

SIMPLORER ALS REMOTE-SIMULATOR

Dr. Ing. Lutz Zacharias

Ansoft GmbH & Co KG, Chemnitz

zacharias@simplorer.com

Kurzfassung

Das Simulationssystem SIMPLORER wird von der Firma Ansoft entwickelt und vertrieben. Es wurde speziell für die Belange der Elektronik / Antriebstechnik und Mechatronik entwickelt. Dieser Beitrag stellt die aktuellen Arbeiten zur Remote – Simulation mit dem Simulationswerkzeug SIMPLORER vor.

1. Einleitung

Die Firma Ansoft GmbH & Co KG hat den Simulator SIMPLORER entwickelt, der für mechatronische, kontinuierliche und diskrete Systeme geeignet ist. Seit mehreren Jahren bietet die Firma Unterstützung auf dem Gebiet der Designoptimierung an. Im Rahmen der BMBF-Projekte MIMOSYS und DEMIS wurden Optimierungsverfahren entwickelt, implementiert und getestet.

Das System SIMPLORER wird im OMID-Projekt zur Modellierung, Simulation und Optimierung von Anwendungen eingesetzt. Dabei werden die bereits vorhandenen Ergebnisse umgesetzt und anwendbar gemacht. Ziel ist die Verifikation der Tools am konkreten Designprozeß (mikrofluidischer Prozess, Aktorplatte) und die Integration in die Simulationsumgebung SIMPLORER. Diese wird so erfolgen, daß die Unterstützung für die verschiedenen Entwurfsschritte gegeben ist. Besonderer Schwerpunkt ist dabei die Anpassung der Entwicklungsumgebung an die Erfordernisse des Designprozesses. Dazu muss

eine über die reine Simulation hinausgehende Lösung erarbeitet werden, die auch in der Lage ist, Arbeitsabläufe abzubilden und sich in heterogene Arbeitsumgebungen einzuordnen.

Die angestrebte Lösung führt zu einer breit nutzbaren Umgebung mit integrierten Hilfstools, wie sie so derzeit am Markt nicht angeboten wird. Für den Aufbau eines Dienstleistungsangebots über INTERNET ist die Remotefähigkeit der einzelnen Tools Grundvoraussetzung. Durch die Verwendung von **ORBIT** (Internet basierte Simulation und Optimierung) wird die Umgebung remotefähig gemacht werden, womit international neuartige Möglichkeiten des Mikrosystementwurfs entstehen.

2. Das Multi-Domain-Simulationswerkzeug SIMPLORER

SIMPLORER, als ein kombinierter Schaltungs- und Systemsimulator adressiert Anwender der Bereiche Mikrosystemtechnik, Mechatronik, Leistungselektronik, Elektrische Antriebs- und Energietechnik sowie MSR-Technik. Einfach und schnell können Anordnungen mit sowohl elektrischen als auch nichtelektrischen Komponenten modelliert und simuliert werden. Dabei werden die nachzubildenden Anordnungen praxisnah mit Hilfe von elektrischen Netzwerken, regelungstechnischen Blockdiagrammen und Zustandsgrafiken beschrieben. Stabile Rechenalgorithmen sichern, im Gegensatz zu SPICE-basierten Simulatoren, numerische Konvergenz auch bei schnell schaltenden elektronischen Systemen.

Die umfangreiche Modelldatenbank beinhaltet neben aktiven und passiven elektrischen Grundbauelementen, Operationsverstärkern, regelungstechnischen Übertragungsfunktionen und Nichtlinearitäten auch statische und dynamische Leistungshalbleiter, Sensoren, elektrische Maschinen sowie Mechanik- und Hydraulikkomponenten. Gerade in Gestalt der angeführten Bibliothekserweiterungen stellt SIMPLORER nunmehr noch bessere Funktionalitäten auch außerhalb seiner üblichen Domänen zur Verfügung.

Mit der neu verfügbaren Mechanik-Toolbox kann man translatorisch und rotatorisch bewegte Systeme modellieren. Die für die Arbeit mit der Toolbox relevanten Parameter sind Trägheitsmomente, Steifigkeiten sowie Dämpfungs- und Reibungskoeffizienten. Die in der Bibliothek enthaltenen Basisbausteine sind u.a. Masse-Element (MAS), Steifigkeitselement (STF), Koordinatentransformation (TSF), Getriebeübersetzung, Momentenquelle (SRCF) sowie mechanisches "Bezugspotential" (GND).

So können alle vorhandenen SIMPLORER-Standardmodelle von translatorischen oder rotatorischen Aktoren durch die Ergänzung weiterer mechanischer Modelleigenschaften in ihrer Detailtreue verbessert werden. Das kann zum Beispiel durch das Ankoppeln eines entsprechenden Masselementes geschehen.

Anwendungsvorteile bei Einsatz der SIMPLORER Mechanik-Toolbox ergeben sich insbesondere durch

- Einfacheres und schnelleres Abbilden von (mikro)mechanischen Systemen im Vergleich zur Modellierung mit blockorientierten Signalflußgraphen.
- Erhöhung der numerischen Stabilität und der Rechengeschwindigkeit bei der Simulation mechanischer und mechatronischer Systeme.

Die des weiteren neu verfügbare Hydraulikbibliothek adressiert Entwickler von hydraulischen und elektrohydraulischen Systemen,

der Mikrofluidik, sowie Anwender, die neben der Hydraulikseite auch Leistungselektronik, elektrische Antriebe, Regelungen oder Mechanik betrachten wollen.

Die grafische Repräsentation der hydraulischen Modellkomponenten orientiert sich im Unterschied zu anderen Modellierungsansätzen durchweg an den entsprechenden DIN-ISO Normen. Die Modellbibliothek beschreibt das stationäre Verhalten hydraulischer Systeme sehr genau. Die Parametrierung der vorhandenen Simulationsmodelle geschieht mit Hilfe durchschaubarer und für den Anwender einfach zu beschaffender Parameter. Die Bibliothek enthält u.a. verschiedene Pumpentypen (konstante/variable Fördermenge) und Hydraulikzylinder, eine Vielzahl von Ventilkombinationen sowie Hydromotoren und Druckspeicher. Nützliche Kleinigkeiten, wie z.B. Modelle von End- und Druckschaltern oder Manometern runden die Funktionalität ab.

Durch das Vorhandensein animierter graphischer Symbole kann die Wirkungsweise hydraulischer Systeme praxisnah visualisiert werden, so daß sich der Einsatz der hier vorgestellten Modellbibliothek auch im Ausbildungssektor anbietet.

Verfügbare Erweiterungsmodule, wie Optimierung und Programmierschnittstellen für C, Pascal, Fortran runden die Funktionalität weiter ab. Ein offenes Simulatorinterface gestattet die Kopplung mit Matlab/Simulink, MathCad, speicherprogrammierbaren Steuerungen sowie elektromagnetischer Feldberechnungssoftware auf FEM-Basis (Maxwell-EM-Solver-Produktfamilie).

3. Implementation der Remote-Fähigkeit

Ziel des Entwicklungsarbeiten ist die Integration von SIMPLORER in ORBIT.

SIMPLORER in der Version 4.x ist als PC-Programm unter WINDOWS 9.x NT4 WIN 2000 und WIN ME lauffähig. Ein SIMEC eigenes Kommunikationsprotokoll realisiert die Kommunikation der Programme untereinander. Eine Fernsteuerung von SIMPLORER durch Scripte ist zum gegenwärtigen Zeitpunkt nicht möglich.

Es ergaben sich folgende, zu lösende Aufgaben:

Mehrfach Instanzierung des Simulators ermöglichen Script – Steuerbarkeit Schaffung der Remote –Fähigkeit von SIMPLORER Die Arbeitspakete wurden wie folgt untersetzt:

- Mehrfach Instanzierung des Simulators
- Modifikation des Speicher-managements
- Interfacedefinition für Script Access
- Script – Steuerbarkeit
- Starten / Kontrolle der laufenden Simulation
- Modelltransport
- Ergebnistransport
- Mehrfach-Instanzierung des Simulators
- Modifikation des Speicher-managements

Die SIMPLORER-Version 4.x ist standardmäßig nicht in der Lage, mehrfach auf einem PC aufgerufen zu werden. Hauptaufgabe für dieses Arbeitspaket war die Modifikation der Berechnungsmodule und der internen Modellverwaltung. Die Anpassung erfolge in Hinblick auf die voll dynamische Speicherverwaltung verbunden mit einer schnellen Lösung der Simulationsaufgabe. Durch den Einsatz moderner Analysewerkzeuge konnte das

Management so optimiert werden, dass maximale Effektivität erreicht wurde. Ein erster Prototyp ist verfügbar und steht zu Testzwecken bereit.

Bedingt durch die Veränderungen in der Speicherverwaltung musste die Modellverwaltung vollkommen neu konzipiert und implementiert werden. Hierbei kommen objektorientierte Programmierung und ein neuer hierarchischer Ansatz für die Modellverwaltung zum Einsatz. Dieser ist gleichzeitig die Grundlage für die Parameterübernahme, die im Arbeitspaket Modelltransport realisiert werden soll.

3.1. Interfacedefinition für Script Access

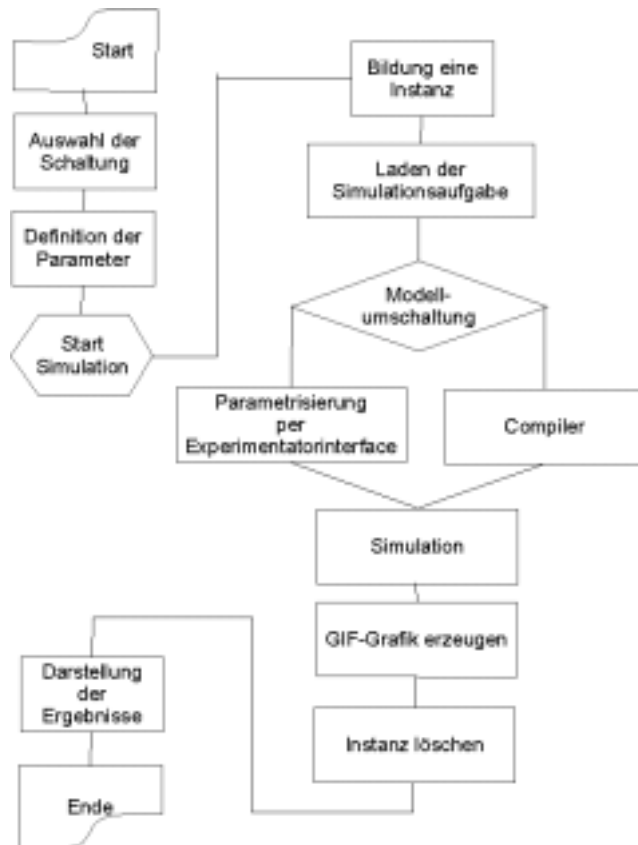
Der Modell – und Parametertransport stellt besondere Anforderungen an das System. Er muss unabhängig von der Hardware und dem Betriebssystem gestaltet sein. Gleichzeitig muss die Multiuser-Funktionalität beachtet werden. D.h. es müssen die Modelle, Parameter, Job-Steuerung und Auswertung separat steuerbar sein.

Für die erste Phase der Realisierung wurden folgende Vereinfachungen getroffen:

- Verwendung von SML – Modellen in Textform
- Vordefinierte Parameterliste
- Keine Fehlerüberwachung der Eingabe
- Keine Modellumschaltung

Durch diese Vereinfachungen können Modelle mit der aktuellen Version von SIMPLORER erzeugt und durch die Remote – Lösung von SIMPLORER simuliert werden. Der Modelltransport wird durch Steuerscripte realisiert.

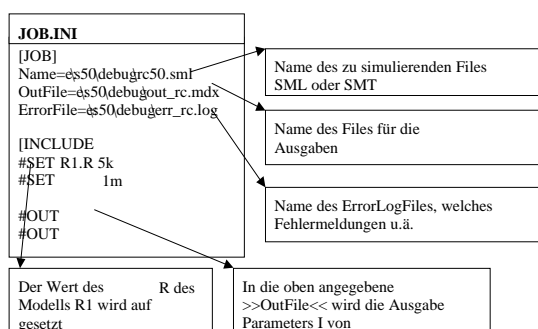
Die Abbildung zeigt den groben Ablauf, der beschrieben wird, wenn ein Anwender die Simulation startet.



Der Transport der Simulationsaufgabe wird durch das JOB – Beschreibungsfile realisiert. Dieses ASCII File beinhaltet alle Informationen, die zur Lösung der gestellten Simulationsaufgabe notwendig ist.

- Modell
- Ziel für die Ausgaben
- Ziel für Fehlermeldungen
- Definition der Modellparameter
- Definition der Ausgaben

Die folgende Abbildung illustriert den Aufbau in der 1. Version.



3.2. Script – Steuerbarkeit

Starten / Kontrolle der laufenden Simulation
Der Startvorgang der Simulation wird durch den Startmanager realisiert. Dieser wurde speziell für die Bedürfnisse von SIMPLORER entwickelt. Durch den Startmanager wird gewährleistet, dass ein einfacher Aufruf von einem VB Script oder PHP - Script möglich ist. Der Manager behält die Kontrolle, bis die Simulation vollständig abgearbeitet oder abgebrochen wurde.

3.3. Modelltransport

Für jede Anforderung wird ein eigenes Environment erzeugt, das Modell aufgenommen und die Ausgabe vorbereitet.

3.4. Ergebnistransport

Die Ergebnisse speichert der Simulator direkt im vorgegebenen Environment. Die nachfolgende Bearbeitung erzeugt die notwendigen Grafiken, führt definierte Berechnungen aus und stellt alle Ausgaben als HTML Text bereit. Per http Protokoll werden diese Ausgaben zum Anwender übertragen.

4. Anbindung an ORBIT

Die Integration von SIMPLORER in das System ORBIT erfordert die Anbindung von SIMPLORER an das von der Firma InterConnect bereitgestellt Tool-Interface. Der ToolAgent basiert auf dem SOAP - Protokoll (Simple Object Access Protocol). Weitere Informationen zu SOAP sind unter www.w3c.org erhältlich.

Die Anbindung wurde für die SOAP Version 1.0 erstellt und wird auf die aktuelle SOAP Version 2.0 umgestellt. Per Test – Tool kann die Ansteuerung von SIMPLORER getestet

werden. Der verfügbare Prototyp gibt bereits Aussagen über notwendige Erweiterungen im Modelltransport - Protokoll. So fehlen zum Beispiel in der aktuellen Ausbaustufe Aussagen über die Modellausgaben. Aus SIMPLORER-Sicht ist es sinnvoll, daß der Anwender Ausgaben aus einem Fundus zu- oder abschalten kann. Das reduziert den auf der Festplatte benötigten Speicherplatz und erhöht auch die Simulationsgeschwindigkeit.

Gegenwärtig in der Diskussion ist weiterhin ein Cleaning – Mechanismus. Dieser wird notwendig, wenn man die Größe der Files von Simulationsergebnissen untersucht. Hier gilt es zu untersuchen, ob durch Komprimierung mit Standardverfahren eine Verbesserung zu erreichen ist. Besondere Beachtung gilt dabei den notwendigen Automatismen zur Komprimierung und Dekomprimierung, damit dieser Vorgang als transparent in Bezug auf den Anwender bezeichnet werden kann.

5. Remote - SIMPLORER Stand alone Version

Prototyp für die Remote-Simulation mit SIMPLORER. Parametrisierung von Komponenten unter Verwendung des WEB – Interfaces.

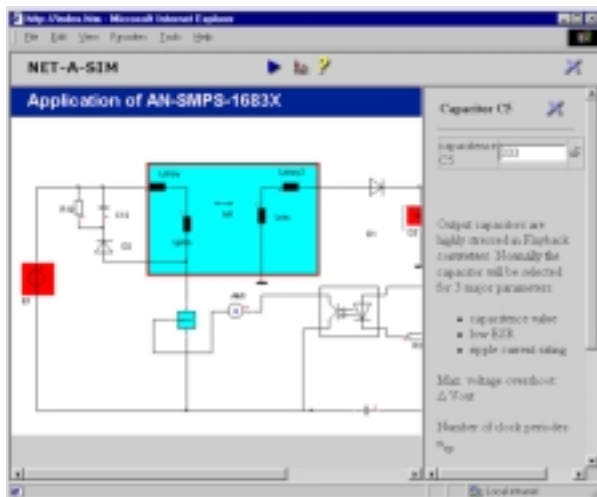


Abbildung 1: Parametereingabe

Voraussetzung für eine derartige Simulation ist das Vorhandensein und die volle Funktionsfähigkeit des Modells. Zusätzlich zum Modell werden noch die Boundaries Benötigt. Der Modellentwickler kann diese für jeden Parameter mit angeben. Aus diesen Informationen erstellen die Exportscripte den notwendigen JavaScript für den WEB-Client

Mit Hilfe diese WEB Interfaces kann der Anwender das Modell in den vorgegebenen Grenze variieren. Automatisch generierte Scripte prüfen auf der Client-Seite sofort die Eingaben auf deren Richtigkeit.

Der Anwender kann die Simulation vom Interface aus direkt starten. Es ist davon auszugehen, dass die Simulationszeit mit entsprechender Komplexität des Modells steigt. Statusinformationen informieren den Anwender automatisch über den Fortschritt der Berechnung.

Schon während der Simulation werden die Ergebnisse pro Zeitschritt zum Client übertragen. Damit erreicht man auch bei geringer Bandbreit akzeptable Simulationsgeschwindigkeiten erreichen, zumal der Hauptanteil der Last auf die Übertragung entfällt. In Kombination mit einer einfachen Online – Komprimierung der Ergebnisdaten erreicht man die geforderten Antwortzeiten des Systems.

Die Darstellung der Simulationsergebnisse erfolgen direkt im WEB- Browser unter Verwendung von JAVA- Applets. Die folgende Abbildung illustriert eine mögliche Präsentation der Simulationsergebnisse.

Als besondere Herausforderung hat sich die Darstellung in den unterschiedlichen Browser-Versionen herausgestellt.

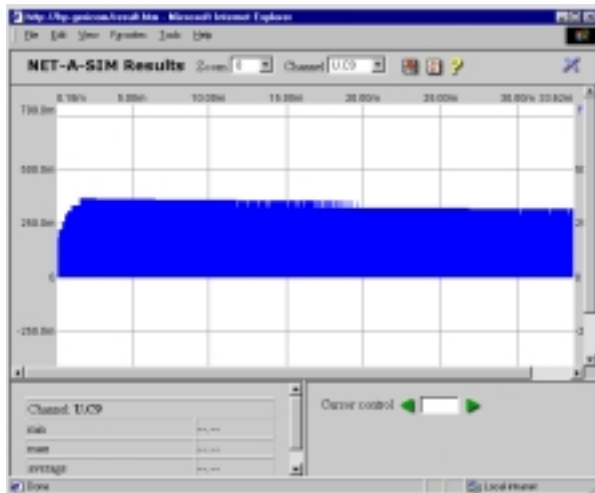


Abbildung 2 : Prototyp der Ergebnisdarstellung

6. Ausblick / Weiterentwicklung

Der erste Prototyp der WEB Simulation ist unter www.net-a-sim.com erreichbar und demonstriert die Machbarkeit der Remote-Simulation, unter der Annahme, dass das Modell in einer vorgefertigten und lauffähigen Form vorliegt.

Die folgende Entwicklungsstufe wird sich mit dem Thema der Modellparametereingabe beschäftigen. Hier gilt es automatisierte Lösungen zu entwickeln. Eine interessante Option liegt in der Möglichkeit, dass man direkt während der Modellierung oder beim Modell-export die Parameterbereiche angeben kann. Ein weiterer Entwicklungsschwerpunkt wird die Ergebnisausgabe sein. Hier sollen Lösungen untersucht werden, wie man unter Berücksichtigung von geringen Bandbreiten eine optimale Darstellung der Ergebnisse erreicht.

INTERNETBASIERTE DESIGNOPTIMIERUNG MIT HYGLEAM

Wilfried Jakob

Institut für Angewandte Informatik, Forschungszentrum Karlsruhe (FZK)

Email: jakob@iai.fzk.de

Dagmar Peters

Institut für Theoretische Elektrotechnik und Mikroelektronik, Uni Bremen (ITEM)

Email: peters@item.uni-bremen.de

Andreas Reiffer

InterConnect Software GmbH, Karlsruhe

Email: reiffer@interconnect.de

Kurzfassung

Mit der internetbasierten Variante von HyGLEAM können Optimierungen über einen Remoteaufruf auf einem entfernten Rechner eines Application Service Providers ausgeführt werden. HyGLEAM integriert zwei allgemein anwendbare lokale Suchverfahren in den global arbeitenden Evolutionären Algorithmus GLEAM und kombiniert die Vorteile der beteiligten Verfahren unter Vermeidung ihrer jeweiligen Nachteile. Dazu wird eine neue Methode zur Steuerung des Verfahrenseinsatzes benutzt. Die Robustheit und die gesteigerte Performance des resultierenden hybriden Verfahrens werden an Hand mathematischer Benchmarkfunktionen und dreier Designanwendungen nachgewiesen.

1 Einleitung

Das Verbundprojekt OMID (Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen) hat unter anderem zum Ziel, die im Vorgängerprojekt DEMIS entwickelten und getesteten Optimierungsverfahren zu integrieren, die dabei entstehenden Hybride zu bewerten und das resultierende Optimierungswerkzeug in einer webbasierten Remote-Version zur Verfügung zu stellen. In DEMIS wurde deutlich, dass alle untersuchten Verfahren ihre Stärken und Schwächen haben [1, 2]. So liefern lokale Suchverfahren wie der Rosenbrock-Algorithmus oder das Complex-Verfahren vergleichsweise schnell ein Ergebnis, das aber bei multimodalen Aufgabenstellungen meist nur ein lokales Optimum dar-

stellt. Globale Suchverfahren wie die Evolutionären Algorithmen (EA) bieten hingegen eine wesentlich größere Chance, das globale Optimum zu finden, benötigen dafür aber zum Teil wesentlich mehr Berechnungen der Qualitätsfunktion. Um die meist aufwendigen Simulationen zur Bestimmung der Qualität einer Designlösung zu verringern ohne den globalen Charakter der Suche aufzugeben, wurde versucht, die beiden Algorithmenklassen so geschickt miteinander zu verbinden, dass der resultierende Hybrid die Vorteile der beteiligten Verfahren ohne deren Nachteile in sich vereint. Erste Tests an Designoptimierungsaufgaben aus dem laufenden Projekt zeigten vielversprechende Ergebnisse [3].

In Kap. 2 werden die am FZK entwickelten Hybridisierungsmethoden [4, 5] vorgestellt. Integriert wurden zwei am ITEM implementierte lokale Suchverfahren [1] und das GLEAM-Verfahren (General Learning Evolutionary Algorithm and Method) [6, 7], das am FZK zu HyGLEAM (Hybrid General Purpose Evolutionary Algorithm and Method) weiterentwickelt und als GADO (Genetischer Algorithmus zur Designoptimierung) und HyGADO zur Designoptimierung eingesetzt wurde. Kap. 3 beschreibt die untersuchten Testfälle bestehend aus mathematischen Benchmarkfunktionen und OMID-Anwendungen. In Kap. 4 sind die Testergebnisse und die sich daraus ergebenden Schlussfolgerungen enthalten. Kap. 5 stellt die auf dem Orbit-System der Firma InterConnect [8] basierende Remote-Lösung von HyGLEAM vor, während Kap. 6 mit einer Zusammenfassung des Beitrags schließt.

2 Das HyGLEAM-Konzept

Das in [4, 5] ausführlicher dargestellte Konzept von HyGLEAM kann hier nur in der gebotenen Kürze erläutert werden. Zu den beiden lokalen Verfahren gibt Schwefel [9] eine ausführlichere Beschreibung mit weiteren Anwendungsfällen.

2.1 Benutzte Optimierungsverfahren

2.1.1 GLEAM

GLEAM ist ein eigenständiger evolutionärer Algorithmus, der Elemente der Evolutionsstrategie und der (reellcodierten) Genetischen Algorithmen mit Konzepten der Informatik (abstrakte Datentypen) verbindet. Die Codierung der zu optimierenden Parameter erfolgt in den sogenannten Aktionen. Eine Aktion besteht aus ein oder mehreren Parametern vom Typ Bool, Integer oder Real entsprechend ihrem anwendungsspezifisch definierten Aktionstyp. Eine Aktion drückt damit eine bestimmte Eigenschaft eines Designs aus, z.B. die Anwesenheit einer bestimmten Komponente, die durch die Aktionsparameter näher bestimmt wird oder die Relationen zu anderen Komponenten entsprechend den Aktionsparametern eingeht. Im einfachsten Fall codieren die Aktionen nur gewisse einzelne Parameter eines Designs. Die Aktionen bilden eine Aktionsskette (AK), die den Chromosomen des biologischen Vorbilds entspricht. Es gibt derzeit in GLEAM drei AK-Typen: Bei den ersten beiden ist jeder Aktionstyp mit genau einer Aktion vertreten und sie unterscheiden sich darin, ob die Reihenfolge der Aktionen relevant ist. Die Variante ohne bedeutungstragende Reihenfolge wird bei Designoptimierungsaufgaben, bei denen es um reine Parameteroptimierung geht, verwendet. Die andere Variante spielt bei OMID keine Rolle, sie kommt aber bei anderen Anwendungen wie z.B. Schedulingproblemen zum Einsatz [10]. Schließlich gibt es noch den AK-Typ mit variabler Länge, bei dem jeder Aktionstyp mit keiner, einer oder mehreren Aktionen in beliebiger Reihenfolge vertreten sein kann. Dieser Typ findet bei anspruchsvolleren Designoptimierungsaufgaben wie z.B. der in Kap. 3 beschriebenen Akkorplatte Verwendung.

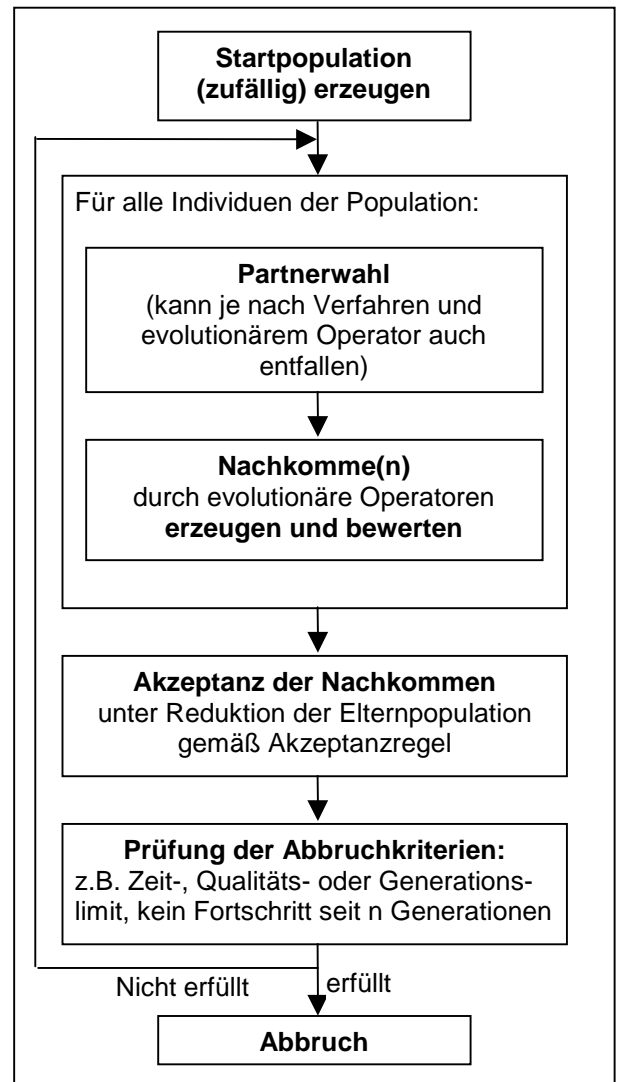


Abb. 1: Prinzipieller Ablauf Evolutionärer Algorithmen

Abb. 1 zeigt den prinzipiellen Ablauf Evolutionärer Algorithmen. Ausgehend von einer meist zufällig erzeugten Startpopulation, die durch manuell oder auf andere Weise erzeugte Lösungen angereichert werden kann, werden alle Individuen generationsweise der nachgebildeten Evolution unterworfen. Dazu werden Nachkommen durch Mutation und/oder durch Rekombination des Genmaterials zweier Eltern erzeugt. Im letzten Fall findet zuvor eine Partnerwahl statt, die meist auf der Fitness der Individuen beruht. In GLEAM wird dafür eine ranking-basierte Selektion benutzt, die die Chancen der besten Individuen, selektiert zu werden, etwas zu Gunsten der schlechteren dämpft. Dadurch soll unter anderem vorzeitiger Konvergenz auf ein Suboptimum vorgebeugt werden. GLEAM enthält einen Satz vordefinierter genetischer Operatoren (Muta-

tionen der Werte oder der Aktionsposition, verschiedene Crossover-Operatoren), die aufgabenspezifisch konfiguriert zur Anwendung gebracht werden können. Die Wertemutationen berücksichtigen die in den Aktionstypdefinitionen mithinterlegten Wertebereichsgrenzen, sodass explizite Restriktionen immer eingehalten werden. Kleine Änderungen sind dabei entsprechend der Evolutionsstrategie wahrscheinlicher als große. Pro Paarung werden in GLEAM mehrere Nachkommen erzeugt, wobei die dabei beteiligten genetischen Operatoren sowie ihre Anwendungswahrscheinlichkeiten problemspezifisch ausgewählt und konfiguriert werden können. Schließlich müssen die Nachkommen der Elterngeneration akzeptiert oder verworfen werden. Die akzeptierten Kinder ersetzen ebensoviel Mitglieder der Elterngeneration, sodass die Populationsgröße konstant bleibt. In GLEAM wird dazu eine elitäre Akzeptanzregel benutzt, bei der das beste Individuum unabhängig von seinem Alter immer überlebt, sodass eine Verschlechterung der pro Generation erreichten Lösungsqualität nicht eintreten kann.

2.1.2 Verfahren nach Rosenbrock

Kernprinzip des Verfahrens nach Rosenbrock [11] ist die Suche des Optimums entlang der Richtungen eines im n -dimensionalen Raum rotierenden Koordinatensystems. Beschränkungen werden mit einer partiellen, internen Straffunktion berücksichtigt.

Das in Abb. 2 dargestellte Verfahren setzt einen zulässigen Startpunkt voraus. Im Anschluss an die Initialisierung wird ein Testschritt in die aktuelle Richtung i gemacht. Sind alle Bedingungen erfüllt und ist der Wert der partiellen Barrierefunktion ermittelt worden, wird untersucht, ob es sich bei dem aktuellen Punkt um einen besseren handelt als bei dem Vorgänger. Ist dies der Fall, vergrößert sich die Schrittweite um den Faktor drei. Wurde mit dem aktuellen Punkt kein Erfolg erzielt, wird die Suche in die entgegengesetzte Richtung mit halbiertem Schrittweite fortgesetzt. Innerhalb einer Iteration muss in jede Richtung des aktuellen Koordinatensystems mindestens ein Erfolg gefolgt von einem Misserfolg erzielt werden, bevor das Abbruchkriterium überprüft wird.

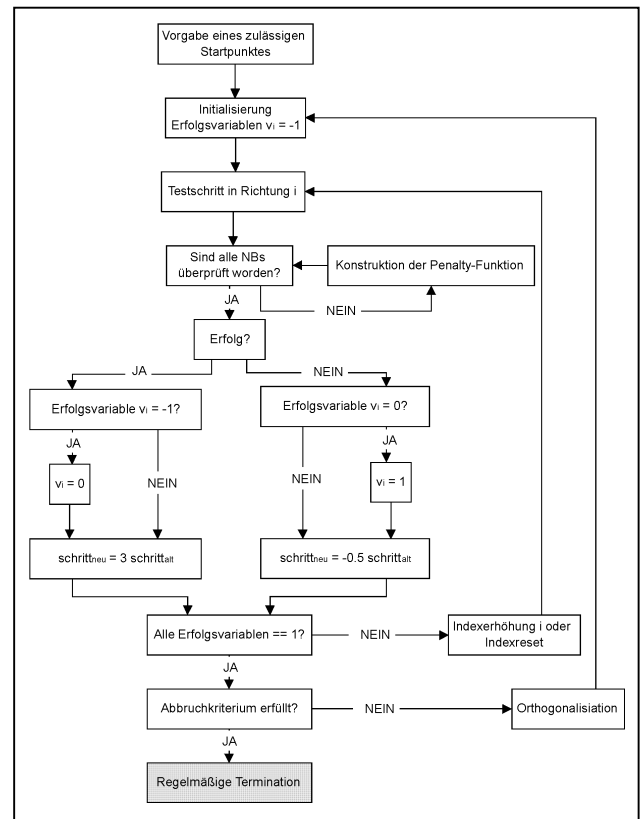


Abb. 2: Ablaufdiagramm des Rosenbrock-Verfahrens

Wird das Abbruchkriterium erfüllt, terminiert das Verfahren, ansonsten wird eine Rotation des Koordinatensystems durchgeführt. Dazu wird die vielversprechendste Richtung der letzten Iteration als Hauptachse des neuen Koordinatensystems festgelegt und die restlichen $n-1$ Achsen orthogonal zu dieser Achse aufgebaut. Mit diesem Koordinatensystem setzt sich die Suche des Optimums so lange fort, bis das Abbruchkriterium erfüllt wird.

2.1.3 Complex-Algorithmus

Das Complex-Verfahren wurde 1965 von Box [12] in Anlehnung an den Simplexalgorithmus für nichtlineare Funktionen von Nelder und Mead entwickelt. Der prinzipielle Ablauf ist in Abb. 3 dargestellt. Der Complex-Algorithmus wurde speziell für beschränkte Problemstellungen entwickelt und verwendet keinen der klassischen Ansätze: externe Straffunktion, interne Straf- bzw. Barrierefunktionen, Lagrange-Multiplier-Verfahren oder partielle Straffunktion.

Ausgehend von einem zulässigen Startpunkt, wird der Startcomplex gebildet. Es folgt die Ermittlung des schlechtesten Punktes sowie

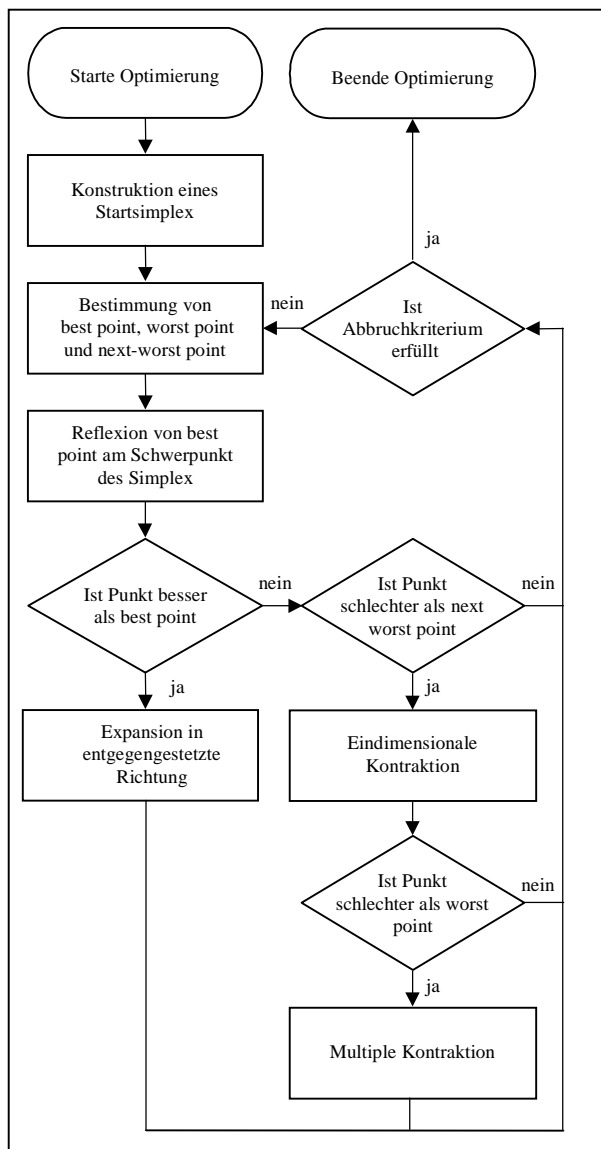


Abb.3: Prinzipieller Ablauf von Polyederverfahren

des Scheitelpunktes der verbleibenden Punkte. Ist der Scheitelpunkt zulässig, wird der schlechteste Punkt an ihm reflektiert. Dabei handelt es sich um eine Normalreflexion, bei der gleichzeitig eine Expansion des Complex durchgeführt wird. Bei Nichtzulässigkeit des Scheitelpunktes wird ein neuer Startcomplex gebildet und das Verfahren neu gestartet. Auch der reflektierte Punkt muss daraufhin überprüft werden, ob er alle formulierten Bedingungen erfüllt. Ist dies nicht der Fall, wird der Reflexionsfaktor halbiert und eine erneute Reflexion durchgeführt, bis ein zulässiger Punkt gefunden wird oder der Reflexionsfaktor kleiner als eine bestimmte untere Grenze geworden ist. In diesem Fall wird der Reflexionsvorgang mit dem nächst schlechtesten

Punkt wiederholt. Kann für keinen Punkt des Complex ein zulässiger Reflexionspunkt gefunden werden, bricht das Verfahren vorzeitig ab. Ein zulässiger Reflexionspunkt wird daraufhin untersucht, ob mit ihm ein verbesserter Wert der Zielfunktion erzielt wird. Ist dies der Fall, wird das Abbruchkriterium überprüft; ist es erfüllt, terminiert das Verfahren regelmäßig, ansonsten wird der schlechteste Punkt ermittelt und die nächste Iteration des Optimierungsprozesses gestartet.

2.2 Hybridisierungsarten

Die Hybridisierung beruht auf folgenden vier Ansätzen:

- **Initialisierung der Startpopulation**
Die Startpopulation wird ganz oder teilweise mit den Ergebnissen von Läufen des lokalen Verfahrens initialisiert. Ein eventueller Rest wird zufällig bestimmt. Der Grundgedanke besteht darin, die evolutionäre Suche bereits mit Lösungen einer gewissen Qualität beginnen zu lassen und so die Zeit zum Auffinden befriedigender Bereiche des Suchraums zu sparen.
- **Nachoptimierung der GLEAM-Ergebnisse**
Die Nachoptimierung der GLEAM-Ergebnisse mit konventionellen Verfahren wird häufig auch als „*Local Hill Finding + Local Hill Climbing*“ bezeichnet. Man benutzt den EA zum Auffinden einiger vielversprechender Regionen des Suchraums und verwendet anschließend ein lokales Verfahren zur Bestimmung des exakten (lokalen) Optimums. Das beste dieser Optima ist dann die Lösung. Auf die Frage, wann der EA sinnvoll beendet und die lokale Suche begonnen werden kann, wird im nächsten Abschnitt eingegangen.
- **Direkte Integration**
Hierbei werden alle oder nur der beste Nachkomme einer Paarungsoperation von GLEAM mit dem lokalen Verfahren optimiert. Damit operiert GLEAM quasi nur noch über den Bergspitzen der Fitnesslandschaft.
- **Verzögerte Direkte Integration**
Bei vielen Aufgabenstellungen bringt eine lokale Optimierung ausgehend von

schlechten Startwerten meist nicht viel. Daher erscheint es sinnvoll, die direkte Integration erst nach einer anfänglichen Phase evolutionärer Suche dazuzuschalten.

Zu diesen vier Hybridisierungsarten kommen noch die Kombinationen von Voroptimierung und (verzögerter) direkter Integration. Bei der Nachoptimierung mit dem Complex wurden zwei Arten untersucht: Entweder wurden alle von GLEAM gelieferten Lösungen zur Bildung eines einzigen Start-Complexes genutzt oder sie dienten einzeln als Startpunkte für entsprechend viele Complex-Läufe. Daraus ergaben sich mit den beiden lokalen Verfahren 13 verschiedene Hybridisierungsarten.

2.3 Steuerung der Hybridisierung

Die auf dem OMID-Statusseminar [3] bereits ausführlich dargestellten Verfahren zur Steuerung der Hybridisierung beruhen auf der genotypischen Varianz in der Population. Diese wird nur bei auftretender Fitness-Stagnation bestimmt, um Rechenzeit zu sparen. Wenn es in der Population weniger als N_{max} Nischen von Individuen gibt, deren genotypische Differenz geringer als ein vorgegebenes ϵ ist und alle Nischen eine geringere Differenz als ϵ_{Pop} aufweisen, gilt die Population als „konvergiert“. Dann wird die Evolution bei der Nachoptimierung abgebrochen und alle Nischen

steuern einen Vertreter zum Gesamtergebnis bei, das noch um den Populationsbesten ergänzt wird, sofern er besser als alle Nischenvertreter ist. Diese Individuen werden mit dem jeweiligen lokalen Verfahren nachoptimiert und das beste ist schließlich das Endergebnis.

Im Falle der verzögerten direkten Integration wird das Konvergenzkriterium weniger streng als bei der Nachoptimierung parametrisiert. Es bestimmt den Zeitpunkt der Zuschaltung der lokalen Suche.

3 Testfälle

Untersucht wurden zusätzlich zu den in DEMIS [1] bereits benutzten Testfällen noch die Testfunktion nach Fletcher und Powel [9, Probl. 2.13] und die im Statusseminar [3] bereits behandelten Anwendungen aus dem laufenden Projekt. Tab. 1 zeigt wichtige Eigenschaften der Testfälle und gibt die Ergebnisse für die einzelnen Verfahren an. Alle Experimente wurden mit vergleichbaren Einstellungen der beteiligten Verfahren durchgeführt. Variiert wurden wie in [3] dargestellt lediglich beim Rosenbrock die Abbruchschranke und bei GLEAM die Populationsgröße. Abweichungen von den in [1] publizierten Ergebnissen erklären sich bei den beiden lokalen

Experiment	Parame- -teranz.	Modalität	Implizite Restrikt.	Genau- igkeit	Erfolgsrate [%] und Aufwand [Evaluationen]					
					GLEAM		Rosenbrock		Complex	
Bracken	2	multimod.	ja	1,0E-03	100	2509	4	1074	70	61
				1,0E-06	100	5000	-	-	98	196
Beale	3	unimod.	ja	1,0E-03	100	28936	37	4854	100	164
				1,0E-06	100	76229	2	1538	98	269
Bigg	5	multimod.	nein	1,0E-03	100	3370501	17	4001	20	164
				1,0E-06	-	-	18	18544	10	799
Helical Valley	3	unimod.	nein	1,0E-06	100	47288	71	15062	-	-
Rastrigin	20	multimod.	nein	1,0E-04	100	51721	-	-	-	-
Fletcher & Powel	5	multimod.	nein	1,0E-05	100	483566	10	1090	10	243
Fraktale Fkt.	20	multimod.	nein	Ziel: -0.05	100	195100	-	-	-	-
Heterodyn-Empf.	3	multimod.	nein		100	5773	8	94	12	102
Drucksensor	3	multimod.	nein		100	29791	-	-	-	-
Aktorplatte	dynam. n x 4	multimod.	ja							

Tabelle 1: Wichtige Eigenschaften der Testfälle und Ergebnisse der drei Verfahren

Verfahren durch den Verzicht, weitere Parameter wie z.B. die initiale Schrittweite anzupassen. Bei der Testfunktion nach Bracken wurde bei GLEAM (GADO) seinerzeit versehentlich der Aufwand zum Auffinden des exakten Optimums anstelle einer 10^{-3} -Genauigkeit angegeben.

In den drei Ergebnisspalten von Tab. 1 wird die Erfolgsrate in Prozent und der durchschnittliche Aufwand in Evaluationen basierend auf jeweils 100 Läufen für die jeweils beste Parametrierung angegeben. Ein – bedeutet, dass das betreffende Verfahren nicht in der Lage war, das Problem unter den gegebenen Einstellungen zu lösen. Beim Rosenbrock wurden folgende normierte Abbruchschranken verwendet: 0.1 (s), 0.01 (n), 10^{-4} (m), 10^{-6} (h) und 10^{-8} (x) und bei GLEAM wurde die Populationsgröße variiert. Ab der fraktalen Funktion sind in der Tabelle keine Genauigkeitsangaben vorhanden, da die exakte Lösung nicht bekannt ist. Bei der Aktorplatte handelt es sich auf Grund Ihrer aufwendigen Simulation um eine „echte Anwendung“ in dem Sinne, dass keine Läufe zum Verfahrensvergleich möglich sind, sodass hier nur die Eigenschaften gelistet sind.

Der Heterodynempfänger [2,13] ist eine Komponente eines mikrooptischen Kommunikationssystems bestehend aus zwei Kugellinsen und einer Photodiode. Der Abstand zwischen dem Lichtleiter und erster Kugellinse sowie die beiden Brechungsindizes sollen so bestimmt werden, dass nicht nur eine optimale Ausleuchtung der Photodiode im Kollimationspunkt erreicht wird, sondern die ganze Anordnung auch möglichst unempfindlich gegenüber Fertigungstoleranzen innerhalb vorgegebener Grenzen ist. Letzteres macht die ansonsten einfache Aufgabe zu einem trotz der nur drei Parameter schwierigen, stark multimodalen Optimierungsproblem.

Die Wandlung der mechanischen Größe Druck in ein elektrisch messbares Signal erfolgt bei dem betrachteten Drucksensorsystem [2] nach dem kapazitiven Wandlerprinzip. Der Drucksensor stellt eine Kondensatoranordnung dar, die aus der Sensormembrane (bewegliche Elektrode), der darunter liegenden Druckkammer und der implantierten n-Wanne (feste Elektrode) besteht. Bei der Designopti-

mierung des Drucksensors wurden zwei Kriterien berücksichtigt: die Sensitivität und die Linearität im Betriebsbereich. Optimierungsparameter sind die Kammerhöhe und –länge sowie die Membrandicke.

Der in Bild 4 dargestellte Mikroaktor [14] besteht aus zwei übereinander geschichteten dünnen Kunststofffolien, in denen mit einem Excimerlaser für die Aktorplatte, Aktorkammer und fluidischen Anschlüssen die entsprechenden Mikrostrukturen eingebracht wurden. Durch eine entsprechende Anzahl, Anordnung und Dimensionierung der ringförmigen und konzentrisch angeordneten Kerben kann das mechanische Verhalten der Aktorplatte gezielt beeinflusst werden. Ein wichtiges Optimierungsziel bei dem Kerbendesign besteht darin, bei einem gegebenen Arbeitsdruck eine möglichst große Auslenkung der Aktorplatte zu erreichen. Da die Aktorplatte gleichzeitig eine lange Lebensdauer haben soll, besteht das zweite Optimierungsziel darin, die Spannung in der Aktorplatte bei Auslenkung möglichst gering zu halten.

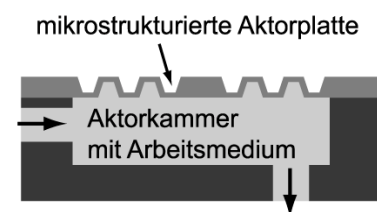


Abb.4: Schematische Darstellung der Aktorplatte

Die hier vorliegende Optimierungsaufgabe beschränkt sich somit nicht auf eine rein geometrische Optimierung, sondern sie umfasst zusätzlich die Optimierung der Topologie in Form von Anzahl und Anordnung der Kerben. Dies führt zu einem Parametervektor dynamischer Länge (pro Kerbe 4 Parameter), wodurch die meisten deterministischen und stochastischen Optimierungsverfahren nicht mehr anwendbar sind. Auch die Implementierungen vieler EAs erlauben im Gegensatz zu GLEAM keine Chromosomen dynamischer Länge, obwohl die Methode dies zulässt. Durch die diskrete Anzahl der Kerben ist der Lösungsraum zudem multimodal. Aufgrund der Genauigkeitsanforderungen und der Topologievariabilität ist der Einsatz von FEM-Modellen bei der Bewertung einzelner Lösungsvorschläge unumgänglich. Dies führt auf

einer Sun-Workstation (Ultra Sparc 10) zu Evaluationszeiten zwischen 3 und 10 Minuten. Die Optimierungen wurden daher mit einer Parallelversion von GLEAM durchgeführt [15]. Diese Ergebnisse dienen als Referenzwerte für die hier vorgestellten Resultate der hybriden Optimierung.

4 Ergebnisse

Die auf dem Statusseminar präsentierten Ergebnisse [3] und die Resultate weitergehender Untersuchungen [5] führen hinsichtlich der Hybridisierungsarten zu folgenden Schlussfolgerungen:

1. Die Vorooptimierung bringt bei einigen Testfällen eine gewisse Verbesserung, wobei einmal der Complex und ein anderes mal der Rosenbrock besser abschneidet.
2. Die Nachoptimierung verbessert zwar in der Regel die Ergebnisse der Evolution, erreicht aber lange nicht die Zuverlässigkeit bei der Lösungsfindung wie GLEAM.
3. Die (verzögerte) direkte Integration des Rosenbrock findet immer das Optimum (**GR** und **GvR**). Bei Verwendung des Complex (**GC** und **GvC**) wird meist weniger Aufwand benötigt als beim Rosenbrock, sofern die Complex-Integration als solche funktioniert.

Daher werden im folgenden nur die verzögerte und unverzögerte direkte Integration behandelt und ihre Resultate verglichen.

4.1 Ergebnisse der Testfunktionen

Bei der direkten Integration wurde Lamarck'sche Evolution verwendet, das heißt, dass der lokal verbesserte Nachkomme im Falle seiner Akzeptanz auch genotypisch an das gefundene (lokale) Optimum angepasst wurde. Bei der verzögerten direkten Integration werden die in Tab.2 angegebenen drei Parametrierungen für die Nischenbildung verglichen:

Parametrierung	ϵ	ϵ_{Pop}
P1	0.005	0.01
P2	0.002	0.005
P3	0.001	0.003

Tabelle 2: Steuerparameter

N_{max} variiert in Abhängigkeit von der Populationsgröße P: 2 für $P < 30$, 3 für $P = 30$, 4 für $P = 50$ und 5 für $P > 50$. Um die Ergebnisse der lokalen Verfahren mit GLEAM und den HyGLEAM vergleichen zu können, muss die Anzahl der Läufe bestimmt werden, bei der eine vergleichbare Sicherheit besteht, den Zielwert zu finden. Angesichts eines Stichprobenumfangs von 100 mag eine Sicherheit von 95,5% genügen. Damit ergeben sich die in Tab. 3 dargestellten Resultate.

Testfunktion	Genauigkeit	GLEAM	Aufwand bei 99,5% Sicherheit		HyGLEAM			
			Rosenbrock	Complex	GR	GvR	GC	GvC
Bracken	10^{-3}	2 509	139 620	305	31 264	21 834	439	545
	10^{-6}	5 000	-	392	314 834	546 499	1 433	1 785
Beale	10^{-3}	28 936	58 248	164	14 793	18 088	811	897
	10^{-6}	76 229	229 162	538	48 300	48 613	1 319	1 419
Bigg	10^{-3}	3 370 501	120 030	3 936	300 247	5 675 628	24 493	136 190
	10^{-6}	-	500 688	40 749	556 809	7 663 112	30 143	127 627
Helical Valley	10^{-6}	47 288	75 310	-	67 950	8 118	2 576	2 674
Rastrigin	10^{-4}	51 721	-	-	43 864	47 378	-	4 267 176
Fletcher&Powel	10^{-5}	483 566	55 590	12 393	13 535	10 206	4 684	4 635
Fraktale Fkt.		195 100	-	-	30 626	30 473	56 287	952 620

Tabelle 3: Vergleich der benötigten Individuen

(jeweils 100 von 100 Läufen erfolgreich oder hochgerechnet auf 95,5% Erfolgswahrscheinlichkeit)

Es fällt auf, dass der Complex in den Fällen, in denen er konvergiert, das beste Ergebnis der drei Verfahren liefert. Tab. 4 und Abb. 5 vergleichen die erreichten Verbesserungen gegenüber GLEAM: kein Eintrag bedeutet, dass die Aufgabe nicht gelöst werden konnte; eine Null bzw. ein flaches Feld, dass zur erfolgreichen Lösung mehr Aufwand notwendig war. Die Testfunktionen sind entsprechend der besseren Übersichtlichkeit im Diagramm

sortiert. Die geforderte Genauigkeit des Zielwertes ist in Klammern angegeben. Bigg (6) wurde in der Tabelle weggelassen, da ein Vergleich mit GLEAM nicht möglich ist. Für die Hybridisierung ist der Fall aber insofern ein Erfolg, als dass die direkte Complex-Integration am besten abschneidet. In Abb. 5 musste Bigg (3) aus Gründen der Darstellbarkeit weggelassen werden.

Test-Funktionen	Verbesserung gegenüber GLEAM						Parametrierungen					
	Rosen.	Compl.	GR	GvR	GC	GvC	GLEAM	Rosen.	GR	GvR	GC	GvC
Rastrigin			1,2	1,1			5		5 m	5 m P3		
Fraktale F.			6,4	6,4	0	0	20		5 n	5 n P2	10	5 P2
Bracken (3)	0	8,2	0	0	5,7	4,6	20	m	5 n	5 n P2	5	5 P2
Bracken (6)		12,8	0	0	3,5	2,8	10		5 n	5 n P2	5	5 P2
Hel.Valley	0		0	5,8	18,4	17,7	5	m	5 m	10 m P2	5	5 P2
Fletcher&P.	8,7	37,4	35,7	47,4	103,3	104,2	600	m	10 m	5 m P2	5	5 P3
Beale (3)	0	176,4	2,0	1,6	35,7	32,3	10	x	5 n	5 n P2	5	5 P2
Beale (6)	0	141,7	1,6	1,6	57,8	53,7	10	x	5 h	5 x P2	5	5 P2
Bigg (3)	28,1	856,3	11,2	0	137,6	24,7	500	h	30 m	20 m P2	20	5 P2

Tabelle 4: Verbesserung gegenüber GLEAM und beste Parametrierungen
 (Parametrierungen: Populationsgröße, Rosenbrock-Abbruchschranke, Nischenparametrierung (s. Tab.2))

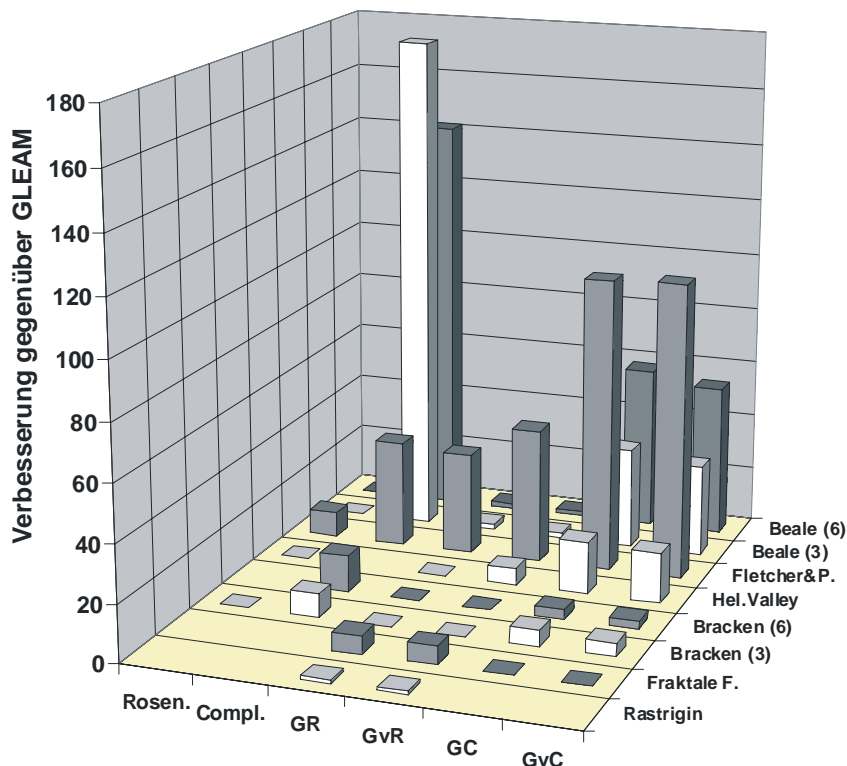


Abb.5: Vergleich der erreichten Verbesserungen des jeweils besten Laufs gegenüber GLEAM.
 (kein Eintrag: Zielwert nicht erreicht, flacher Eintrag: Zielwert bei höherem Aufwand als GLEAM erreicht)

Man kann feststellen, dass die hinsichtlich

Parameteranzahl und Suchraumeigenschaften

einfacheren Funktionen Bracken, Beale und Bigg am besten durch den Complex gelöst werden und die breiter angelegte globale Suche von GLEAM zu aufwendig für das Problem ist oder im Falle von Bigg GLEAM ohne Unterstützung des Complex-Verfahrens kaum eine Chance hat. Bei den schwierigeren Problemen versagt bis auf Fletcher & Powel der Complex und die Hybridisierung schneidet regelmäßig am besten ab. Allerdings arbeitet GLEAM dabei mit wechselnden Partnern. Betrachtet man nun die Parametrierungen der erfolgreichsten Läufe auf der rechten Seite von Tab. 4, so kann man folgendes feststellen: Alle hybriden Varianten kommen mit kleinen Populationsgrößen zwischen 5 und 30 aus, die Abbruchschranke des Rosenbrock-Verfahrens nimmt alle Werte zwischen n (niedrig) und x (sehr hoch) an und die Nischenparametrierung P2 dominiert. Es gibt also keine allgemein-günstige Parametrierung. Insbesondere ist unklar, welches lokale Verfahren besser geeignet ist und im Falle des Rosenbrock, mit welcher Abbruchschranke gearbeitet werden soll. Darauf wird im letzten Abschnitt dieses Kapitels noch weiter eingegangen werden.

4.2 Ergebnisse der Designoptimierungen

Der Vergleich der verschiedenen Verfahren und Parametrierungen erfolgt auf der Basis der zur Erreichung einer bestimmten Designqualität notwendigen Evaluationen. Da die Ergebnisse bereits auf dem Statusseminar [3] ausführlich dargestellt wurden, wird hier nur eine Zusammenfassung gegeben.

4.2.1 Heterodynempfänger

Wegen des relativ hohen Simulationsaufwands wurden jeweils nur 50 Läufe durchgeführt. Die lokalen Verfahren erreichen das Qualitätsziel mit unterschiedlicher Zuverlässigkeit, was in Tab. 5 mit den sich daraus ergebenden Aufwänden für eine Sicherheit von 99,5% dargestellt ist. GLEAM erreicht ab einer Populationsgröße von 150 zuverlässig die Zielqualität, wobei der niedrigste Aufwand bei einer Populationsgröße von 180 entsteht, siehe Abb.6.

Verfahren	Rosen. Abbr.	Erfolg [%]	Aufwand [Eval.]	95,5% Sicherheit:	
				Läufe	Eval.
Rosenbrock	n	8	94	64	6 016
	m	6	232	86	19 952
	h	30	764	15	11 460
Compl.		12	102	42	4 284

Tabelle 5: Ergebnisse der lokalen Verfahren

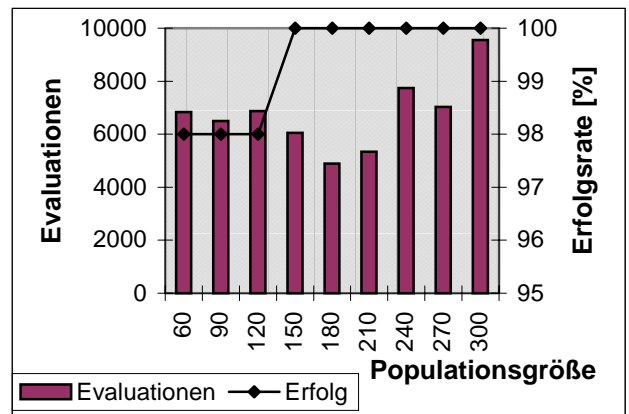


Abb.6: Aufwand und Erfolgsrate von GLEAM

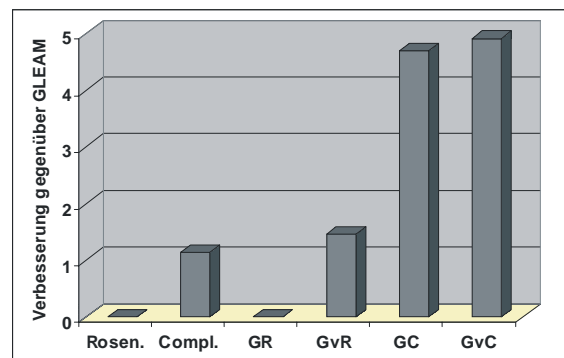


Abb.7: Vergleich der erreichten Verbesserungen des jeweils besten Laufs gegenüber GLEAM

Abb. 7 vergleicht die erreichten Verbesserungen gegenüber GLEAM basierend auf den jeweils besten Läufen der Hybriden und der berechneten Werte für die lokalen Verfahren. Alle Hybriden arbeiten mit einer Populationsgröße von 5 und die günstigste Hybridisierung GvC mit der Nischenparametrierung P3.

4.2.2 Drucksensor

Die Ergebnisse basieren auf 100 Läufen pro Parametrierung und Verfahren. Bei der hier benutzten strengeren Bewertung als in [2] erreichen die beiden lokalen Verfahren das Optimierungsziel in maximal 3% der Fälle (Rosenbrock) bzw. gar nicht (Complex). Bei GLEAM wird erst ab einer Populationsgröße

von 210 das Qualitätsziel zuverlässig erreicht und ab 360 auch bei niedrigen Evaluationsraten.

Die direkte Integration des Rosenbrock-Verfahrens mit Abbruchschranke s bringt hingegen stabile Verhältnisse, siehe Abb. 8. Der Aufwand ist mit dem der besten GLEAM-Läufe über einen weiten Bereich der Populationsgröße vergleichbar. Am besten schneidet eine Begrenzung der Iterationen des Rosenbrock auf 100 ab (Ros-100), während bei 250 (Ros-250) etwas mehr Evaluationen bei gleichem Erfolg benötigt werden. Andere Abbruchschranken führen zu deutlich schlechteren Ergebnissen. Die Integration mit dem Complex führt hier hingegen zu einem Mehraufwand.

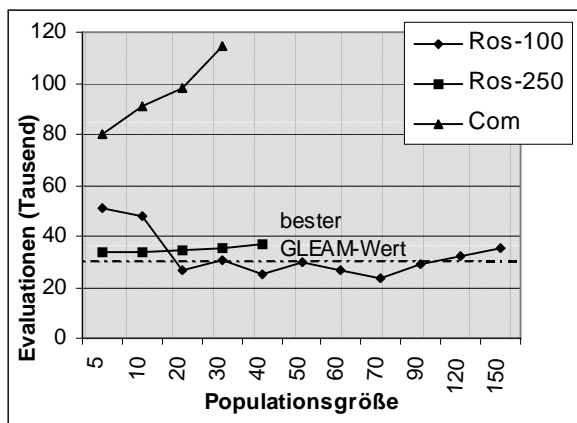


Abb.8: Direkte Integration

4.2.3 Aktorplatte

Bei der Aktorplatte verbieten sich auf Grund der langen Simulationszeiten vergleichende Läufe mit unterschiedlichen Parametrierungen. Stattdessen sind die bisher gemachten Erfahrungen praktisch anzuwenden. Verglichen werden Ergebnisse, die mit einer Parallelversion von GLEAM erzielt wurden [15], mit HyGLEAM unter Verwendung des Rosenbrock-Verfahrens bei direkter Integration und mittlerer Abbruchschranke, wobei die Hälfte der Startpopulation vorgeneriert und voroptimiert wurde [14]. Tab. 6 zeigt die Werte eines anfänglichen manuellen Designs und die der beiden Optimierungsergebnisse. Beim Vergleich des Aufwands ist folgendes zu beachten: Die Komponente zur Ankopplung des FEM-Simulators speichert die simulierten Parametervektoren zusammen mit den Ergebnissen ab und vergleicht neue Anforderungen mit den bereits errechneten, um so Simulationszeit zu sparen. Dies kommt besonders beim Rosenbrock-Verfahren zum Tragen, da in der Konvergenzphase relativ häufig sehr ähnliche Vektoren berechnet werden, die vom Filter als identisch bewertet werden. Dementsprechend wird in der Tabelle zwischen angeforderten und ausgeführten Simulationen unterschieden. HyGLEAM benötigt also nur etwa die Hälfte an real durchgeführten Simulationen, um ein besseres Ergebnis zu erzielen als die nichthybride Variante. Dies führte zu einer Laufzeit von 2 Tagen und 15 Stunden auf 8 Suns unterschiedlicher Leistung (504 CPU-Stunden) bei GLEAM, während HyGLEAM mit 12 Tage und 16 Stunden (304 CPU-Stunden) auf einer Sun Ultra Sparc 10 auskam. Neuere Ergebnisse zu variierenden Drücken und E-Modulen sind in [16] im gleichen Band zu finden.

rungen mit den bereits errechneten, um so Simulationszeit zu sparen. Dies kommt besonders beim Rosenbrock-Verfahren zum Tragen, da in der Konvergenzphase relativ häufig sehr ähnliche Vektoren berechnet werden, die vom Filter als identisch bewertet werden. Dementsprechend wird in der Tabelle zwischen angeforderten und ausgeführten Simulationen unterschieden. HyGLEAM benötigt also nur etwa die Hälfte an real durchgeführten Simulationen, um ein besseres Ergebnis zu erzielen als die nichthybride Variante. Dies führte zu einer Laufzeit von 2 Tagen und 15 Stunden auf 8 Suns unterschiedlicher Leistung (504 CPU-Stunden) bei GLEAM, während HyGLEAM mit 12 Tage und 16 Stunden (304 CPU-Stunden) auf einer Sun Ultra Sparc 10 auskam. Neuere Ergebnisse zu variierenden Drücken und E-Modulen sind in [16] im gleichen Band zu finden.

	Auslenkung		Spannung		Evaluierungen	
	g [μm]	[%]	[N/mm ²]	[%]	angef.	ausgef.
manuell	104	100	12	10		
GADO	238	229	16	13	2897	1130
HyGAD	251	241	17	14	1228	5935
O				2	47	

Tab.6: Optimierungsergebnisse

4.3 Schlussfolgerungen

Wie in den beiden vorigen Abschnitten dargestellt, gibt es keine problemunabhängige günstige Parametrierung für die Hybridisierung: Bei einer neuen Aufgabe ist weder klar, welches der beiden lokalen Verfahren günstiger ist, noch mit welcher Abbruchschranke der Rosenbrock betrieben werden soll oder ob und wenn ja mit welcher Verzögerung die lokale Suche zuzuschalten ist. Für die Populationsgröße kristallisiert sich immerhin heraus, dass kleine Werte genügen. Sicherheits halber sollte man einen Wert von zehn und bei wahrscheinlich schwierigen Problemen von 20 verwenden. Für die anderen Einstellungen sind selbstadaptive Lösungen anzustreben. So kann die Aufteilung zwischen Rosenbrock und Complex ausgehend von

einer Gleichgewichtung durch die Bewertung von jeweils erreichtem Erfolg und dazu notwendigem Aufwand angepasst werden.

Man kann davon ausgehen, dass bei der Nachkommensverbesserung anfänglich auf das genaue (lokale) Optimum verzichtet werden kann und die Präzision und damit der Aufwand erst im Laufe der Evolution ansteigen brauchen. Dies kann bei beiden Verfahren durch eine Begrenzung der Iterationen erreicht werden, die im Laufe der Evolution gelockert wird. Beim Rosenbrock kommt noch die Abbruchschranke als Einstellmöglichkeit hinzu. Beide Parameter können wieder durch das Verhältnis von erreichter Verbesserung bei investiertem Aufwand gesteuert werden. Als weitere Steuergröße kommt noch die Nischenbildung in Betracht, mit der der Fortschritt der Suche beurteilt werden kann.

Als Konsequenz aus den durchgeführten Untersuchungen kann gesagt werden, dass eine Hybridisierung auf der Grundlage der (verzögerten) direkten Integration bei hinreichend schwierigen Problemen eindeutig den geringsten Aufwand an Evaluationen verursacht. Dabei ist es problemabhängig, wie viel gegenüber GLEAM oder einem der lokalen Verfahren im Falle ihrer Konvergenz eingespart werden kann. Es wurden Einsparungen bis zum Faktor 104 (Fletcher & Powel) und 137 (Bigg (3)) beobachtet. Bei nicht zu schwierigen Problemen empfiehlt es sich, auch aus Gründen des geringen Aufwands erst den Complex auszuprobieren.

5 Webbasiertes HyGLEAM mit ORBIT

5.1 ORBIT

Abb. 9 gibt einen Überblick über Remote-Anwendungen mit dem in [8] ausführlicher beschriebenen ORBIT-System. Es besteht aus den in Abb. 10 dargestellten drei Komponenten Internet Web-Server, Data-Server und ORBIT-Engine, die im folgenden kurz beschrieben werden.

Der Internet Web-Server stellt die externe Schnittstelle zum ORBIT-System dar. Er verfügt über dynamische Webseiten, die je nach Anwendung automatisch generiert werden.

Hierzu zählen auch HTML-Formulare, die der Anwender zur Konfiguration der Modelle benutzt. Nach der Durchführung der jeweiligen Anwendung können auf den Seiten des Internet Web-Servers die Ergebnisseiten abgerufen werden.

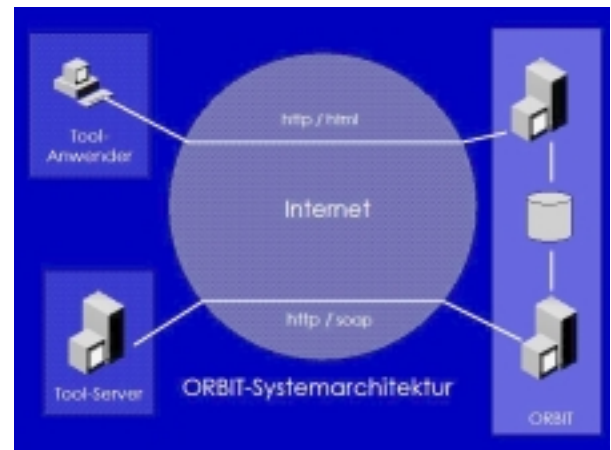


Abb.9: ORBIT-Systemarchitektur

Der Data-Server dient als Schnittstelle zwischen Internet Web-Server und ORBIT-Engine. In der Datenbank des Data-Servers werden Modell- und Ergebnisdaten sowie Laufzeitdaten von instanziierten Workflows vorgehalten. Der Data-Server verfügt über eine SOAP-Schnittstelle zur Generierung neuer Modellstrukturen und der Konfiguration von Tools bzw. Workflows.

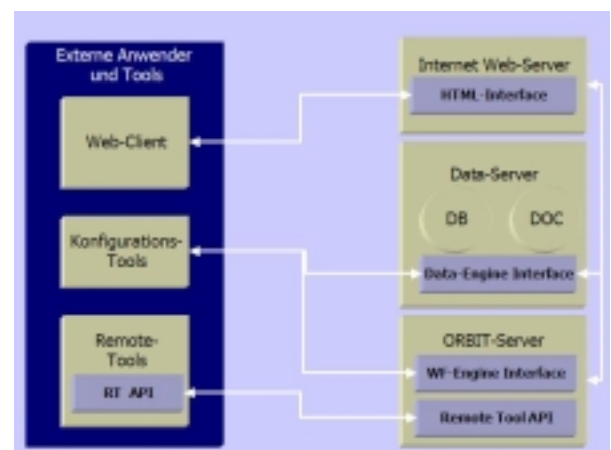


Abb.10: Komponenten des ORBIT-Systems

Die ORBIT-Engine stellt den eigentlichen Kern des Systems dar. Sie verwaltet Workflowdefinitionen und -instanzen und steuert den Fortschritt von laufenden Workflowinstanzen. Über die Remote-Tool-API werden externe Tools über SOAP in den Workflow-

prozess integriert und die Toolaufrufe gesteuert. Darüber hinaus verfügt die ORBIT-Engine über eine SOAP-basierte Service-Schnittstelle, mittels derer externe Tools Zugriff auf Konfigurationsdaten im ORBIT-System erhalten.

5.2 HyGLEAM in ORBIT

Zur Einbindung von HyGLEAM in das ORBIT-System wurde eine HyGLEAM-Engine ohne Benutzerinterface implementiert, deren Steuerung vollständig über die Aufrufparameter und Dateien erfolgt. Die HyGLEAM-Engine wird in einen eigenen in Java implementierten Tool-Agenten eingebettet, der die Kommunikation mit der ORBIT-Engine übernimmt. Abb. 11 zeigt das Einstiegsformular

des webbasierten Aufrufs eines Optimierungslaufs. Zuerst ist eine vorkonfigurierte Aufgabenstellung auszuwählen und dann das Optimierungsverfahren zu bestimmen. Je nach Wahl des Verfahrens gelangt man dann zu einem Unterformular, mit dessen Hilfe man die notwendige Parametrierung vornehmen kann, wobei für alle Eingabefelder Defaultwerte angeboten werden. In Abb. 12 ist das Formular für GLEAM selbst dargestellt während Abb. 13 die Parametrierung der Hybridisierungsart „Voroptimierung“ zeigt. Während eines Laufs kann der Benutzer Zwischenergebnisse abfragen und gegebenenfalls den Lauf abbrechen. Am Ende eines Laufs erhält er die Ergebnisdaten.

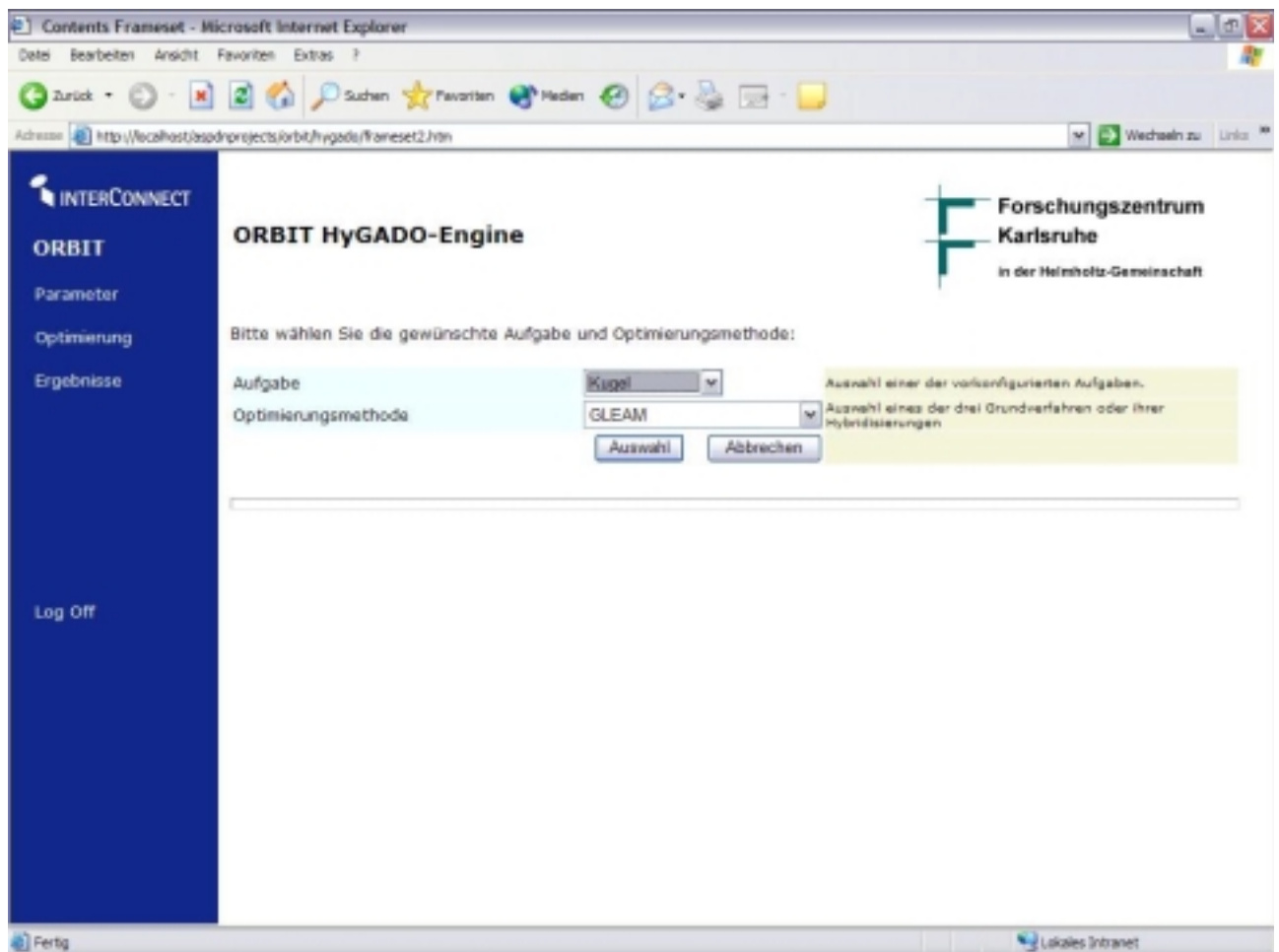


Abb.11: Einstiegsformular für einen Optimierungslauf

The screenshot shows a web browser window titled "Contents Frameset - Microsoft Internet Explorer" with the address bar displaying "http://localhost/ispd/projects/orbit/hygado/frameset1.htm". The page features a blue sidebar on the left with the "INTERCONNECT" logo and a menu containing "ORBIT", "Parameter", "Optimierung", "Ergebnisse", and "Log Off". The main content area is titled "ORBIT HyGADO-Engine" and includes the logo of the "Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft". Below the title is the "Unterformular GLEAM" section, which contains a table of parameters for configuration:

Populationsgröße	128	5..1000
Dauer	1440	Dauer in Minuten, > 0
Zielnote	1000000	Zielqualität, 0..1000000
Generationslimit	5000	Max. Generationsanzahl, > 0
Generationen ohne lokale Verbesserung	500	Stagnationsindikator bei dem d. Lauf vorzeitig angebrochen wird, > 0
Generationen ohne lokale Akzeptanz	50	Stagnationsindikator bei dem d. Lauf vorzeitig angebrochen wird, > 0

At the bottom of the form are two buttons: "Auswahl" and "Abbrechen".

Abb.12: Unterformular zur Parametrierung eines GLEAM-Laufs

The screenshot shows the same web browser window as in Abb.12, but the main content area is titled "Unterformular Voroptimierung". The parameters are as follows:

Populationsgröße	128	5..1000
Dauer	1440	Dauer in Minuten, > 0
Zielnote	1000000	Zielqualität, 0..1000000
Generationslimit	5000	Max. Generationsanzahl, > 0
Generationen ohne lokale Verbesserung	500	Stagnationsindikator bei dem d. Lauf vorzeitig angebrochen wird, > 0
Generationen ohne lokale Akzeptanz	50	Stagnationsindikator bei dem d. Lauf vorzeitig angebrochen wird, > 0
Lokales Suchverfahren	Rosenbrock	Verfahren
Rosenbrock-Präzision	niedrig	Rosenbrock-Abbruchkriterium
Voroptimierungsanteil	20	Prozentanteil voroptimierender Individuen der Startpopulation

At the bottom of the form are two buttons: "Auswahl" and "Abbrechen".

Abb.13: Unterformular zur Parametrierung der Hybridisierungsart „Voroptimierung“

6 Zusammenfassung

Die wichtigsten Ergebnisse der umfangreichen Untersuchungen zum Verhalten globaler und lokaler Suchverfahren basierend auf dem Evolutionären Algorithmus GLEAM, dem Rosenbrock-Algorithmus und dem Complex-Verfahren unter Verwendung mathematischer Benchmarkfunktionen und realer Designprobleme sind, dass

1. es kein generell günstigstes Verfahren gibt,
2. bei einfacheren Aufgaben mit geringer Parameteranzahl die lokale Verfahren mit weniger Aufwand zum Ziel führen als das global arbeitende GLEAM,
3. ab einem gewissen Schwierigkeitsgrad die Hybridisierung in Form der direkten Integration zuverlässig und mit vergleichsweise geringem Aufwand zum Ziel führt,
4. hinsichtlich einer adaptiven Steuerung der Rechenzeitaufteilung zwischen globaler und lokaler Suche und der adaptiven Auswahl des geeigneten lokalen Verfahrens weiterer Forschungs- und Entwicklungsbedarf besteht, siehe Abschn. 4.3.

Mit der webbasierten Version von HyGLEAM wird eine Remote-Optimierung und damit eine wirtschaftlichere Nutzung der Simulationswerkzeuge durch einen Application Service Provider ermöglicht.

Literatur

- [1] D. Peters, M. Gorges-Schleuter, W. Jakob, S. Parodat: *Modellbasierte Optimierung mit mathematischen Methoden – ein Vergleich von klassischen und evolutionären Verfahren*. 7. GMM-Workshop Methoden und Werkzeuge zum Entwurf von Mikrosystemen. Paderborn, 1999.
- [2] W. Jakob, B. Knorr, S. Parodat, D. Peters, A. Uhlig: *Optimierung von Mikrosystemen*. 8. GMM-Workshop Methoden und Werkzeuge zum Entwurf von Mikrosystemen. Berlin, 1999.
- [3] W. Jakob, D. Peters: *Designoptimierung mit HyGADO*. In: R. Laur, D. Peters (Hrsg.): OMID – Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen (Statusseminar), VDI/VDE, 2001.
- [4] W. Jakob: *HyGLEAM: Hybrid General purpose Evolutionary Algorithm and Method*. Proc. of 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI'2001), 2001.
- [5] W. Jakob: *HyGLEAM - An Approach to Generally Applicable Hybridization of Evolutionary Algorithms*. Proc. of PPSN VII, LNCS 2439, Springer-Verlag, Berlin, 2002.
- [6] C. Blume: *GLEAM - A System for Simulated Intuitive Learning*. Proc. of PPSN I, LNCS 496, Springer-Verlag, 1991.
- [7] C. Blume, W. Jakob: *GLEAM - an Evolutionary Algorithm for Planning and Control Based on Evolution Strategy*. In: E. Cantú-Paz (Hrsg.): GECCO 2002, Vol. Late-Breaking Papers, L. Livermor National Laboratory, 2002.
- [8] A. Reiffer, P. Rohnacher: ORBIT – Internetbasierte Plattform für Remote-Anwendungen. In: H. Eggert, A. Reiffer (Hrsg.): OMID – Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen (Abschlußseminar des BMBF-Verbundprojekts OMID), FZK, 2002.
- [9] H.-P. Schwefel: *Evolution and Optimum Seeking*. John Wiley & Sons, Chichester, 1995.
- [10] C. Blume, W. Jakob: *Cutting Down Production Costs by a New Optimization Method*. Proc. of the Japan - U.S.A. Symposium on Flexible Automation, ASME, 1994
- [11] H. H. Rosenbrock: *An Automatic Method for Finding the Greatest or Least Value of a Function*, Comp. J. 3, 1960.
- [12] M. J. Box: *A New Method of Constrained Optimization and a Comparison with other Methods*, Comp. J. 8, 1965.
- [13] I. Sieber, H. Eggert, H. Guth, W. Jakob, K.-P. Scherer, P. Ziegler: *Design Optimization Considering Tolerance Effects of Microoptical Benches*. MicroSystem Technologies 98, VDE-Verlag, 1998.
- [14] A. Quinte: *Modellierung und Optimierung der Aktorplatte*. In: R. Laur, D. Peters (Hrsg.): OMID – Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen (Statusseminar), VDI/VDE, 2001.
- [15] W. Jakob, A. Quinte, K.-P. Scherer, H. Eggert: *Optimization of a Micro Fluidic Component using a Parallel Evolutionary Algorithm and Simulation Based on Discrete Element Methods*. S. Hernández, C.A. Brebbia (Eds.): Computer Aided Optimum Design of Structures VII, WIT Press, Southampton, 2001.
- [16] A. Quinte: *Modellierung und Optimierung der Aktorplatte*. In: H. Eggert, A. Reiffer (Hrsg.): OMID – Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen (Abschlußseminar des BMBF-Verbundprojekts OMID), FZK, 2002.

DESIGNOPTIMIERUNG VON MIKROSYSTEMEN MIT MODOS

D. Peters*, H. Bolte*, O. Nüssen*, W. Jakob**, A. Quinte**, A. Schneider***, R. Laur*

*Universität Bremen, ITEM, Postfach 33 04 40, 28334 Bremen

**Forschungszentrum Karlsruhe, Institut für Angewandte Informatik

*** Fraunhofer Institut für Integrierte Schaltungen, Dresden

Email: peters@item.uni-bremen.de

Kurzfassung

Die Designoptimierung von Mikrosystemen und deren Komponenten ist eine vielfältige Aufgabe, die nicht nur flexible Werkzeugunterstützung erfordert, sondern insbesondere auch die Entwicklung leistungsstarker Optimierungsstrategien erforderlich macht. In diesem Beitrag wird das Optimierungswerkzeug MODOS mit den im Rahmen dieses Forschungsvorhabens entwickelten angepassten Optimierungsstrategien anhand verschiedener Anwendungen vorgestellt.

1 Einleitung

Die Verfahren der Mikrostrukturierung werden in den verschiedensten technischen Bereichen zur Fertigung zuverlässiger und kostengünstiger Produkte eingesetzt. Aufgrund des Zusammenwirkens verschiedener physikalischer Domänen sowie aufgrund des wachsenden Integrationsgrades von Mikrosystemen nimmt die bei der Entwicklung zu beherrschende Systemkomplexität stark zu. Designer sind daher mehr und mehr auf die Unterstützung von CAD-Werkzeugen angewiesen.

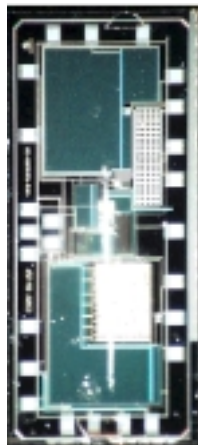
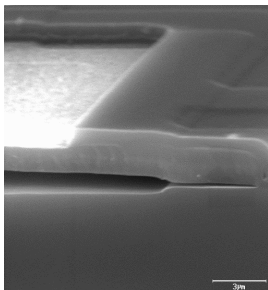


Abb. 1:
Mikrosystemtechnische
Komponenten

Abbildung 1 zeigt zwei typische mikrosystem-

technische Komponenten: links, im Querschnitt, einen mikromechanischen Drucksensor und rechts daneben einen für medizintechnische Anwendungen entwickelten Telemetriechip.

Beide Komponenten sind Bestandteile medizintechnischer Produkte zur Messung des Drucks in verschiedenen Körperbereichen: z.B. Blutdruck, Druck im Inneren des Auges, Druck im Kopf. Bei jeder dieser Anwendungen ist der Sensor in den Körper des Menschen zu implantieren. Die Messsignale sind mit einer Signalvorverarbeitung so aufzubereiten, dass sie von der Telemetrieinheit aus dem Körperinneren in Form eines digitalen Datenwortes drahtlos nach außen gesendet werden können. Gleichzeitig übernimmt die Telemetrieinheit die Energieversorgung des intelligenten Sensorchips.

Neben einer stabilen Energieversorgung müssen die Messsignale des Sensors möglichst fehlerfrei übermittelt werden. Die geforderte Auflösung des Sensors ist zu realisieren und der maximal zulässige Leistungsverbrauch darf nicht überschritten werden. Allein diese Teilaspekte des Entwurfs des hier prototypisch vorgestellten Mikrosystems verdeutlichen die Entwurfskomplexität solcher Systeme. Üblicherweise wird ein modularer Ansatz gewählt, der das System in technisch weitgehend abgeschlossene Blöcke unterteilt. Diese werden unter Zuhilfenahme von Modellierung und Simulation zunächst „virtuell“ entwickelt. Soweit als möglich wird die Funktionalität des Gesamtsystems entsprechend der Spezifikation, basierend auf Simulationen, sichergestellt. Anschließend werden erste Prototypen gefertigt.

Doch auch dieser modulare Ansatz wird mit wachsender Integrationsdichte zunehmend komplizierter und schwieriger zu handhaben. Designer benötigen neben Modellierung und Simulation weitere Entwurfsunterstützungen,

mit denen aus der Vielzahl existierender Designvarianten diejenigen ausgewählt werden können, die die Spezifikation bestmöglich erfüllen. Derartige Entwurfsunterstützungen können durch Verfahren der modellbasierten Designoptimierung geliefert werden.

Ein Vorteil der modellbasierten gegenüber beispielsweise der auf symbolischen Berechnungen basierenden Designoptimierung besteht darin, dass sie sich nahtlos in den heute bereits etablierten Designprozess für Mikrosysteme einbinden lässt. Der Entwickler benötigt lediglich ein Werkzeug zur Optimierung, das mit beliebigen Simulatoren gekoppelt werden kann oder bereits gekoppelt ist. Problematisch dabei ist lediglich, dass die einzelnen Simulationen, die zur Bewertung eines Satzes von Designparametern benötigt werden, sehr zeitintensiv sein können. Des Weiteren liegt die meist nichtlineare Zielfunktion nicht in analytischer Form vor. Es sind also neben einem flexiblen Optimierungswerkzeug leistungsfähige Optimierungsalgorithmen zur Verfügung zu stellen, die möglichst sicher die beste Designalternative ermitteln können und gleichzeitig wenig Simulationen für die Suche benötigen.

In diesem Beitrag werden Ergebnisse der Arbeiten aus dem Forschungsvorhaben „Modellbasierte Designoptimierung von Mikrosystemen“ zur Weiterentwicklung eines Werkzeuges für die modellbasierte, rechnergestützte Designoptimierung (MODOS) präsentiert.

Zielsetzungen dieses Forschungsvorhabens hinsichtlich der Werkzeugentwicklung waren:

- die Entwicklung von an die Problemstellungen der Mikrosystemtechnik angepassten Optimierungsalgorithmen;
- die Entwicklung von hybriden Optimierungsalgorithmen;
- die Kopplung des am ITEM entwickelten Optimierungswerkzeugs MODOS an diverse Simulatoren;
- die Einbindung von MODOS in das verteilt arbeitende, web-basierte System MOSCITO [1].

In Abschnitt 2 wird exemplarisch die Anwendung des modellbasierten Designoptimierungssystems MODOS auf die Optimierung von Teilaspekten bei der Entwicklung des anfangs vorgestellten Mikrosystems dargestellt. Im Abschnitt 3 werden drei an die Erfordernisse der Mikrosystemtechnik angepasste Optimierungsverfahren vorgestellt. Abschnitt 4 und 5 stellen die Ergebnisse zur Optimierung zweier Anwendungen des Verbundprojektes OMID vor, bevor im Abschnitt 6 die Zusammenfassung folgt.

2 Exemplarische Anwendung von MODOS in der Mikrosystemtechnik

Das Modellbasierte Designoptimierungssystem MODOS (Abbildung 2) wurde speziell für die Belange des Designs von Mikrosystemen entwickelt. Im Folgenden wird das Konzept von MODOS wie auch eine seiner möglichen Anwendungen im Entwurfsprozess von Mikrosystemen erläutert.



Abb. 2: MODOS - GUI

2.1 Modellbasierte Designoptimierung von Mikrosystemen

Die bei der Entwicklung von Mikrosystemen auftretenden Fragestellungen sind vielfältig. Dazu zählen unter anderem Modellparametrisierungen (insbesondere von Verhaltens- oder Makromodellen), Machbarkeitsstudien, Komponenten- und Systemoptimierungen hinsichtlich der vorgegebenen Spezifikation. Es ist also nicht sinnvoll, ein an feste Charakteristika einer Designaufgabe angepasstes Optimierungsverfahren zu entwickeln. MODOS stellt daher eine Auswahl verschiedener ein-

und mehrdimensionaler, beschränkt sowie unbeschränkt arbeitender Algorithmen zur Verfügung. Die Formulierung der Bewertungsfunktion (Zielfunktion) bleibt dem Anwender überlassen. Dies kann von Nachteil sein, wenn nur wenig Vorkenntnisse im Bereich der Optimierung vorhanden sind, birgt jedoch den Vorteil in sich, dass MODOS auf beliebige Problemstellungen angewendet werden kann.

Des Weiteren wurde MODOS mit diversen Simulatoren gekoppelt, da bei der Entwicklung von Mikrosystemen sowohl feld- als auch netzlistenbasierte und Verhaltenssimulatoren eingesetzt werden. Prinzipiell kann jeder beliebige Simulator mit MODOS betrieben werden. Realisiert wurden bislang folgende Kopplungen: PSPICE, HSPICE, TITAN, SABER, MATHEMATICA. Da MODOS als sogenannte TCL/Tk-Applikation realisiert wurde, kann es mit anderen TCL/Tk-Applikationen wie beispielsweise SABER direkt über Techniken der Interprozesskommunikation verbunden werden [2].

Um dem Anwender die Nutzung von MODOS zu erleichtern, wurde eine interaktive Hilfefunktion in die graphische Oberfläche eingebaut.

Die Anwendung von MODOS zur Optimierung von Mikrosystemen wird anhand des in der Einleitung bereits diskutierten Systems exemplarisch erläutert. Es handelt sich dabei um eine medizintechnische Anwendung von der im Folgenden der Drucksensor mit Signalverarbeitung näher betrachtet wird.

2.2 Modellevaluation

Bevor Verhaltensmodelle mikrosystemtechnischer Komponenten zur Designoptimierung eingesetzt werden können, ist sicherzustellen, dass alle wesentlichen Effekte in der abstrakteren Modellvariante enthalten sind. Zu diesem Zweck werden Verhaltensmodelle üblicherweise manuell mit Werten parametrisiert, die aus FEM-Simulationen oder Messungen bekannt sind. Ein visueller Vergleich zwischen gemessenen bzw. FEM-Kennlinien und den entsprechenden Kennlinien der Verhaltenssimulation liefert den Nachweis über die

Gültigkeit des Modells. Aufgrund der Vielzahl von Parametern, den zwischen ihnen auftretenden Wechselwirkungen und den über einen Wafer verteilten Parameterschwankungen ist diese Vorgehensweise sehr aufwendig.

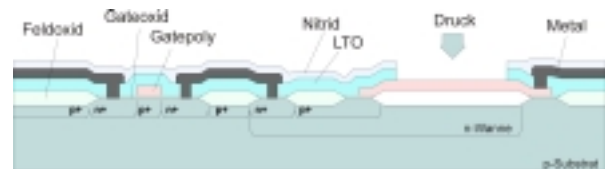


Abb. 3: Querschnitt des Drucksensors

Effektiver und auch zuverlässiger kann die Entwicklung eines Modells gestaltet werden, wenn Optimierungswerkzeuge wie MODOS unterstützend verwendet werden. Ausgehend von Messungen oder FEM-Simulationen wird die Kennlinie des Verhaltensmodells an die vorgegebene Kennlinie über Parametervariationen angepasst. Die Nominalwerte der Design- bzw. Prozessgrößen werden in physikalisch sinnvollen Grenzen vorgegeben, soweit sie bekannt sind. Kann das Modell die vorgegebene Kennlinie mit Abweichungen, die kleiner als die Messgenauigkeit sein sollten, nachbilden, und sind die Parameterwerte physikalisch sinnvoll gewählt worden, so bildet das Verhaltensmodell den Zusammenhang zwischen der gemessenen Größe und den Designparametern korrekt nach.

Anhand dieser Vorgehensweise wurde das Verhaltensmodell des Drucksensors mit MODOS evaluiert [3]. Verwendet wurden hierzu der Complex-Algorithmus [4], der ComplInt-Algorithmus [5], das Verfahren nach Rosenbrock [6] sowie der RCT-Algorithmus [7]. Um möglichst große Flexibilität zu bieten, wurde das Sensormodell in Form einer C-Funktion realisiert, die leicht in MODOS und in SABER eingebunden werden kann.

Ergebnis der Modellevaluation ist, dass der Zusammenhang zwischen der Sensorkapazität, den geometrischen Parametern sowie einigen Prozessparametern korrekt modelliert wurde und das Modell somit für Komponenten- wie auch Systemoptimierungen einge-

setzt werden kann.

2.3 Ableitung von Designvorgaben

Das Drucksensorsystem besteht aus einem kapazitiv arbeitenden Absolutdrucksensor (Abbildung 3) und einem SC-Oszillator zur Signalaufbereitung (Abbildung 4).

Die parasitären Kapazitäten C_{p1} am Eingang und C_{p2} am Ausgang des Sensors, die Verstärkung des Operationsverstärkers A_D sowie die Dimensionierung der Referenzkapazität C_R und der Feedbackkapazität C_F haben einen nicht zu vernachlässigenden Einfluss auf die Güte des Ausgangssignals. Sie führen zu einem Fehler Δu_a , der eine Veränderung des Pulsweitenverhältnisses und damit eine Verschlechterung der Auflösung nach sich zieht.

$$\Delta u_a = \Delta u_{a,ideal} - \frac{C_{p2}}{C_F} \cdot \frac{u_a}{A_D}$$

Um die vom Anwender geforderte minimale Auflösung realisieren zu können, ist der Einfluss der parasitären Kapazität C_{p2} zu minimieren. Dies kann erreicht werden, indem die Feedbackkapazität C_F bzw. die Verstärkung des Operationsverstärkers A_D möglichst groß gewählt werden. Zu beachten ist dabei, dass Kapazitäten sehr viel Chipfläche verbrauchen und deswegen nicht beliebig groß gewählt werden können. Da auch die Verstärkung des Operationsverstärkers nicht beliebig groß gewählt werden kann, ist genau das Design gesucht, das für bestimmte maximal zulässige parasitäre Kapazitäten die vom Anwender geforderte Auflösung erzielt.

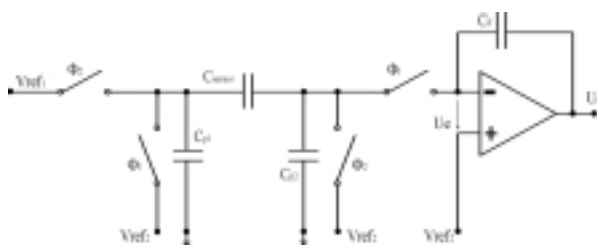


Abb. 4: SC-Oszillator

Beim manuell optimierten Nominaldesign konnte die geforderte Auflösung nicht erreicht werden. Die mit MODOS und SABER durchgeführten Designoptimierungen erzielen dage-

gen ein Design, dass bei einer Vergrößerung der Kantenlänge um ca. 15% und einer Verdoppelung der Verstärkung des Operationsverstärkers die Spezifikation erfüllt. Im nächsten Schritt wäre ein entsprechender Operationsverstärker zu entwickeln und das Sensordesign zu überarbeiten.

3 Entwicklung angepaßter Optimierungsverfahren

Die besondere Problematik der modellbasierten Designoptimierung besteht darin, daß Simulationsergebnisse zur Bewertung einer Designvariante benötigt werden, die Zielfunktionen also nicht in analytischer Form vorliegen. Da Simulationen in der Mikrosystemtechnik sehr zeitaufwendig sein können, sind für die modellbasierte Designoptimierung effiziente mathematische Suchverfahren zur Verfügung zu stellen. Sie sollten möglichst global konvergieren und möglichst wenige Iterationsschritte zur Ermittlung des Ergebnisses benötigen.

Neben der im Abschnitt 3.3 vorgestellten hybriden Strategie, die genetische Algorithmen mit direkten Suchverfahren kombiniert, wurden im Rahmen dieses Forschungsvorhabens zwei leistungsfähige stand-alone Verfahren implementiert: ein für beschränkte Problemstellungen entwickeltes Quasi-Newton-Verfahren sowie ein neuartiges Polyederverfahren. Quasi-Newton-Verfahren zeichnen sich in der Regel durch schnelle lokale Konvergenz aus, während das Polyederverfahren so konstruiert wurde, dass es lokal und global ausgerichtete Elemente bei der Suche miteinander kombiniert.

3.1 Quasi-Newton Verfahren

Quasi-Newton Verfahren zeichnen sich durch ihre schnelle Konvergenz aus. Sie weisen jedoch den Nachteil auf, dass sie in der Regel in das dem Startpunkt am nächsten gelegene lokale Optimum konvergieren. Dieses Verhalten resultiert daraus, dass zur Ermittlung der Suchrichtung die ersten sowie zweiten Ableitungen der Zielfunktion herangezogen werden.

Quasi-Newton Verfahren bilden im aktuellen Designpunkt ein quadratisches Modell der Zielfunktion mit Hilfe eines Taylorreihenansatzes zweiter Ordnung:

$$f(\mathbf{x}) \approx f(\mathbf{x}_k) + \mathbf{f}'(\mathbf{x}_k)^T (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_k)^T \mathbf{A} (\mathbf{x} - \mathbf{x}_k)$$

Die Suchrichtung für den nächsten Schritt wird über das Optimum dieses lokal gültigen Modells ermittelt; die Schrittlänge in diese Richtung bestimmt man unter Verwendung eines beliebigen eindimensionalen Liniensuchverfahrens.

Problematisch wird die Verwendung dieser Verfahren dann, wenn Beschränkungen bei der Optimierung zu berücksichtigen sind. Dies ist bei fast jeder technischen Aufgabenstellung der Fall. Deshalb wurden im Rahmen dieses Forschungsvorhabens verschiedene Ansätze zur Berücksichtigung von Nebenbedingungen untersucht. Diese Untersuchungen basieren alle auf dem sogenannten Straffunktionsansatz. Kritisch gestaltet sich insbesondere die Implementation eines eindimensionalen beschränkten Suchverfahrens zur Ermittlung der optimalen Schrittlänge. Hier werden derzeit zwei Strategien untersucht: das Liniensuchverfahren nach Moré und Thuente [8] sowie ein Einschachtelungsansatz.

Tab. 1: Testfunktion Beale

	Boxing	Boxing	Thuente	Thuente
Startpkt	Δ in %	Sims	Δ in %	Sims
1	0,2	2793	3,6	67
2	1,2	8969	2,5	59
3	0,5	10113	1,4	59
4	0,5	9333	-	-
5	0,8	10022	-	-

Tab. 2: Testfunktion Rosenbrock

	Boxing	Boxing	Thuente	Thuente
Startpkt	Δ in %	Sims	Δ in %	Sims
1	0,2	2917	0,2	47
2	0,2	3104	0,8	81
3	0,2	3086	1,9	72
4	0,2	2948	4,4	32
5	0,2	2783	-	-

Bei den derzeit durchgeführten Evaluationen anhand von mathematischen Testproblemen

(Tabelle 1 bis 3) zeichnet sich der Trend ab, dass das Verfahren mit dem Thuente-Ansatz schneller, dafür aber ungenauere Ergebnisse liefert, während die das Einschachtelungsverfahren anwendende Strategie relativ lange für die Konvergenz benötigt, dafür aber genauere Ergebnisse liefert. Beide Ansätze weisen das Problem der Wahl des Strafparameters auf.

Tab. 3: Testfunktion Schwefel 2.46

	Boxing	Boxing	Thuente	Thuente
Startpkt	Δ in %	Sims	Δ in %	Sims
1	0,1	1717	0,1	532
2	1,0	1932	0,1	490
3	1,4	571	0,9	1093
4	0,7	808	0,3	1081
5	1,2	4180	0,1	216

3.2 Polyederstrategie

Polyederverfahren wie der Complex-Algorithmus nach Box [4] oder der Simplex-Algorithmus nach Nelder und Mead [9] zeichnen sich dadurch aus, dass sie unter Anwendung verschiedener Operationen (Spiegelung, Kontraktion, Reflektion) einen m-dimensionalen Polyeder durch den n-dimensionalen Raum in Richtung eines Optimums bewegen ($m > n$). Die im Rahmen dieses Forschungsvorhabens entwickelte Polyederstrategie stellt ein Mischverfahren dar, das Elemente des Simplex und des Complex-Algorithmus miteinander kombiniert. Es ist daher sowohl zur Optimierung unbeschränkter als auch zur Optimierung beschränkter Problemstellungen geeignet.

3.2.1 Theoretische Darstellung des Verfahrens

Neben den für Polyederverfahren üblichen Operationen Spiegelung, Reflektion und Kontraktion führt das hier vorgestellte Verfahren nach erfolgreicher Reflektion eine Überreflektion durch. Auch kann bei nicht erfolgreicher Kontraktion eine sogenannte multiple Kontraktion durchgeführt werden, bei der sich der gesamte Polyeder um den bislang besten Punkt herum zusammenzieht. Je nach Art der Optimierungsaufgabe führt das Verfahren mit multipler Kontraktion besser oder weniger gut zum gewünschten

Ergebnis. Daher hat der Anwender die Möglichkeit, dieses Feature gezielt auszuwählen. Außerdem wird der Reflexionsfaktor bei dem hier vorgestellten Verfahren automatisch an die Eigenschaften der zu bearbeitenden Optimierungsaufgabe angepaßt, bleibt während einer Optimierung jedoch konstant.

Neben diesen Charakteristika zeichnet sich das Verfahren im wesentlichen durch folgende Eigenschaften aus:

- üblicherweise arbeitet das Verfahren mit einem ausgehend vom durch den Anwender vorgegebenen Startpunkt zufällig aufgebauten Polyeder;
- kombiniertes Abbruchkriterium der beiden oben aufgeführten Verfahren;
- automatischer Teilrestart nach zu großer Anzahl von erfolglosen Kontraktionen;
- lokale oder globale Ausrichtung des Restarts, je nach Vorgabe des Anwenders.

Erste Evaluationen, basierend auf mathematischen Testfunktionen (Tabelle 4 und 5), führen zu vielversprechenden Ergebnissen, insbesondere bei den beschränkten Testfunktionen.

Bei den Benchmarkfunktionen ohne Beschränkungen ist jeweils die Anzahl der Simulationen des besten Laufs dargestellt¹, da für statistische Auswertungen nicht genügend Referenzdaten zur Verfügung standen. In Tabelle 2 sind dagegen jeweils die Anzahl der Simulationen dargestellt, die die Algorithmen zur Erzielung einer 99%igen Konvergenzsicherheit benötigen würden.

Bei den unbeschränkten Testproblemen ist das Polyederverfahren von der Güte her vergleichbar mit dem speziell für diese Aufgabenstellungen entwickelten Verfahren nach Nelder und Mead; bei den beschränkten Testbeispielen schneidet das Polyederverfahren weitaus besser ab als die Verfahren nach Box und Guin.

Tab. 4: Evaluierung Polyederverfahren (Benchmarks ohne Beschränkungen)²

	T1	T2	T3	T4	T5	T6	T7
Pol	225	165	390	360	100	30	185
Box	300	400	710	n.e.	175	n.e.	295
Sub	290	250	n.e.	370	180	35	270
Sim	210	220	890	n.e.	80	n.e.	140
Nel	200	190	770	330	75	40	160

Tab. 5: Evaluierung Polyederverfahren (Benchmarks mit Beschränkungen)

	Smith	Wolfe	Bracken
Pol	270	545	59
Box	32676	7041	448
Guin	35896	9169	466

3.2.2 Modelloptimierung magnetischer Komponenten

Das Polyederverfahren wurde aufgrund seiner zuverlässigen Konvergenzeigenschaften zur Parametrisierung eines Kompaktmodells für hartmagnetische Materialien [14] eingesetzt.

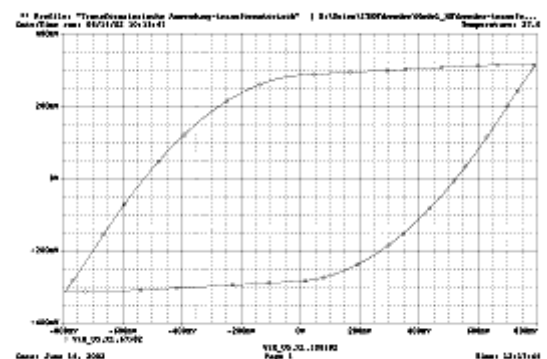


Abb. 5: minor loop

Dabei waren die Modellparameter jeweils so zu wählen, dass das Modell die zu bestimmten Materialien vorgegebenen minor loops (Kleinsignalverhalten, Abb. 5) wie auch Hysteresekurven (Großsignalverhalten, Abb. 6) nachbilden kann. Die für Klein- und Großsignalverhalten ermittelten Parametersätze führten zu sehr guten Übereinstimmungen mit den Messwerten. Die Ergebnis-

¹ n.e. bedeutet "Optimum nicht erreicht"

² Pol: Polyederverfahren; Box: Complex Algorithmus nach Box; Sub: Simplexverfahren; Sim: Simplexverfahren; Nel: Simplex nach Nelder und Mead; Guin: Complex Algorithmus nach Box

kurven sind in Abbildung 5 und Abbildung 6 dargestellt.

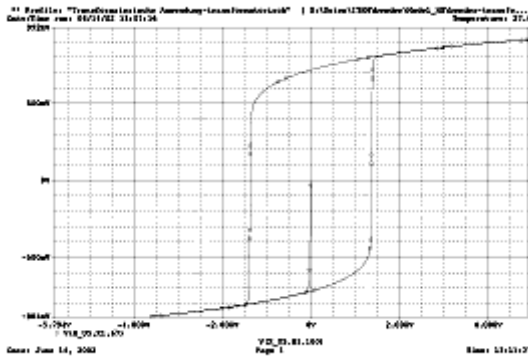


Abb. 6: Hysteresekurve

3.3 Verfahrensvergleiche mit Hilfe der MOSCITO-Schnittstelle

Im Laufe des Verbundprojektes OMID wurde MODOS in des System zur Verteilten Simulation und Optimierung, MOSCITO, eingebunden. Hierzu waren die entsprechenden Schnittstellen und Agentenbeschreibungen bereit zu stellen. Die Arbeiten wurden in enger Kooperation mit der Fraunhofer Gesellschaft Dresden durchgeführt.

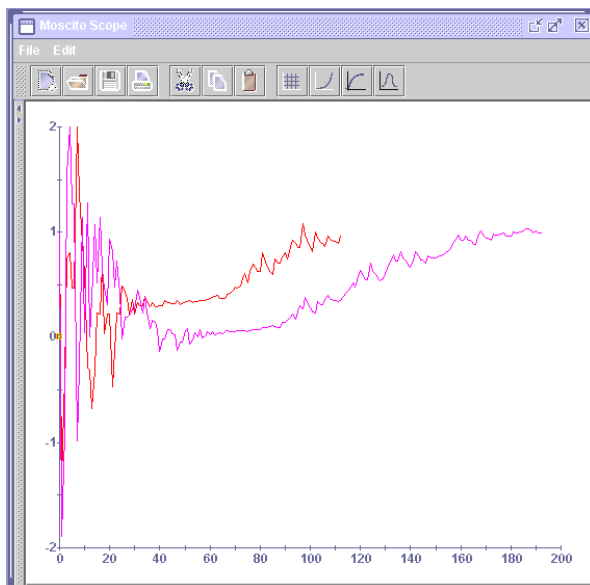


Abb. 7: MODOS-Optimierungen mit MOSCITO

In Abbildung 7 ist die Veränderung eines Designparameters während zwei verschiedener Optimierungsläufe dargestellt. Beide Läufe wurden mit dem in Abschnitt 3.2 vorgestellten Polyederverfahren durchgeführt

und unterscheiden sich lediglich in der Wahl des Abbruchkriteriums. Der erste Lauf (rot) arbeitete mit einer Fehlervorgabe von $1.0e-3$ und erzielte im Ergebnis einen Zielfunktionswert von $1.4823e-3$ ³. Der zweite Optimierungslauf (rosa) arbeitete mit einer Fehlervorgabe von $1.0e-4$ und erzielte im Ergebnis einen Zielfunktionswert von $1.8054e-5$.

An diesen beiden Ergebnissen sind die zentralen Eigenschaften des Polyederverfahrens deutlich zu erkennen:

- einstellbare Konvergenzgeschwindigkeit die mit der Genauigkeit des Ergebnisses korreliert ist;
- im hier gewählten lokalen Modus zieht sich der Polyeder relativ schnell zusammen, was insgesamt betrachtet zu einer verhältnismässig schnellen Konvergenz führt;
- durch die Überreflektionen und Neustart hat der Polyeder aber auch wieder die Möglichkeit zu expandieren (ist jeweils gegen Ende der Suche zu beobachten). Dieses Verhalten verhindert eine vorzeitige Konvergenz.

3.4 Hybride Strategien

In Kooperation mit dem Forschungszentrum Karlsruhe wurde während dieses Verbundvorhabens an der Entwicklung eines hybriden Optimierungsverfahrens gearbeitet. Grundlage dafür stellten zum einen das am Forschungszentrum entwickelte GADO-Verfahren [10] sowie die am ITEM implementierten direkten Suchverfahren nach Rosenbrock [6] und Box [4] dar. Das GADO-Verfahren verbindet die positiven Eigenschaften klassischer genetischer Algorithmen geschickt mit den Vorzügen der evolutionären Algorithmen.

3.4.1 Eingesetzte Umschaltkriterien

Den kritischsten Punkt bei der Entwicklung hybrider Strategien stellt die Formulierung erfolgreicher Strategien zur Kombination der Optimierungsalgorithmen dar. Bei den hier

³ Das tatsächliche Optimum liegt bei einem Zielfunktionswert von 0.0.

durchgeführten Arbeiten wurden folgende Kopplungsstrategien untersucht:

- Vorinitialisierte Startpopulation
10% - 100% der Individuen der Startpopulation werden voroptimiert
- Nachkommensverbesserung
lokales Suchverfahren wird zur Optimierung des besten Nachkommens eingesetzt
- Nachoptimierung
Finetuning der Evolutionsergebnisse mit lokalem Suchverfahren

Die theoretischen Hintergründe dieser Strategie werden in [11] detailliert erläutert.

3.4.2 Ergebnisse zur Strukturoptimierung der Aktorplatte

Aufgrund der Möglichkeit, Problemstellungen mit variierender Anzahl an Designparametern behandeln zu können, eignet sich das hybride Verfahren insbesondere zur Strukturoptimierung der Aktorplatte (Abbildung 7).

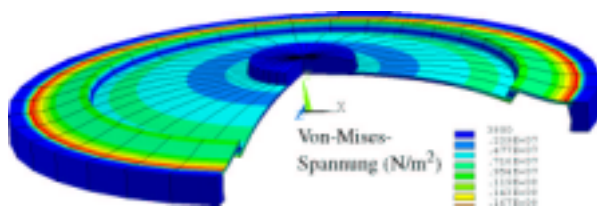


Abb. 7: Aktorplatte

Die Aktorplatte kann beispielsweise in fluidischen Systemen als Ventil eingesetzt oder zum Aufbau eines Braillearrays verwendet werden. Auf Realisierung und Funktion der Aktorplatte geht der Beitrag der Bartels mikrotechnik AG genauer ein [12]. Ziele der Optimierung dieses mikrotechnischen Elements war es, neben einer möglichst großen Auslenkung nur kleine Verspannungen in der Platte zu haben.

In Abbildung 8 sind die Ergebnisse zur Strukturoptimierung der Aktorplatte dargestellt. Mit dem manuell optimierten Entwurf wurden weitaus schlechtere Ergebnisse erzielt als mit den automatisierten Ansätzen zur Optimierung des Systems. Überraschend ist jedoch insbesondere, dass das beste Ergebnis

von der hybriden Strategie geliefert wurde. Auch liegt die Auslegung dieser Designvariante nicht unbedingt auf der Hand. Manuelle Designoptimierungen hätten wahrscheinlich kaum zu solch einem Ergebnis geführt.

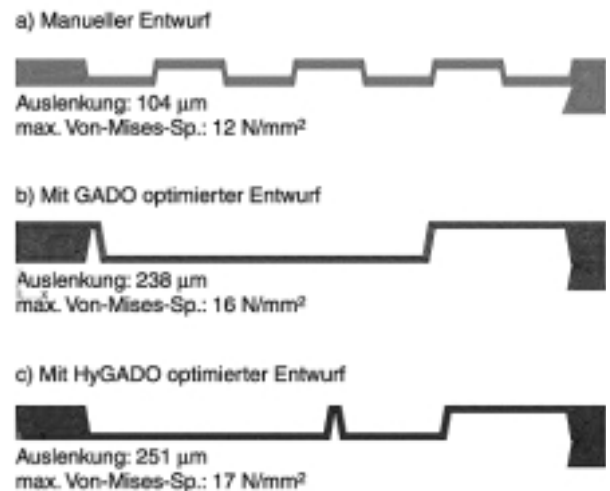


Abb. 8: Ergebnisse zur Strukturoptimierung der Aktorplatte

4 Multikriterielle Optimierung einer Mehrfachionisationskammer

Die im Verbundprojekt OMID im Rahmen einer vorwettbewerblichen Entwicklung untersuchte Mehrfachionisationskammer (Abbildung 9), kurz MFIK, der Firma VacuTec Meßtechnik GmbH, erweitert die Messwerte klassischer Ionisationskammern um eine eindimensionale Ortsauflösung. Mit der MFIK werden Materialbahnen während der Produktion bezüglich ihrer Kantenlage oder Flächenmasse überwacht.

Bei der Optimierung sind drei Kriterien zu berücksichtigen, die teils gegenläufig sind:

- der Strom in den Zellen soll absolut maximiert werden, um einen guten Rauschabstand zu erhalten
- die Steigung der Stromflanke soll maximiert werden, um eine exakte Kantendetektion zu ermöglichen
- der Durchstrahleffekt soll minimiert werden, um die Signalverfälschung zu verringern

Für die Optimierung sind der Kammerdruck, die Abstände von der Quelle zur Materialbahn und von der Materialbahn zur Kammer, der Elektrodenabstand und die Abschirmdicke variabel. Insgesamt sind also fünf Parameter für die Optimierung in den konstruktionsbedingten Grenzen frei, und es existieren drei Kenngrößen, anhand derer eine Aussage über die Optimalität getroffen werden soll.

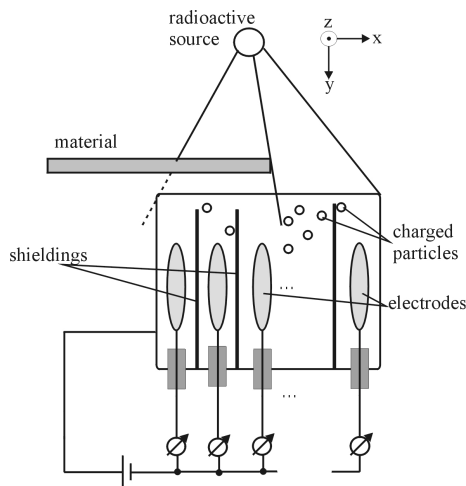


Abbildung 9: Skizze der Mehrfachionisationskammer

Ziel der Designoptimierung für die MFIK war neben der eigentlichen Systemoptimierung die Untersuchung des Einflusses der Bewertungsfunktion.

Für die MFIK wurde am ITEM ein physikalisch parametrisierbares C-Modell entwickelt, das einfach mit MODOS gekoppelt und in Verhaltenssimulatoren wie SABER eingebunden werden kann. Die diversen Zielfunktionsformulierungen wurden ebenfalls als C-Funktionen zu MODOS hinzugefügt. Untersucht wurden verschiedene, statisch gewichtete Ansätze sowie ein Ansatz zur dynamischen Wichtung der Einzelkriterien. Die Durchführung der Optimierungen erfolgte mit dem Complex-Algorithmus nach Box [4] bzw. mit dem RCT-Algorithmus [7].

Im Vorfeld der eigentlichen Designoptimierungen wurden die Maxima sowie die Minima der Einzelkriterien ermittelt. Erst dadurch werden gezielte Skalierungen und Gewichtungen der Einzelkriterien möglich. Unabhängig

von der gewählten Zielfunktionsart ergaben die statischen Gewichtungen immer eines der folgenden qualitativen Ergebnisse:

- Verbesserung hinsichtlich des Absolutwertes des Stromes und hinsichtlich der Stromflankensteigung bei gleichzeitiger Verschlechterung des Durchstrahleffektes
- Verbesserung des Durchstrahleffektes bei gleichzeitiger Verschlechterung des Absolutwertes des Stromes sowie der Stromflankensteigung

Erst die in [13] vorgeschlagene dynamische Gewichtung der Einzelkriterien führt zu einer gleichzeitigen Verbesserung aller drei Ziele: die Durchstrahlung konnte um die Hälfte reduziert werden, wobei die Stromflankensteigung in etwa verdoppelt wurde und der Absolutwert des Stromes mit dem Faktor 2,5 zunahm.

5 Ergebnisse zur modellbasierten Designoptimierung medizinischer Teststreifen

Ebenfalls im Rahmen des Verbundprojektes OMID wurde eine Optimierung des Mikrosystems „Medizinische Teststreifen“ durchgeführt. Abbildung 10 zeigt den mit Säulen strukturierten Bereich eines Mikrokanals.

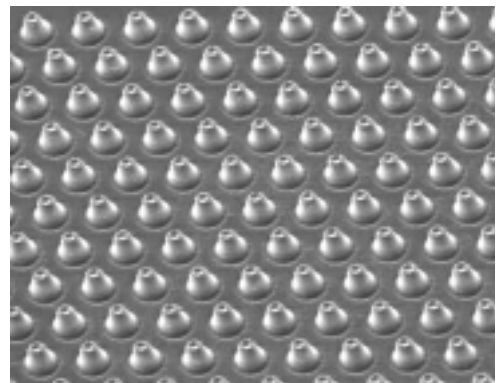


Abbildung 9: Strukturierter Mikrokanal

In Kooperation mit dem Industriepartner STEAG microParts GmbH und dem Forschungszentrum Karlsruhe wurde am ITEM ein Verhaltensmodell entwickelt, das die Grundlage der modellbasierten Optimierung bildet [15].

Ziel der Optimierung war es zunächst, einen optimalen Zusammenhang der Strukturhöhe

und des Säulenabstands zu finden. Für eine vorzugebende Befüllungszeit werden die entsprechenden Werte ermittelt, so dass der Industriepartner einen Wert vorgibt, beispielsweise nach prozesstechnischen Belangen und über den anderen die gewünschte Befüllungszeit einstellen kann. In der Anwendung entspricht diese Zeitvorgabe Ablösezeiten für Reagenzien oder chemischen Reaktionszeiten, die möglichst genau erreicht werden sollen.

Als Messtrecke dient der Kanalbereich zwischen der Position 1 mm hinter dem Kanaleinlass und 7 mm. Dieser Kanalbereich soll in einer bestimmten Zeit von der Fließfront überstrichen werden.

Es ist ein Zusammenhang zwischen Strukturhöhe, Säulenabstand, Druck und Volumenstrom gesucht, der den jeweils gültigen Arbeitspunkt bestimmt. Eine geschlossene Zielfunktion ist hierfür nicht definierbar. Die Grundlage zur Lösung dieses Problems bilden Variantensimulationen, in denen der Fluss in Abhängigkeit vom Druck einerseits, sowie von Strukturhöhe und Säulenabstand in verschiedenen Kombinationen andererseits bestimmt wird. Die Überführung der Simulationsergebnisse in eine analytische Beziehung erfolgte zunächst anhand der einzelnen Parameterkurven, die mit der gleichen Fitfunktion approximiert wurden, anschließend wurden die erhaltenen Fitparameter in Abhängigkeit der Optimierungsparameter erneut approximiert.

Für Aussagen über die Befüllungszeit der Teststrecke benötigt man eine genaue Erfassung von simulierter Zeit und erreichter Fließfrontposition. Daher wurde ein linearer Interpolationsansatz zwischen den räumlichen und zeitlichen Simulationswerten implementiert, welche die Start- und Endpositionen der Zeitmessung umgeben. Somit werden die genauen Zeit- und Positionsmarken für die Teststrecke ausreichend genau approximiert.

5.1 Optimierungsergebnis

Je nach Vorgabe des Optimierungszieles gibt es bei zwei gleichgerichtet wirkenden Parametern einen mehr oder weniger großen Lösungsraum an Arbeitspunkten, in dem sich für jeden Wert des einen Parameters ein Wert des anderen finden lässt, um die gleiche vor-

gegebene Befüllungszeit für eine vorgegebene Teststrecke zu erreichen. Das Optimierungsergebnis ist unendlich vieldeutig, d.h. es ergibt sich kein optimaler Arbeitspunkt, sondern ein als optimal ermittelter Zusammenhang zwischen den beiden Parametern, wie in Abbildung 10 dargestellt.

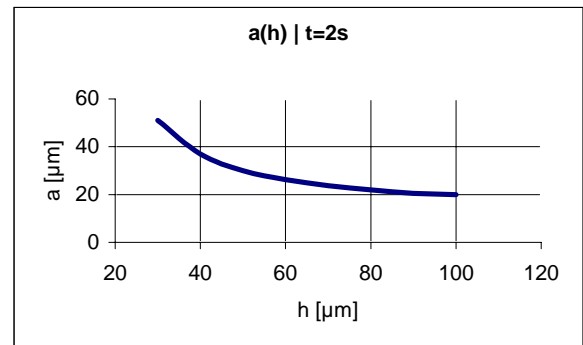


Abbildung 10: Parameteroptimierung für Abstand und Höhe der Säulen

Für eine frei gewählte Zeit von 2 Sekunden für die Teststrecke 1 mm bis 7 mm ergibt sich die obige Zuordnung von Wertepaaren. Qualitativ ist nachvollziehbar, dass sich bei einer großen Strukturhöhe der Kanalwiderstand verringert, was durch enger stehende Säulen, also einem kleineren Säulenabstand ausgeglichen wird, um die gleiche Befüllungszeit, d.h. den gleichen Gesamtfluss zu erhalten und umgekehrt.

Da es für eine zu formulierende Zielfunktion kein eindeutiges Minimum gibt, werden zunächst eindimensionale Optimierungsansätze verfolgt, bei denen für feste Werte der Strukturhöhe der Säulenabstand dem Zeitkriterium entsprechend eingestellt wird. Für die Aufnahme einer Kurve wie in Abbildung 10 wird somit je ein Optimierungslauf pro Stützstelle benötigt.

6 Zusammenfassung

In diesem Beitrag wurde der Hintergrund zur Entwicklung des modellbasierten Designoptimierungssystems MODOS kurz erläutert, seine Anwendung im Bereich der Mikrosystemtechnik an verschiedenen Anwendungen demonstriert sowie die Entwicklung verschiedener neuer mathematischer Optimierungsalgorithmen erläutert.

Die im Projekt betrachteten Anwendungen verdeutlichen verschiedene Aspekte, die bei

der automatisierten Auslegung von Mikrosystemen zu berücksichtigen sind. So sind in der Regel multikriterielle Optimierungsaufgaben zu bearbeiten, wobei die einzelnen Optimierungsziele auch gegenläufig sein können, wie dies bei der MFIK der Fall war. Die Strukturoptimierung der Aktorplatte hat auf beeindruckende Weise verdeutlicht, welchen Vorteil die Entwickler aus der Optimierung ziehen können.

Literatur

- [1] Schneider, A.; Schneider, P.; Gramatová, E.; Ivask, E.: Internetbasierter Systementwurf mit MOSCITO. 10. E.I.S. Workshop, Dresden, 3.-5. April 2001
- [2] Gibron, M.: Universelle Schnittstelle zur Werkzeugintegration in SABER. 2. Statusseminar zum Verbundprojekt Modellbildung für die Mikrosystemtechnik, Paderborn, 1:75-82, Dezember 1997
- [3] Peters, D.; Laur, R.: Microsystem Model Evaluation Via Optimization Strategies. Proceedings of the 4th International Conference on Modeling and Simulation of Microsystems, South Carolina, März 2001
- [4] Box, M. J.: A new method of constrained optimization and a comparison with other methods. *Computer Journal*, 8:42-52, 1965
- [5] Peters, D.; Bolte, H.; Nüssen, O.; Marschner, C.; Laur, R.: Enhanced optimization algorithms for the design of microsystems. DTIP, April 2001
- [6] Rosenbrock, H. H.: An Automatic Method for finding the Greatest or Least Value of a Function. *Computer Journal*, 3:175-184, 1960
- [7] Peters, D.; Laur, R.: An Improvement of Rosenbrock's Algorithm for modelbased Design Optimization of Microsystems. Sixth IEEE International Conference on Electronics, Circuits and Systems, Zypern, 1999
- [8] Moré, J.J.; Thuente, D.J.: Line Search Algorithms with Guaranteed Sufficient Decrease. *ACM Transactions on Mathematical Software*, Vol. 20, 3:286-307, September 1994
- [9] Nelder, J.A.; Mead, R.A.: A simplex method for function minimization. *Computer Journal*, 7:308-313, 1965
- [10] Jakob, W.: HyGLEAM - An Approach to Generally Applicable Hybridization of Evolutionary Algorithms. in: J.J.Merelo, A. Panagiotis, H.-G. Beyer (Hrsg.): *Conf. Proc. PPSN VII, LNCS 2439*, Springer-Verlag, Berlin. 2002.
- [11] Jakob, W.; Peters, D.: Internetbasierte Designoptimierung mit HyGLEAM. Dieser Tagungsband.
- [12] Tahhan, I.: Die Aktorplatte – ein universeller fluidischer Mikroaktor. Dieser Tagungsband.
- [13] Bolte, H. et al.: Verhaltensmodelle und multikriterielle Optimierung im Entwurfsprozess von komplexen Systemen am Beispiel einer Mehrfachionisationskammer. ASIM 2001
- [14] Brachtendorf, H.G.; Laur, R.: A Hysteresis Model for Hard Magnetic Core Materials. *IEEE Transactions on Magnetics*, vol. 33, (1)723-727, 1997
- [15] Nüssen, O. et al.: Modellbasierte Optimierung Medizinischer Teststreifen aus Kunststoff, Beitrag im Tagungsband des OMID-Abschluss-Seminars, November 2002