhindert durch Synchronisationseffekte, die sich über weite Gebiete des Abhängigkeitsgraphen \mathcal{V} ausbreiten. Diese Effekte sind emergent, d. h. im Allgemeinen nicht mit P-schnell auswertbaren Charakterisierungen aus der Systemdefinition vorher-sagbar [14].*

Aussage (4) zeigt, dass die Schwierigkeiten bei der Entkopplung verkoppelter Systeme nicht bereits aus der Systemdefinition heraus mit P-schnellen Algorithmen erkannt werden können. Es ist daher sicher nicht möglich, dieses Problem allgemein in algorithmischer Weise zu lösen. Allgemeine Lösungsansätze zum Entkopplungsproblem sollten daher notwendigerweise heuristische Verfahren und CI-Methoden mit einbeziehen.

Die allgemeinen Konzepte zur Lösung des Entkopplungsproblems können mit den folgenden Schritten gefunden werden [13]:

Orientierung an einem paradigmatischen Beispiel: Wegen der Analogie zwischen Entkopplungsproblemen

und Zwangsbedingungserfüllungs-Problemen (SAT-Problemen) [15] kann ein Satz von Schäfer verwendet werden, der eine Kennzeichnung der P-schnell lösbaren SAT-Instanzen und der zugehörigen Lösungsmethoden liefert.

Verallgemeinerung: Indem diese Lösungsmethoden fuzzi-fiziert werden, also „all-Aussagen“ durch „fast-immer-Aussagen“ und „nicht-Aussagen“ durch „selten-Aussagen“ ersetzt werden, erhält man nach einigen Umformungen der Aussagensysteme das folgende Regelsystem:

Regeln zur Bildung koordinierender Zusatzbedingungen:

Regel 1. *Vermeide die Erzeugung von „Behinderungen“*

Unter einer „Behinderung“ durch Agent k verstehen wir die Erzeugung von Bedingungen, die wegen seines Zustandswertes \vec{X}_k durch $\vec{g}_i(\dots \vec{X}_k, \dots \vec{X}_i, \dots) = \vec{h}_i$ für andere Agenten i entstehen, die die „üblichen“ Aktionen, die in Teilsystem i ausgeführt werden, nun unmöglich machen. Regel 1 verpflichtet die Agenten auf Grund eines allgemeinen Systemwissens und erfordert daher noch keine Koordination.

Regel 2. *Koordination von Agentengruppen:*

- *Finde Konfigurationen von Zwangsbedingungen $\vec{g}_i = \vec{h}_i$ mit $i \in \mathcal{V}_{\text{koord}} \subset \mathcal{V}$ und $|\mathcal{V}_{\text{koord}}| \ll K$, die schwierig gemeinsam zu erfüllen sind.*
($|M|$ beschreibt die Anzahl der Elemente von M)
- *Die Agenten der Gruppe $\mathcal{V}_{\text{koord}}$ koordinieren sich unter Berücksichtigung der Zwangsbedingungen $\{\vec{g}_i = \vec{h}_i \mid i \in \mathcal{V}_{\text{koord}}\}$ mittels der Methode I.*

Regel 3. *Jeder Agent soll sich „gut verhalten“, d. h. seine Aktionen so wählen, dass die Freiheitsgrade der anderen Agenten zur Erfüllung ihrer Aufgaben möglichst groß bleiben.*

Diese Strategie ist dann einfach realisierbar, wenn Charakterisierungen des Systems $C(\vec{X}_1, \dots, \vec{X}_k, \dots, \vec{X}_K)$ bekannt sind, die eine besonders gute bzw. schlechte Effizienz des Systems kennzeichnen. Dabei können diese Charakterisierungen, im Gegensatz zu Regel 2, auch globale Begriffe enthalten, die das gesamte System berücksichtigen. Jeder Agent soll sich an diesen Charakterisierungen orientieren und solche Aktionen wählen, durch die die Effizienz des Systems erhöht wird.

Regel 4. *Den Agenten werden, relativ zu den Freiheitsgraden, die ihnen ihre Anforderungen lassen, Prioritäten zugeordnet. Hat Agent k weniger Freiheiten, seine Anforderungen $\vec{g}_k = \vec{h}_k$ zu erfüllen, als Agent i die seinen $\vec{g}_i = \vec{h}_i$, so erhält Agent k gegenüber Agent i die höhere Priorität. Die Agenten führen dann ihre Aktionen in der Reihenfolge ihrer Prioritäten aus und teilen diese den anderen mit.*

Regel 5. *Falls eine Aktionsauswahl aller Agenten nicht bezüglich der Regeln 1 bis 4 erfolgreich durchführbar ist, wird eine andere Aktionsauswahl versucht, wobei nicht ef-*

fektive Aktionen in einer TABU-Liste gespeichert werden, um Schleifen beim Suchprozess zu vermeiden.

Regel 6. Portfolio-Algorithmen werden erzeugt, indem von einer Menge möglicher Algorithmen jeder durch gewisse Aussagen charakterisiert wird, die Systeme kennzeichnen, auf denen er besonders effektiv ist. Für ein neues Problem wird dann mittels dieser Charakterisierungen der geeignetste Algorithmus des Portfolios ausgewählt.

Die Regeln dienen als heuristischer Leitfaden, um mittels der Methoden II und III effektiv koordinierende Zusatzbedingungen zu finden. Das folgende Beispiel soll ihre Anwendung verdeutlichen:

Beispiel: Koordination von Überwachungsagenten

Das Koordinierungsproblem für Multiagentensysteme (MAS) besteht aus zwei Teilen [16]:

- dem Auffinden von Formationen, in denen das MAS effektiv ist (formation control), und
- der Einigung der Agenten auf eine Formation (consensus problem).

Auch das Koordinierungsproblem für ein MAS wird durch die Gleichungen (1) und (2) dargestellt. Gleichung (1) beschreibt dabei Zustandsänderungen, die jeder Agent durch seine Aktionen \tilde{U}_i bewirken kann, um die an ihn gestellten Forderungen, die durch Gleichung (2) spezifiziert sind, zu erfüllen. Durch Koordinierungsregeln soll ein Konsens erzwungen werden, der außerdem noch die Effektivität des MAS erhöht. Dies wird an folgender **Aufgabenstellung** erläutert [17]:

- K Agenten überwachen ein Gebiet, um dort bei Notfällen zur Stelle zu sein. Die Notfälle ereignen sich rein zufallsverteilt, ohne dass weitere Information verfügbar wäre. Ein Agent kann einen Notfall erkennen, wenn er diesem bis auf den Abstand ϵ nahe kommt.

Lösung: Eine **erste Verhaltensregel** wäre: *Jeder Agent führt einen Random Walk aus und greift ein, sobald er einen Notfall entdeckt.*

Die Effektivität dieses Verhaltens wird durch folgende **Charakterisierung** bewertet: *Befinden sich alle Agenten zusammen in einem kleinen Teil des Gebiets, so ist ihre Effektivität gering.*

Regel 3 ergibt dann die mit Methode III realisierbare **Strategie**:

Es werden Beobachtungspunkte \tilde{x}_l , ($l = 1, \dots, L$) im Gebiet des MAS verteilt, zu denen jeder Agent seinen jeweiligen Abstand bestimmt. In einem Wissenspeicher, der allen zugänglich ist, bilden die Agenten die Werte:

$$d(\tilde{x}_l) = \sum_{k=1}^K \exp(-\|\tilde{X}_k - \tilde{x}_l\|^2)$$

wobei \tilde{X}_k die Position des Agenten k beschreibt. Die Werte $d(\tilde{x}_l)$ sind ein Maß für Dichte der Agenten in der Nähe der Beobachtungspunkte.

Mittels dieser Information zur *Koordination eines Idealverhaltens* kann jeder Agent sein Verhalten so

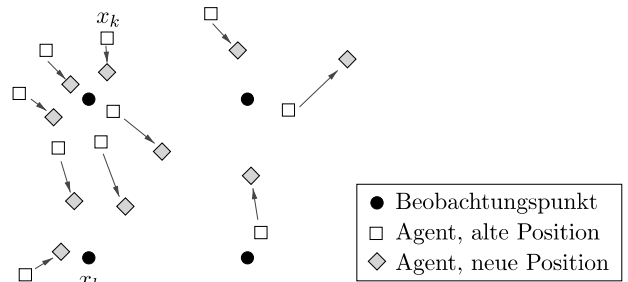


Bild 2 Wanderung der Agenten.

anpassen, dass er Beobachtungspunkte mit großer Agentendichte verlangsamt anfährt, bzw. beschleunigt verlässt und bei Beobachtungspunkten mit unterdurchschnittlicher Agentendichte sich umgekehrt verhält (vgl. Bild 2).

Lemma. Sei d_0 der Erwartungswert von $d(\tilde{x}_l)$, der sich für im Gebiet gleichverteilte Agenten ergibt, dann ist

$$\mathcal{V}(\tilde{X}_1, \dots, \tilde{X}_k, \dots, \tilde{X}_K) = \sum_{l=1}^L (d(\tilde{x}_l) - d_0)^2$$

ein Maß für die Ungleichverteilung der Agenten. Für jeden Agenten ist die Funktion $\frac{\partial \mathcal{V}}{\partial \tilde{X}_k} = \sum_{l=1}^L 2(d(\tilde{x}_l) - d_0) \exp(-\|\tilde{X}_k - \tilde{x}_l\|^2) \frac{\partial(-\|\tilde{X}_k - \tilde{x}_l\|^2)}{\partial \tilde{X}_k}$ aus den ihm bekannten Größen $d(\tilde{x}_l)$ und \tilde{X}_k bestimmbar.

Sei $\vec{V}_k = \frac{d\tilde{X}_k}{dt}$ die Geschwindigkeit des Agenten k , ρ ein Zufallswert für die Beschleunigung im Random Walk und λ geeignet gewählt, dann wird durch die *Regel*:

$$\vec{V}_k(t + \Delta t) = \vec{V}_k(t) + \Delta t \cdot \left(\rho - \lambda \frac{\partial \mathcal{V}_k(\tilde{X}_k)}{\partial \tilde{X}_k} \right)$$

eine Geschwindigkeitsänderung der Agenten erreicht, durch die sich die erwartete Zeit verringert, die vergeht bis ein Agent an einem neuen Notfall eintrifft.

Beweis. Durch die Regel wird eine erhöhte Gleichverteilung der Agenten über das Gebiet bewirkt. Dadurch verringert sich die Wahrscheinlichkeit dafür, dass ein Notfall in einer größeren Region ohne Agenten auftritt. Lange Wartezeiten werden unwahrscheinlich. Die Stärke dieses Effekts hängt von den Systemparametern ab. Eine eingehende neuere Untersuchung des Überwachungsproblems wird in [18] gegeben.

4 Angepasste Aktions-Ausführung

Im diesem Abschnitt diskutieren wir die Entkopplung von Teilsystemen, die über wechselseitige Einflüsse verkoppelt sind. Dabei steht die Art und Weise im Vordergrund, wie dann aus vorliegenden Konzepten mittels den Schritten der Konzeptauswahl und deren algorithmischer Realisierung ein konkretes Lösungsverfahren entsteht. Diese Schritte erläutern wir am Problem der störungsarmen bzw. tolerierbaren Aktionsausführung, d. h. der Erfüllung der Anforderung an alle Teilsysteme, ihre

jeweiligen Aktionen so auszuführen, dass deren Störeinflüsse auf andere Systemteile gering bleiben.

4.1 Aufgabenstellung zur störungsarmen Aktionsausführung

Wir betrachten verkoppelte Systeme, die ausschließlich mit Gleichung (1) definiert sind, für die also keine Zusatzbedingungen (2) gefordert werden. Eine erste Idee, die gegenseitige Beeinflussung der Systemteile zu unterdrücken, wäre, den Einfluss von System k auf System i als Störung zu interpretieren und die Methode der Störunterdrückung anzuwenden [2]. Eine andere, aus der Physik bekannte Methode [19], besteht darin, die Dimensionalität der Verkoppelung so weit als möglich zu reduzieren. Für das System:

$$\begin{aligned}\vec{X}_1(t+1) &= \vec{f}_1(\vec{X}_1(t), \vec{X}_2(t), \vec{U}_1) \\ \vec{X}_2(t+1) &= \vec{f}_2(\vec{X}_1(t), \vec{X}_2(t), \vec{U}_2)\end{aligned}$$

würde eine Funktion $\phi: \mathcal{X}_1 \rightarrow \mathcal{V}_1 \subset \mathcal{X}_1$ gesucht mit $\dim(\mathcal{V}_1) < \dim(\mathcal{X}_1)$, s. d. gilt:

$$\vec{f}_2(\phi^{-1}(\phi(\vec{X}_1(t))), \vec{X}_2(t), \vec{U}_2) = \vec{f}_2(\vec{X}_1(t), \vec{X}_2(t), \vec{U}_2).$$

Es müsste dann nur die Information $\phi(\vec{X}_1)$ zum Teilsystem 2 übertragen werden (und analog bzgl. \vec{X}_2).

Beide Methoden sind dann geeignet, wenn die Systemgleichungen festgelegt sind und wenn eine vollständige Entkopplung möglich ist und wirklich benötigt wird. Die Ansätze machen komplizierte differentialgeometrische Überlegungen erforderlich, die für jedes spezielle System neu durchgeführt werden müssten. Durch Verwendung von CI-Verfahren kann dagegen eine Bewertung der gegenseitigen Störung erst während dem Lösungsprozess festgelegt werden.

Um die Unterdrückung der gegenseitigen Beeinflussung für erheblich größere Klassen verkoppelter Systeme möglich zu machen, führen wir zunächst eine wesentlich schwächere Bedeutung für den Begriff der gegenseitigen Beeinflussung ein und zeigen dann, wie Steuerfolgen, die dieser neuen Definition entsprechen, mit den Methoden der CI in sehr allgemeiner Weise, d. h. für große Funktionsklassen \vec{f}_k , berechnet werden können.

Definition 1 (Tolerierbare Steueraktionen). Sei $\mathcal{X}_{0i} \in \mathcal{X}_i$ eine Menge von Zustandswerten in denen sich das Teilsystem i normalerweise befindet und seien \vec{U}_{0i} die dann üblichen Steuerfolgen. Eine Aktion \vec{U}_k , d. h. eine Folge von Steuerwerten $\{\vec{U}_k(T_0), \dots, \vec{U}_k(T_0 + \Delta T)\}$ für Teilsystem k heißt $(\epsilon, \Delta T)$ tolerierbar, falls für alle Teilsysteme i mit $i \neq k$ gilt, dass der Zustandswert $\vec{X}_i(T_0 + \Delta T)$, der im Teilsystem i nach der Zeit ΔT erhalten wird, wenn System k die Aktion \vec{U}_k ausgeführt, sich vom Zustand $\vec{X}_{i0}(T_0 + \Delta T)$ der entstanden wäre, wenn k sich normal verhalten, also \vec{U}_{0k} ausgeführt hätte, um nicht mehr als ϵ betragsmäßig unterscheidet, d. h. wenn gilt:

$$\|\vec{X}_i(T_0 + \Delta T) - \vec{X}_{i0}(T_0 + \Delta T)\| < \epsilon$$

Unter einer vollständigen Menge $\mathcal{A}_{k,\epsilon}$ tolerierbarer Steueraktionen für Teilsystem k wird eine Menge von $(\epsilon, \Delta T)$ tolerierbaren Steuerfolgen \vec{U}_k verstanden, mittels denen k alle seine Anforderungen erfüllen kann.

Das Auffinden einer $(\epsilon, \Delta T)$ tolerierbaren Steuerfunktion für Teilsystem k erfordert die Lösung der folgenden

Aufgabe: Sei $\vec{U}_k = \{\vec{U}_k(T_0), \dots, \vec{U}_k(T_0 + \Delta T)\}$ eine Folge von Steuerwerten für Teilsystem k und \vec{U}_{0k} beschreibe die normalen Steuerwerte im Teilsystem k , $\mathcal{X}_0 = \{(\vec{X}_{01}, \dots, \vec{X}_{0k}, \dots, \vec{X}_{0K})^T \mid \vec{X}_{0i} \in \mathcal{X}_{0i} \text{ für } i \neq k, \text{ und } \vec{X}_{0k} \text{ sei der Anfangszustand im Teilsystem } k\}$, dann wird nach ΔT Zeitschritten der folgende Zustand des Gesamtsystems erreicht: $\vec{f}_k(\vec{X}_0, \vec{U}_k) = \vec{f}_k(\dots \vec{f}_k(\vec{f}_k(\vec{X}_0(t_0), \vec{U}_k(t_0)), \vec{U}_k(t_0 + 1)) \dots \vec{U}_k(t_0 + \Delta T))$ mit $\vec{X}_0 \in \mathcal{X}_0$. Finde \vec{U}_k , so dass für den gewünschten Zustand $\vec{X}_{k,\text{soll}}$ gilt: $\vec{f}_k(\vec{X}_0, \vec{U}_k) = \vec{X}_{k,\text{soll}}$, und \vec{U}_k ist $(\epsilon, \Delta T)$ tolerierbar für alle anderen Teilsysteme. Genauer: bezeichnet \vec{X}_k die Folge der Zustandswerte, während des Steuervorgangs im Teilsystem k und \vec{X}_{k0} diejenigen Zustandswerte, die beim Normalverhalten in Teilsystem k aufgetreten wären, so muss gelten: $\vec{f}_i(\vec{X}_0, \vec{X}_k, \vec{U}_i) = \vec{f}_i(\vec{X}_0, \vec{X}_{k0}, \vec{U}_{i0})$ für alle üblichen Steuerfolgen \vec{U}_{i0} in den Teilsystemen $i \neq k$.

Für jedes Teilsystem k muss eine Menge von $(\epsilon, \Delta T)$ tolerierbaren Aktionen \mathcal{A}_k gefunden werden, mit denen alle seine Anforderungen erfüllt werden können. Die Modalitäten dieser Aufgabe hängen natürlich von der Schranke ϵ bzw. von ΔT ab. In jedem Teilsystem werden dann alle Steueraufgaben mittels der Hintereinanderausführung von Aktionen aus \mathcal{A}_k gelöst. Kann eine Aufgabe nicht nur mit solchen tolerierbaren Aktionen gelöst werden, so müssen sich gegenseitig nicht tolerabel beeinflussende Systemteile gegenseitig absprechen.

Die Liste der Aktionen wird off-line erstellt, so dass während des Betriebs ein schneller Zugriff möglich ist.

4.2 Charakterisierung verschiedener CI-Verfahren für den Entwurf eines Suchalgorithmus zum Auffinden tolerierbarer Steueraktionen

Da wir zum Problem von Abschnitt 4 über keine speziell entworfenen Lösungskonzepte verfügen, suchen wir diese unter den allgemein bekannten CI-Verfahren. Dabei orientieren wir uns am Bild 1. Jedem CI-Verfahren entsprechen einige grundlegende Konzepte, die mittels diesem algorithmisch realisiert werden. Ein CI-Verfahren „passt“ zu einem Problem, wenn seine Konzepte mit den Begriffen und Voraussetzungen des Problems korrespondieren. Diese Auswahl soll am Beispiel der Aufgabe aus Abschnitt 4.1 genauer dargestellt werden. Wir charakterisieren einige CI-Verfahren und diskutieren diese Charakterisierungen qualitativ im Vergleich zu den Bedingungen beim Entkopplungsproblem.

Evolutionäre Verfahren basieren wesentlich auf dem Austausch von Gen-Gruppen zwischen den Eltern (crossover). Dieses Konzept ist effektiv, wenn diese Gen-Gruppen oder Schemata „für sich“ als vorteilhaft oder

unvorteilhaft bewertbar sind (vgl. Goldbergs Schema-Theorem). Das Verfahren realisiert also wesentlich das Konzept: „Bilde eine Lösung aus effizienten Teilen“. Zum Problem des Auffindens von Steuerfunktionen passt dieses Konzept nicht, da eine Folge von Steuerwerten nur als ganzes und nicht mittels einer losgelösten Bewertung ihrer Teile beurteilt werden kann.

Computed Swarm-Verfahren verwenden das Konzept: „Orientiere Dich am Besten“ ([20]). Da sehr viele verschiedene Steuerfolgen ähnlich gute Ergebnissen ergeben können, würde dieses Konzept durch Mischung die Lösungsfindung, bei alternativen Startpunkten für das Suchverfahren, in falsche Richtungen führen.

Neuronale Netze realisieren beim überwachten Lernen das Konzept: „Lass viele Gruppen nach einer Lösung suchen und wähle die beste aus“. Die Gruppen werden dabei durch einige Neuronen realisiert, deren Verbindungsgewichte durch das Lernverfahren (Back-Propagation-Algorithmus) so eingestellt werden, dass die Gruppe zusammen eine Lösung erzeugt. Verbindungen anderer Gruppen von Neuronen, die zur Lösung nichts beitragen, werden dagegen vom Lernverfahren bei der Anpassung des Netzes an die vorgelegte Aufgabe nur wenig geändert. Der Nachteil dieses Lösungsverfahrens, für die vorliegende Aufgabe besteht darin, dass schon die Steuerfolgen relativ viel Speicherplatz benötigen. Um ein Überangebot von problemlösenden Neuronengruppen zu haben, müsste die verdeckte Neuronenschicht sehr groß gewählt werden. Außerdem hätte ein Neuronales Netz Schwierigkeiten bei vielen Steuerfolgen mit gleicher oder ähnlicher Bewertung.

Tabu-Suchverfahren koppeln einen „Gierigen Suchalgorithmus“ mit einer Tabuliste, die ein zirkuläres Suchen und eine Stagnation in lokalen Optima verhindert [21]. Ein solches Verfahren eignet sich dann, wenn, wie im vorliegenden Fall, die gesuchte Lösung keine zusätzliche Strukturierung ermöglicht, die bereits vor der eigentlichen Lösungssuche festgelegt werden könnte.

4.3 Neue Diversifikationsgetriebene

Tabu-Such-Heuristik mit Abkühlung

Einer Idee von F. Glover et al. folgend [22; 23], verbinden wir die Tabu-Methode mit einer Neustartroutine, die dann aufgerufen wird, wenn die Umgebung in der der Gierige Suchalgorithmus eine Lösung sucht, keine Verbesserung enthält oder vollständig Teil der Tabuliste ist. Die Neustartroutine sucht dann einen neuen Startwert für das Suchverfahren, der möglichst in einem bisher noch wenig durchsuchten Gebiet des Lösungsraums liegt. Mittels einer Abkühlungsroutine wird es möglich, die Präzision der Suchroutine in Abhängigkeit von der erreichten Suchgenauigkeit zu verbessern.

Vom Algorithmus genutzte Listen:

Lscaling: Liste der möglichen skalierten Steuerwerte,

FlipFreq(i): Anzahl der Rechenschritte, in denen die Komponente $U_i(t)$ geändert wurde,

EliteSol = $[\check{U}_1, \dots, \check{U}_p]$: Liste der bisher besten Lösungen,
Distab = $[\check{U}_1, \dots, \check{U}_M]$: Liste der M bisherigen Startwerte mit größtem gegenseitigen Abstand. Falls mehr als M Neustarts erforderlich sind, wird der Startwert \check{U}_m aus *Distab* entfernt, für den $\max_{m_1 \neq m_2} |\check{U}_{m_1} - \check{U}_{m_2}|$ maximal bleibt.

newLscaling: Darstellungsbereich auf den umskaliert werden soll.

PROZEDUR DIVERSIFICATION-DRIVEN-TABU-SEARCH WITH ANNEALING:

INITIALISIERUNG: Bestimme Zufallswert $\check{U}_1 \in Lscaling$ und trage diesen in *Distab* ein.

WHILE (STOP-Bedingung nicht erfüllt):

CALL TABU-SEARCH

IF (TABU-SEARCH liefert keine Verbesserung)

THEN CALL DIVERSIFICATION

ENDIF

IF (*EliteSol* \subset *newLscaling*) AND rescaling-condition

THEN CALL ANNEALING und justiere alle

Elemente von *EliteSol* mit TABU-SEARCH

ENDIF

ENDWHILE

Ausgabe der besten Lösung aus *EliteSol*.

ROUTINE TABU-SEARCH:

Die Routine erzeugt aus einer gegebenen Steuerfolge mittels einer endlichen Menge von Änderungsoperatoren, falls deren Anwendung nicht wegen der Tabuliste ausgeschlossen wird, eine Menge neuer Steuerfolgen. Aus diesen Steuerfolgen wird die Steuerfolge \check{U}_k ausgewählt, für die der folgende Ausdruck den minimalen Wert liefert:

$$\sum_{\check{X}_0 \in \mathcal{X}_0} \alpha \| \tilde{f}_k(\check{X}_0, \check{U}_k) - \check{X}_{k, \text{soll}} \| + (1 - \alpha) \| \tilde{f}_i(\check{X}_0, \check{X}_k, \check{U}_i) - \tilde{f}_i(\check{X}_0, \check{X}_{k0}, \check{U}_{i0}) \| \quad (3)$$

($\alpha \in \mathbb{R}_+$ ist ein geeignet gewählter Parameter).

ROUTINE DIVERSIFICATION:

Die Routine beachtet folgende **Regeln**:

- (i) Ändere keine stabilen Zuweisungen.
- (ii) Ändere nicht beständig die selben Variablen.
- (iii) Wähle den neuen Startwert so weit entfernt als möglich von früheren Startwerten.

Die Routine besteht aus den Schritten:

1. Weise den Variablen $u_{ik}(t)$ (den Komponenten der Vektoren $\check{U}_k(t)$) Wahrscheinlichkeiten p_i so zu, dass gemäß den Regeln (i) und (ii) p_i klein ist für sehr kleine oder große Werte von *FlipFreq(i)* und für mittlere Werte von *FlipFreq(i)* größer gewählt wird. Es ergibt sich dadurch eine Wahrscheinlichkeitsverteilung $\{p_1, \dots, p_N\}$ über den Komponenten $u_{ik}(t)$.
2. Wähle eine Anzahl von m Variablen $u_{ik}(t) \in Lscaling$ aus, bezüglich der Wahrscheinlichkeitsverteilung $\{p_1, \dots, p_N\}$.
3. Wähle Zufallswerte für die ausgewählten Variablen und erzeuge R verschiedene Steuerfolgen.

4. Wähle aus den R gebildeten Steuerfolgen diejenige aus, die zu den Steuerfolgen in $Distab$ den größten Abstand d_{\max} hat. (\dot{U}, d_{\max}) sei das Ergebnis.
5. Trage (\dot{U}, d_{\max}) in einen freien Platz von $Distab$ ein, bzw. ersetze $(\dot{U}_{\min}, d_{\min}) = \arg \min_{Distab} d$ durch (\dot{U}, d_{\max}) in $Distab$.
6. RETURN(\dot{U}).

ROUTINE ANNEALING:

In der Anfangsphase der Berechnung werden Änderungen der Variablen $u_{ik}(t)$ mit einer zu $|t - \tau|$ umgekehrt proportionalen Größe auch für $u_{ik}(\tau)$ vorgenommen. Die Änderungsgröße und die Nachbarschaften, in denen diese Übertragungen der Änderungen vorgenommen werden, reduzieren sich während des Suchprozesses. Falls der Suchvorgang stagniert, wird eine Neuskalierung der Variablen $u_{ik}(t)$ bezüglich einer engeren Werteskala mit beschränkterem Wertebereich durchgeführt. D. h. alle Werte werden in einem verkleinerten Darstellungsbereich (*EliteSol*) in feinerer Skalierung dargestellt.

Parameteroptimierung. Zunächst werden Lösungen mittels kurzzeitigen Suchen für verschiedene Parametereinstellungen erzeugt. Mit den Parametern, die in diesen Voruntersuchungen die besten Lösungen ergaben, werden dann die Langzeit-Suchen durchgeführt. Dieser Schritt soll in späteren Ausbaustufen des Verfahrens automatisiert werden. Das Programm ist so konzipiert, dass die Funktionen \vec{f}_k als Unterrouтины realisiert sind, die beliebig ausgetauscht werden können.

4.4 Fallstudie: Portalkran-Steuerung

Ein Kran werde von zwei Bedienern gesteuert (vgl. Bild 3). Dem Kranfahrer, der die Kraft F auf die Katze bestimmt und dem Entlader, der das Moment am Lastarm steuert. Der Kranfahrer muss darauf achten, dass der Lastarm nicht zu weit ausschwingt und der Entlader sollte durch seine Aktion keine Änderung der Position der Katze bewirken.

Zur Beschreibung des Systems werden folgende Bezeichnungen eingeführt:

- M Masse der Katze in kg
- m Masse des Lastarms in kg
- Θ Trägheitsmoment des Lastarms in kg m^2
- L halbe Länge des Lastarms in m
- $s(t)$ Position der Katze in m
- $\psi(t)$ Winkel des Lastarms bzgl. der Vertikalen in rad
- $F(t)$ Kraft auf die Katze in N
- $P(t)$ Moment am Lastarm in N m
- g Erdbeschleunigung $9,81 \text{ m/s}^2$.

Für die Variablen x, φ erhält man, entsprechend zu Gleichung (1), die Differentialgleichungen (vgl. [24]): (Aus Gründen der besseren Lesbarkeit wird der Zeitparameter t weggelassen.)

$$(M + m)\ddot{s} + mL \cos(\psi)\ddot{\psi} = mL \sin(\psi)\dot{\psi}^2 + F \quad (4)$$

$$mL \cos(\psi)\ddot{s} + (mL^2 + \Theta)\ddot{\psi} = -gmL \sin(\psi)P \quad (5)$$

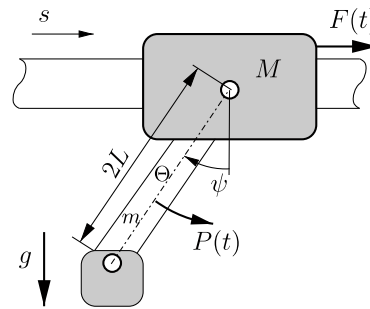


Bild 3 Skizze des Portalkrans.

Als System von Differentialgleichungen erster Ordnung geschrieben, ergibt sich das Gleichungssystem (6), (7) für die Variablen $x_1 = s, x_2 = \dot{s}, \varphi_1 = \psi$ und $\varphi_2 = \dot{\psi}$:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(\varphi_1)F + g_1(\varphi_1, \varphi_2, P) \end{aligned} \quad (6)$$

mit dem Systemterm

$$f_1(\varphi_1)F = \frac{(ML^2 + \Theta)F}{M(mL^2 + \Theta) + m^2L^2 \sin^2(\varphi_1) + m\Theta}$$

und dem Beeinflussungsterm

$$g_1(\varphi_1, \varphi_2, P) = \frac{(mL^2 + \Theta)mL \sin(\varphi_1)\varphi_2^2 - gm^2L^2 \cos(\varphi_1) \sin(\varphi_1) - mL \cos(\varphi_1)P}{M(mL^2 + \Theta) + m^2L^2 \sin^2(\varphi_1) + m\Theta}$$

$$\begin{aligned} \dot{\varphi}_1 &= \varphi_2 \\ \dot{\varphi}_2 &= f_2(\varphi_1, \varphi_2, P) + g_2(\varphi_1)F \end{aligned} \quad (7)$$

mit dem Systemterm

$$f_2(\varphi_1, \varphi_2, P) = \frac{m^2L^2 \cos(\varphi_1) \sin(\varphi_1)\varphi_2^2 - (M + m)gmL \sin(\varphi_1) + (M + m)P}{M(mL^2 + \Theta) + m^2L^2 \sin^2(\varphi_1) + m\Theta}$$

und dem Beeinflussungsterm

$$g_2(\varphi_1)F = \frac{mL \cos(\varphi_1)F}{M(mL^2 + \Theta) + m^2L^2 \sin^2(\varphi_1) + m\Theta}$$

Zeitdiskretisierung mit Abtastrate Δt ergibt das Gleichungssystem:

$$\begin{aligned} x_1(t + \Delta t) &= x_1(t) + \Delta t x_2(t) \\ x_2(t + \Delta t) &= x_2(t) + \Delta t f_1(\varphi_1(t))F(t) \\ &\quad + \Delta t g_1(\varphi_1(t), \varphi_2(t), P(t)) \end{aligned} \quad (8)$$

$$\begin{aligned} \varphi_1(t + \Delta t) &= \varphi_1 + \Delta t \varphi_2 \\ \varphi_2(t + \Delta t) &= \varphi_2(t) + \Delta t f_2(\varphi_1(t), \varphi_2(t), P(t)) \\ &\quad + \Delta t g_2(\varphi_1)F(t) \end{aligned} \quad (9)$$

Das Normalverhalten des Kranfahrers und des Entladers seien jeweils festgelegt durch $x_2 = 0, F \equiv 0$, bzw.

$\varphi_1 = 0, \varphi_2 = 0, P \equiv 0$. Damit ergeben sich die folgenden Aufgabenstellungen:

Tolerierbare Steueraktionen des Kranfahrers. Finde Steuerfolgen $\{F(t_0), F(t_0 + 1), \dots, F(t_0 + T - 1)\}$, so dass das System aus den Anfangswerten $(x_1(t_0) = s_0; x_2(t_0) = 0)$ in den Endwert $(x_1(t_0 + T) = s_{\text{soil}}; x_2(t_0 + T) = 0)$ überführt wird, unter der Nebenbedingung $(\varphi_1(t_0) = \varphi_1(t_0 + T) = 0; \varphi_2(t_0) = \varphi_2(t_0 + T) = 0)$ mit $P(t) \equiv 0$.

Tolerierbare Steueraktionen des Entladers. Finde Steuerfolgen $\{P(t_0), P(t_0 + 1), \dots, P(t_0 + T - 1)\}$, so dass das System aus den Anfangswerten $(\varphi(t_0) = 0; \varphi_1(t_0) = 0)$ in den Endwert $(\varphi_1(t_0 + T) = \varphi_{\text{soil}}; \varphi_2(t_0 + T) = 0)$ überführt wird, unter der Nebenbedingung $(x_1(t_0) = x_1(t_0 + T) = 0; x_2(t_0) = x_2(t_0 + T) = 0)$ mit $F(t) \equiv 0$.

Tolerierbare Steuerfolgen wurden mit der im Abschnitt 4.3 beschriebenen Heuristik für verschiedene Sollzustände berechnet. Die Trajektorien, die während der Aktionsausführung entstehen, werden in den folgenden Diagrammen (Bilder 4 und 6) dem Normalverhalten gegenübergestellt.

Die Startbedingung war jeweils: $(x_1, x_2, \varphi_1, \varphi_2) = (0, 0, 0, 0)$ und die Steuerwerte $F(t)$ (bzw. $P(t)$), für $(t = 1, \dots, 12)$ wurden aus dem Intervall $(-10; 10)$ gewählt.

Der Kranfahrer sollte die Position/Geschwindigkeit $(x_1; x_2) = (3; 1)$ erreichen und dabei den Endwert $(\varphi_1; \varphi_2) = (0; 0)$ einhalten, der dem Normalverhalten $(\varphi_1(t); \varphi_2(t)) \equiv (0; 0)$ entspricht.

Der Entlader sollte $(\varphi_1; \varphi_2) = (0, 3; 0)$ erreichen und die Endbedingung $(x_1; x_2) = (0; 0)$ erfüllen die zum Normalverhalten $(x_1(t); x_2(t)) \equiv (0; 0)$ gehört.

In einem ersten Berechnungsansatz, dessen Ergebnis in Bild 4 und 5 dargestellt ist, wurden nur die Endwerte vorgegeben und somit das Problem in eine reine Steuerungsaufgabe überführt.

Durch Einfügen des Zusatzterms $\gamma(\|\varphi_1\| + \|\varphi_2\|)$ in Gleichung (3) der TABU-SEARCH-Routine kann aber auch erreicht werden, dass die Störung des Armwin-

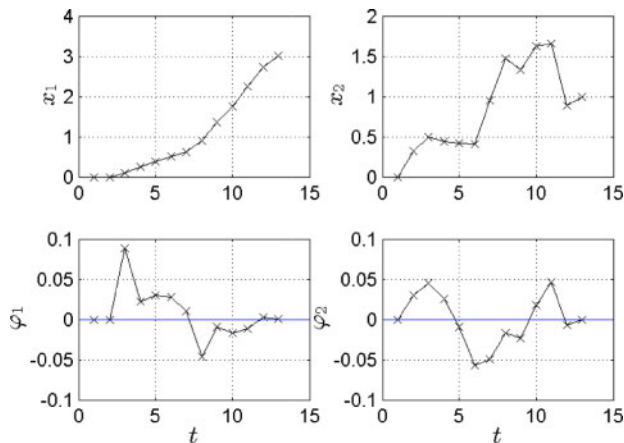


Bild 4 Trajektorien der Wegsteuerung bei reiner Endwertvorgabe.

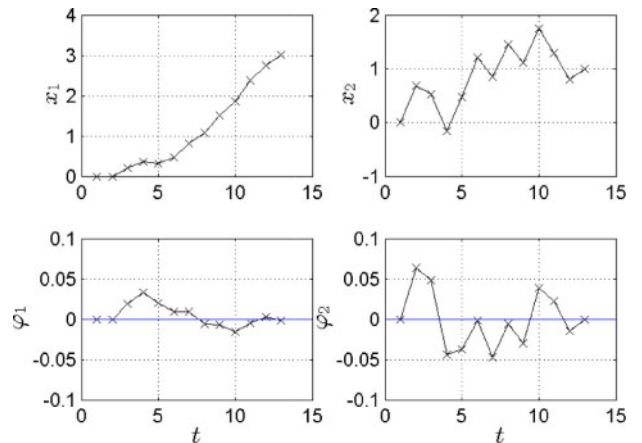


Bild 5 Trajektorien der Wegsteuerung bei Winkelbeschränkung über entsprechende Kostenkomponente (soft constraint).

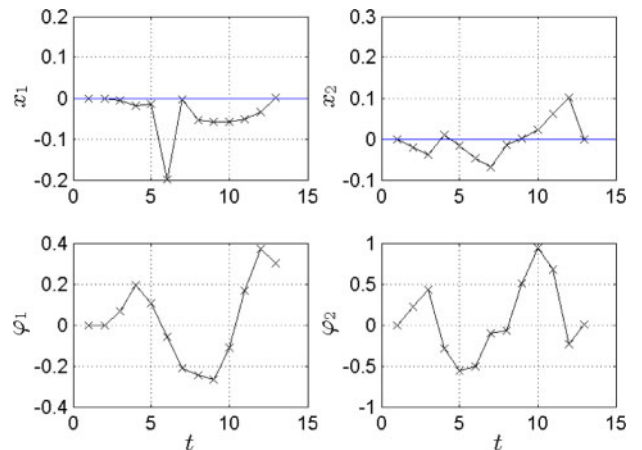


Bild 6 Trajektorien der Winkelsteuerung bei reiner Endwertvorgabe.

kels während der gesamten Steuerungszeit klein bleibt. Wird dabei der Wert γ in der Endphase der Berechnung sehr klein gemacht und damit die von ihm ausgehende ‚Störung‘ der Endwertoptimierung reduziert, so wird auch die Endwertforderung $(\varphi_1(t); \varphi_2(t)) \equiv (0; 0)$ mit hoher Genauigkeit erfüllt (vgl. Bild 5).

Schon dieses einfache Beispiel zeigt die Vorteile eines CI-Verfahrens gegenüber den üblichen Berechnungsansätzen (z. B. mittels einer prädiktiven Regelung [25]). Auch ein CI-Verfahren kann natürlich keine Lösung zu einem Problem liefern, das rein algorithmisch nicht wenigstens approximativ lösbar ist. Aber während bei den üblichen Ansätzen geänderte Bedingungen und Anforderungen oft die vollständige Änderung des Lösungsalgorithmus erfordern, werden die Lösungskonzepte der CI-Verfahren dadurch meist nur wenig betroffen und es werden nur geringe Modifikationen in deren numerischer Realisierung erforderlich. Dadurch wird es möglich, diese Bedingungen für die Lösungen erst während des Lösungsvorgangs exakt zu spezifizieren und sie den Lösbarkeitseigenschaften des Problems anzupassen. Diese Methode entspricht daher den Anforderungen, die heute verstärkt in der industriellen Praxis gefordert werden.

5 Diskussion

Das Problem der Koordinierung komplexer Systeme in der Anlagentechnik, der Robotik, dem Fahrzeugbau und bei Energieversorgungssystemen wird in der Zukunft wegen den immer stärker integrierten Systemen und den einschränkender werdenden Auflagen auch für die Regelungstechnik an Bedeutung gewinnen. Dabei ist dieses Problem jedoch mit vielen anderen Anforderungen verknüpft, die technischer aber auch organisatorischer Art sein können. Am Beginn eines Projekts stehen oft nur vage Ideen, die erst mit dem Projektfortschritt genauer präzisiert werden, wobei sie sich gegenseitig beeinflussen. Es ist zunächst unklar, welches Zielverhalten überhaupt erfüllbar ist und ob Zusatzanforderungen hinzugefügt werden müssen, damit sich ein eindeutiges Systemverhalten ergibt. Manche Festlegungen können erst getroffen werden, wenn dem Kunden anhand des Erreichten eine Vorstellung vom endgültigen System vermittelt werden kann. Diesen Bedingungen soll das hier vorgeschlagene Verfahren Rechnung tragen. Es ist also keinesfalls ‚besser‘ als klassische Vorgehensweisen sondern ‚anders‘. Im Vergleich etwa mit einem prädiktiven Regler-Ansatz und der Verwendung inverser Modelle, ist völlig unstrittig, dass dieser für das einfache Beispiel in Abschnitt 4.3 (wenn nur die Endwerte betrachtet werden) mathematisch effizientere Lösungen liefern würde. Der klassische Ansatz hat nur den Nachteil, dass er keine Aussage liefert, bevor das Problem mathematisch vollständig definiert ist, in sensibler Weise von kleinen Systemänderungen abhängt und es nicht möglich macht, die letztendlich erfüllten Bedingungen in einem ‚experimentellen‘ Prozess festzulegen und an das System anzupassen. Ein CI-Ansatz ist dagegen ‚unspezifisch‘. Er kann Feinheiten des Problems zur besonders effizienten Lösungsfindung nicht nutzen, aber er hängt auch nicht von solchen spezifischen Eigenschaften ab. Die Methoden der CI liefern keine ‚besseren‘ Lösungen als die klassischen Verfahren, ihr Vorteil zeigt sich nicht in der letztlich gefundenen Lösung, sondern am Weg, der zu dieser Lösung führt.

Dieser Weg besteht aus drei Schritten: der Bereitstellung von Lösungskonzepten, einem Auswahlverfahren, das die geeigneten Konzepte auswählt und schließlich deren Umsetzung in konkrete Algorithmen.

Der erste Schritt besteht üblicherweise darin, Verfahren aus der Natur zu übernehmen. Dabei haben wir in Abschnitt 3 gezeigt, dass es durchaus auch theoretische Methoden gibt um Lösungskonzepte zu generieren.

Der zweite Schritt (das Auswahlverfahren) erfolgt auf Grund eines Konzept-Vergleichs. Es werden charakterisierende Eigenschaften der Lösungskonzepte mit den Eigenschaften des Problems verglichen und nach Korrespondenz ausgewählt. Dabei ist die Eignung bzw. Nichteignung i. A. nicht beweisbar. Weder die Auswahl der Regel 3 in Abschnitt 3 noch die Wahl in Abschnitt 4.2 erfolgte rein logisch.

Beim dritten Schritt (der Umsetzung in Algorithmen) befinden wir uns im Zuständigkeitsbereich der Informa-

tik. Unsere einfachen Beispiele sollten eine Vorstellung von der benötigten Algorithmik vermitteln.

6 Zusammenfassung und Ausblick

Die betrachtete Vorgehensweise zeigte, wie in der CI Methoden aus ganz unterschiedlichen Wissensgebieten zusammengeführt werden, der Bionik, der Informatik, der KI, oder den Kognitionswissenschaften. Dem entspricht, dass auch die eingesetzte Intelligenz nicht nur eine algebraisch mathematische Intelligenz sein kann, sondern in Richtung einer ‚allgemeinen‘ Intelligenz gehen muss [6]. Wäre etwa die Eignung eines CI-Verfahrens für ein Problem beweisbar, so müssten alle Bearbeitungsmittel in den Rahmen einer mathematischen Theorie eingebracht werden und damit wären alle Methoden einer ‚allgemeinen‘ Intelligenz ausgeschlossen. Gerade diese Methoden sind es aber, mit denen Heuristiken wirksam eingesetzt werden können und die somit die Vorteile der CI-Methoden begründen. Ein CI-Verfahren sollte für die Intelligenz und das Vorwissen seines Anwenders offen sein und diesem in einfacher Weise erlauben, die Entwurfs-Entscheidungen einzubringen, in denen er überlegen ist.

Es gibt natürlich auch Ansätze, künstliche Systeme mit einer ‚allgemeinen‘ Intelligenz auszustatten und deren Wirksamkeit theoretisch zu verstehen. Hierbei spielt die Frage: Was ist ‚allgemeine Intelligenz‘ die entscheidende Rolle [7]. Aber auch schon ohne dass wir hier auf die Fragen einer AGI genauer eingegangen sind erkannten wir, dass es wesentlich ist, dass ‚allgemein intelligente‘ Systeme auf ‚allgemeinen Begriffsbildungen‘ oder Konzepten basieren. Unser Vergleich zwischen den klassischen und den CI-Verfahren hat darüber hinaus gezeigt, dass CI-Verfahren genau solchen ‚allgemein intelligenten‘ Systemen entsprechen. Ein theoriebasierter Entwurf oder eine Auswahlstrategie für CI-Verfahren sollte daher ebenfalls auf Konzepten begründet sein. Eine Theorie einer ‚künstlichen allgemeinen Intelligenz‘ (AGI) muss die Grundlagen zur Automatisierung von CI-Verfahren bereitstellen.

Die vorliegende Arbeit möchte in die neuen Denkweisen der AGI einführen, durch die sich das Arbeiten mit CI-Verfahren von klassischen Vorgehensweisen unterscheidet.

Literatur

- [1] A. Isidori, „Nonlinear control systems: an introduction“, *Lecture Notes Contr. Inf. Sci.* 72, Springer, Berlin, 1985.
- [2] H. Nijmeijer, A. J. van der Schaft, „Nonlinear Dynamical Control Systems“, *Springer*, New York, 1989.
- [3] H. Schwarz, „Nichtlineare Regelungssysteme: Systemtheoretische Grundlagen“, *Oldenbourg Wissenschaftsverlag*, München, 1991.
- [4] J. C. Bezdek, „What is Computational Intelligence?“, in: J. M. Zurada, R. J. Marks, C. J. Robinson, editors, *Computational Intelligence Imitating Life*, pp. 1–12, IEEE Press, New York, 1994.
- [5] U. Hammel, B. Naujoks, H. P. Schwefel, „Computational Intelligence Methods“, in: H. P. Schwefel, I. Wegner, K. Weinert, editors, *Advances in Computational Intelligence: Theory and Practice*, pp. 1–2, Springer-Verlag, Berlin, Heidelberg, 2003.

- [6] B. Goertzel, C. Pennachin, „Artificial general intelligence“, *Springer*, New York, 2007.
- [7] H. Sommer, L. Schreiber, „Is Logic in the Mind or in the World? Why a Philosophical Question can Affect the Understanding of Intelligence“, *Journal of Artificial General Intelligence* 3(1) 25–47, 2012.
- [8] K. Traube, „Nach dem Super-GAU: Tschernobyl und die Konsequenzen“, *Band 880 von Rororo Aktuell*, 1986.
- [9] K. J. Reinschke, „Multivariable Control: A Graph-Theoretic Approach“, *Akademie-Verlag*, 1988.
- [10] G. Russo, J. J. E. Slotine, „Symmetries, Stabilities, and Control in Nonlinear Systems and Networks“, *Physical Review E* 84, 2011.
- [11] J. Lunze, „Synchronisierbarkeit autonomer Agenten durch vernetzte Regelung“, *at – Automatisierungstechnik* 59, 531–538, 2011.
- [12] M. Weber, „Die ‚Objektivität‘ sozialwissenschaftlicher Erkenntnis (1904)“, in: *Ders.: Gesammelte Aufsätze zur Wissenschaftslehre*, Tübingen, 1988.
- [13] A. Kroll, H. Sommer, „Coordination of Distributed Control Systems“, in: D. M. Dubois, editors, *CASYS'10, AIP Conference Proceedings*, 2012.
- [14] L. P. Kadanoff, „More is the Same; Mean Field Theory and Phase Transitions“, *J. Stat. Phys.* 137, 777–787, 2009.
- [15] M. R. Garey, D. S. Johnson, „Computers and intractability: A Guide to the theory on NP-completeness“, *W. H. Freeman and Company*, 1979.
- [16] P. Lin, Y. Jia, „Distributed rotating formation control of multi-agent systems“, *Systems and Control Letters* 59, 587–595, 2010.
- [17] M. Gassner, „Informationsauswertung in dezentralen Regelsystemen“, *Diplomarbeit*, Universität Kassel FB 15 Mess- und Regelungstechnik, 2011.
- [18] E. Osherovich, „Ant Robotics: Covering Continuous domains by Multi-A(ge)nt Systems“, *arXiv:1206.4185v1[cs.RO]* 19 Jun 2012.
- [19] G. Evenbly, „Foundations and Applications of Entanglement Renormalisation“, *arXiv:1199.5424v1[quant-ph]*, 2011.
- [20] S. Rana, S. Jasola, R. Kumar, „A Review on Particle Swarm Optimization Algorithms and their Applications to Data Clustering“, *Artificial Intelligence Review*, 35, 3, 211–222, 2011.
- [21] J. P. Watson, L. D. Whitley, A. E. Howe, „Linking Search Space Structure, Run-Time Dynamics, and Problem Difficulty: A Step Toward Demystifying Tabu search“, *J. Artificial Intelligence Research* 24, 221–261, 2005.
- [22] F. Glover, Z. Lü, J.-K. Hao, „Diversification-driven tabu search for unconstrained binary quadratic problems“, *4OR-Q. J. Oper. Res.*, DOI:10.1007/S10288–009/0115-y, 2010.
- [23] Z. Lü, F. Glover, J.-K. Hao, „A Hybrid Metaheuristic Approach to Solving the UBQP Problem“, *European J. Operational Research*, 207, 3, 1254–1262, 2010.
- [24] A. Castaneda, „Control de una grua utilizando sistemas de Tagaki Sugeno“, *Technischer Bericht*, Universität Kassel FB 15 Mess- und Regelungstechnik, 2008.
- [25] R. Dittmar, B. Pfeiffer, „Modellbasierte prädiktive Regelung in der industriellen Praxis“, *at – Automatisierungstechnik*, Oldenbourg Wissenschaftsverlag, 54, 590–601, 2006.

Manuskripteingang: 16. März 2012



Dipl.-Ing. Axel Dürrbaum ist Labor-Ingenieur im Fachgebiet Mess- und Regelungstechnik der Universität Kassel. Hauptarbeitsgebiete: Algorithmen, Fuzzy-Technik, Simulation regelungstechnischer Systeme.

Adresse: Universität Kassel, Fachbereich Maschinenbau, Fachgebiet Mess- und Regelungstechnik, Mönchebergstr. 19, 34109 Kassel,
E-Mail: axel.duerrbaum@mrt.uni-kassel.de



Univ.-Prof. Dr.-Ing. Andreas Kroll ist Leiter des Fachgebiets Mess- und Regelungstechnik der Universität Kassel. Hauptarbeitsgebiete: Nichtlineare Identifikations- und Regelungsmethoden, Computational Intelligence, komplexe Systeme.

Adresse: Universität Kassel, Fachbereich Maschinenbau, Fachgebiet Mess- und Regelungstechnik, Mönchebergstr. 19, 34109 Kassel,
E-Mail: andreas.kroll@mrt.uni-kassel.de



Dr. Hanns-Jakob Sommer ist wissenschaftlicher Mitarbeiter im Fachgebiet Mess- und Regelungstechnik der Universität Kassel. Hauptarbeitsgebiete: nichtlineare Regelungstechnik, komplexe Systeme, Artificial general intelligence.

Adresse: Universität Kassel, Fachbereich Maschinenbau, Fachgebiet Mess- und Regelungstechnik, Mönchebergstr. 19, 34109 Kassel,
E-Mail: hans.sommer@mrt.uni-kassel.de