

# Formale Eigenschaften des SILKE-Ansatzes zur Kontrolle selbstoptimierender Systeme

Nils Rosemann

Bastian Neumann

Werner Brockmann

**Abstract:** Im Rahmen des Organic Computing werden auch eingebettete Echtzeitsysteme immer öfter mit selbstoptimierenden Eigenschaften ausgestattet. Die selbsttätige Adaption des Systemverhaltens kann aber potentiell zu kritischen Systemzuständen führen. Für spezielle lernfähige Fuzzy-Systeme wurde der sogenannte SILKE-Ansatz entwickelt, um kontrollierend auf den Selbstoptimierungsprozess einzuwirken und ihn zu gewünschten (Meta-)Eigenschaften hinzuleiten. In dieser Arbeit wird als Erweiterung zu bisherigen empirischen Untersuchungen diese Fähigkeit exemplarisch für die ausgleichende Wirkung des SILKE-Ansatzes formal nachgewiesen.

## 1 Einleitung

Die Randbedingungen für kommende Generationen von technischen Geräten verändern sich aufgrund ihrer immer höheren Verbreitung und Integration in die menschliche Umgebung. So müssen sich diese Systeme intelligent an die aktuelle Umgebung, die aktuelle Problemstellung und den Benutzer adaptieren können, um neue Einsatzgebiete und Anwendungen zu erschließen. Die Anforderungen und die Komplexität der Geräte steigen dadurch. Gleichzeitig sollte der Entwurfsaufwand verringert werden. Diesen sich wandelnden Bedingungen begegnet man im *Organic Computing* durch die Einführung von Selbst-X-Eigenschaften wie Selbstorganisation und Selbstoptimierung, um ein emergentes Verhalten zu erreichen [MSvdMW04].

Die meisten dieser technischen Geräte sind eingebettete Systeme, vielfach mit harten Echtzeitanforderungen. Bei ihnen stellt sich vor allem die Aufgabe, eine schnelle und sichere Selbstoptimierung zu erreichen. Solche Systeme müssen also im laufenden Betrieb den Erfolg ihrer Aktivität beurteilen und daraus Verbesserungsmöglichkeiten ermitteln können (Online-Lernen). Dieser Lernvorgang zeichnet sich dadurch aus, dass Lernvorgaben nur für die aktuelle Betriebssituation möglich sind. Das System muss somit aus zeitlich gestaffelten, lokalen Lernvorgaben ein global optimales Verhalten ausprägen können. Zudem müssen die Lernvorgaben inkrementell verarbeitet werden.

Auf diese Weise ergibt sich neben der eigentlichen Wirkungsschleife zwischen eingebettetem System und der Umgebung, in die es eingebettet ist, eine weitere, indirekte Wirkungsschleife. Denn das, was das System in einer Situation lernt, bestimmt das zukünftige Verhalten des Systems, und damit aber auch den zukünftigen Lernvorgang. Durch eine solche Rückkopplung können kleine Veränderungen und die Selbstoptimierung als solche beliebig große Auswirkungen haben und potentiell zu einem chaotischen Systemverhalten führen.

Veränderungen dieser Art lassen sich aber nicht verhindern. Eine Anpassung an die aktuelle Umgebung durch Selbstoptimierung kann nur anhand von Sensorwerten geschehen, und die sind mit Rauschen und Ausreißern behaftet. Darüber hinaus gibt es natürlich auch Veränderungen durch Alterung und Störungen. Dies führt dazu, dass ein selbstoptimierendes System sowohl stabil sein muss, um nicht durch transiente Störungen und Rauschen gelerntes Wissen zu zerstören, als auch plastisch, um sich flexibel auf neue Gegebenheiten einzustellen (*stability-plasticity-dilemma*). Außerdem muss sichergestellt werden, dass durch die Selbstoptimierung das System niemals in kritische Zustände gerät, der Selbstoptimierungsvorgang muss also kontrolliert werden.

Lernalgorithmen für selbstoptimierende, eingebettete Echtzeitsysteme müssen somit den folgenden Anforderungen gerecht werden: Inkrementelle lokale Lernvorgaben müssen ausreichen, um schnell und stabil ein globales Systemverhalten zu lernen (*Konvergenzgeschwindigkeit und -güte*). Unsichere Sensorwerte dürfen weder die Sicherheit des Gesamtsystems noch den Erfolg der Selbstoptimierung gefährden (*Robustheit, Vertrauenswürdigkeit*). Und schließlich müssen solche Lernalgorithmen den Entwurfsprozess unterstützen, indem sie transparent arbeiten und interpretierbar sind (*Beherrschbarkeit*).

Formal handelt es sich beim Online-Lernen mit verrauschten, inkrementellen und lokalen Lernvorgaben um ein *schlecht gestelltes* inverses Problem, da die Lösung nicht eindeutig ist. Zur Behandlung solcher Probleme müssen zusätzliche Informationen über die Lösung genutzt werden. Dies entspricht einer Umformulierung des Problems und nennt sich *Regularisierung* [Tik43, EHN96]. Für überwachtetes Lernen gibt es dafür verschiedene Ansätze [BPR07]. Dabei werden jedoch stets globale Zusatzinformationen genutzt. Aufgrund obiger Anforderungen muss die Kontrolle der Selbstoptimierung jedoch inkrementell und lokal erfolgen. An dieser Stelle setzt der sog. SILKE-Ansatz an (*System to Immunize Learning Knowledge-based Elementes*) [BH07]. Er erweitert eine Klasse lernfähiger Fuzzy-Systeme um Methoden zur Kontrolle des Selbstoptimierungsvorgangs. Dabei wird der Lernvorgang ähnlich wie bei klassischer Regularisierung, allerdings lokal und inkrementell, in Richtung bestimmter anwendungsspezifischer Meta-Eigenschaften gelenkt. Diese steuernde Wirkung wurde bisher empirisch untersucht [RB07]. Das Ziel der vorliegenden Arbeit ist es nun, den SILKE-Ansatz und seine Wirkung zu formalisieren und darauf aufbauend auch formal für ein typisches Template nachzuweisen, dass der Selbstoptimierungsvorgang in Richtung der gewünschten Meta-Eigenschaft gelenkt wird, in diesem Fall zu einer stabileren Konvergenz.

## 2 Kontrolle selbstoptimierender Systeme mit dem SILKE-Ansatz

### 2.1 Ansatz

Grundlage für den SILKE-Ansatz sind spezielle Takagi-Sugeno (sTS) Fuzzy-Systeme nullter Ordnung. Diese verwenden lineare B-Spline Zugehörigkeitsfunktionen, so dass jede Eingangsdimension durch eine Teilung der Eins zerlegt ist. Insgesamt eignen sie sich ideal für den Einsatz als Online-Lernverfahren in eingebetteten Systemen [FP06]. Wesent-

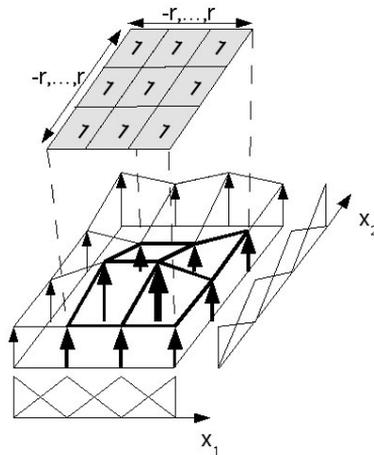


Abbildung 1: Schematische Darstellung der Wirkung eines SILKE-Templates auf die Konklusionen eines Ausschnitts einer sTS-Regelbasis. Der fett gedruckte Pfeil repräsentiert die aktive Regel. Die Zahlen sind die Koeffizienten des Templates. Die Zugehörigkeitsfunktionen bestimmen die Gitterstruktur.

liches Anwendungsgebiet ist die adaptive Regelung. Dabei übernehmen die sTS-Systeme die Rolle eines lernenden Funktionsapproximators, der hier als Regler eingesetzt wird.

Diese sTS-Fuzzy-Systeme lassen sich nun entsprechend der *Observer-Controller*-Architektur [RMB<sup>+</sup>06] um eine weitere Kontrollschleife erweitern. Dazu ermittelt ein sogenanntes *Template* analog zu einem *Observer*, ob die Regelbasis des sTS-Systems in der Nähe des aktuellen Arbeitspunktes vordefinierten Meta-Eigenschaften genügt. Dies kann zur reinen Selbstüberwachung eingesetzt werden [BR08]. Falls dabei zu große Abweichungen festgestellt werden, sollte die Regelbasis des sTS-Systems korrigiert werden, wodurch sich der Wirkungskreis auf höherer Ebene schließt.

Eine wichtige Meta-Eigenschaft ist *Glattheit* der approximierten Funktion bzw. des gelernten Wissens. Denn Glattheit bedeutet, dass kleine Änderungen am Eingang keine krassen Änderungen am Ausgang hervorrufen, dass also der Lern- und Kontrollvorgang mit der Umgebung harmoniert. Im Folgenden wird formal nachgewiesen, dass sich im SILKE-Ansatz diese Glattheitseigenschaft durch ein *mittelndes Template* herstellen lässt.

## 2.2 Formalisierung

Sei  $n$  die Anzahl der Eingänge eines sTS-Fuzzy-Systems. Durch Kompilation einer sTS-Regelbasis wird diese normalisiert, dadurch kann man sie sich als Gitter vorstellen [BR08]. Auf den Gitterpunkten sind die Konklusionen der Regeln eingetragen. Also kann man die Regelbasis als Abbildung  $o : \mathbb{N}^n \rightarrow \mathbb{R}$  betrachten. Ein Gitterpunkt  $p \in \mathbb{N}^n$  entspricht dann

einer *Regel*, und der Wert  $o(p)$  der Regelkonklusion (*Fuzzy-Singleton*). Ein Template  $T$  ist nun eine Funktion

$$T : \hat{N} \rightarrow \mathbb{R}^+ \quad \text{mit} \quad h = T(p) \quad (1)$$

mit  $\hat{N} \subset \mathbb{N}^n$  als Indexmenge der Regeln. Abbildung 1 stellt die Zusammenhänge schematisch dar. Zu sehen ist ein Ausschnitt der Regelbasis in Form eines Gitters, das von den Zugehörigkeitsfunktionen definiert wird. Auf die Nachbarschaft der aktiven Regel wird dann schematisch ein Template angewendet.

Der Wert  $|h|$  heißt *Abweichungsgrad* der Regel  $p$  bzgl.  $T$ . Er gibt an, wie stark die aktuellen lokalen Nachbarregeln von  $p$  die durch  $T$  repräsentierte Meta-Eigenschaft verletzen.  $|h| = 0$  bedeutet, dass die Meta-Eigenschaft vollständig erfüllt ist. Je größer  $|h|$  wird, um so weiter ist die Regelbasis von der vorgegebenen Eigenschaft entfernt.

Die betrachtete Regel kann dann anhand von

$$o_{t+1}(p) = (1 - \alpha) \cdot o_t(p) + \alpha \cdot (o_t(p) - T(p)) \quad (2)$$

korrigiert werden. Die *Anpassungsrate*  $\alpha \in ]0; 1]$  bestimmt dabei, wie stark die Meta-Eigenschaft erzwungen wird.

Im Folgenden betrachten wir Templates, die sich als Faltung der Regelbasis beschreiben lassen:

$$T : \hat{N} \rightarrow \mathbb{R} \quad \text{mit} \quad h = T(x), \quad (3)$$

wobei

$$T(x) = o(x) - \frac{1}{(2r + 1)^n} \sum_{u \in U} o(x - u)m(u). \quad (4)$$

Dabei bezeichnet  $r \in \mathbb{N}^*$  einen Radius. Alle Nachbarregeln, deren  $L_1$ -Abstand zur betrachteten Regel  $x$  kleiner gleich  $r$  ist, werden von dem Template berücksichtigt. Die Summe läuft daher über  $U = \{(u_1, \dots, u_i, \dots, u_n) \mid u_i \in [-r, \dots, 0, \dots, r]\}$ . Die Abbildung  $m(u)$  heißt *Maske* und bestimmt die Funktionsweise des Templates.

### 2.3 Konvergenz des mittelnden Templates

In diesem Abschnitt wird der folgende Satz bewiesen: Es sei gegeben eine Regelbasis  $o_t$  zum Zeitpunkt  $t$ . Verwendet wird das mittelnde Template  $T$ , welches definiert ist durch die Maske

$$m(u) = 1 \quad \forall u \in U. \quad (5)$$

Angenommen, das Template würde wiederholt und ohne Lernvorgabe auf einen der inneren Punkte (also nicht auf die Randpunkte) der Regelbasis angewendet, so verkleinert sich der Abstand der Regelbasis zu einer  $n$ -dimensionalen Ebene immer weiter.

Beweis: Die Beweisidee liegt darin zu zeigen, dass jede Anwendung des Templates die Gesamtabweichung der Regelbasis von einer  $n$ -dimensionalen Ebene echt verkleinert. Für

den Beweis wird eine Eigenschaft von solchen Ebenen benötigt, nämlich die Tatsache, dass für jedes  $u \in U$  gilt

$$2o(x) = o(x + u) + o(x - u). \quad (6)$$

Diese Eigenschaft lässt sich aus der Ebenengleichung herleiten:

$$o(x) = o(0) + (x_1)\delta_1 + \dots + (x_n)\delta_n \quad (7)$$

mit den Richtungsvektoren  $\delta_i$ . Damit folgt:

$$o(x + u) = o(0) + (x_1 + u_1)\delta_1 + \dots + (x_n + u_n)\delta_n$$

und

$$o(x - u) = o(0) + (x_1 - u_1)\delta_1 + \dots + (x_n - u_n)\delta_n,$$

für alle  $u$  und somit

$$\begin{aligned} & o(x + u) + o(x - u) \\ = & o(0) + (x_1 + u_1)\delta_1 + \dots + (x_n + u_n)\delta_n \\ & + o(0) + (x_1 - u_1)\delta_1 + \dots + (x_n - u_n)\delta_n \\ = & 2o(0) + (x_1 + u_1)\delta_1 + (x_1 - u_1)\delta_1 + \dots + (x_n + u_n)\delta_n + (x_n - u_n)\delta_n \\ = & 2o(0) + 2x_1\delta_1 + \dots + 2x_n\delta_n \\ = & 2o(0) + 2(x_1\delta_1 + \dots + x_n\delta_n) \\ = & 2o(x). \end{aligned}$$

Weiterhin wird die Notation vom sogenannten *Maximalindex*  $l$  benötigt. Der Maximalindex ist der Indexvektor  $l$  von der Regelkonklusion, die von der  $n$ -dimensionalen Ebene den größten Abstand hat. (Wenn es mehrere gibt, so nimmt man den mit dem größten Abweichungsgrad  $|h|$ .)

Sei nun im Folgenden  $o_x = o_t(x)$ ,  $\bar{o}_x = o_{t+1}(x)$ ,  $l = l(t)$  und  $a = 2r + 1$ . Man betrachte nun den Abstand des Nachfolgezustands  $\bar{o}_l$  zu der Ebene  $f$ :

$$\begin{aligned} |\bar{o}_l - f_l| &= \left| \frac{(a^n - (a^n - 1)\alpha)o_l + \alpha \sum_{u \in U^*} o_{l-u}}{a^n} - f_l \right| \\ &= \frac{1}{a^n} |(a^n - (a^n - 1)\alpha)o_l - a^n f_l + (a^n - 1)f_l + \alpha \sum_{u \in U^*} o_{l-u} - (a^n - 1)f_l| \end{aligned}$$

mit  $U^*$  als  $U$  ohne den Nullvektor. Da die Summe genau  $a^n - 1$  Summanden enthält, folgt:

$$= \frac{1}{a^n} |(a^n - (a^n - 1)\alpha)o_l - (a^n - (a^n - 1)\alpha)f_l + \alpha \sum_{u \in U^*} (o_{l-u} - f_l)|$$

Wegen Gleichung 6 gilt mit anschließender Anwendung der Dreiecksungleichung:

$$\begin{aligned} &= \frac{1}{a^n} |(a^n - (a^n - 1)\alpha)o_l - (a^n - (a^n - 1)\alpha)f_l + \alpha \sum_{u \in U^*} (o_{l-u} - f_{l-u})| \\ &\leq \frac{1}{a^n} (|(a^n - (a^n - 1)\alpha)o_l - (a^n - (a^n - 1)\alpha)f_l| + \alpha \sum_{u \in U^*} |(o_{l-u} - f_{l-u})|) \end{aligned}$$

Weil  $l$  Maximalindex ist, existiert mindestens ein  $i \in U$ , so dass  $|o_{l-i} - f_{l-i}|$  echt kleiner ist als  $|o_l - f_l|$ . Daher gilt:

$$\begin{aligned} &< \frac{1}{a^n} (|(a^n - (a^n - 1)\alpha)o_l - (a^n - (a^n - 1)\alpha)f_l| + \alpha \sum_{u \in U^*} |(o_l - f_l)|) \\ &= |o_l - f_l|, \end{aligned}$$

also insgesamt

$$|\bar{o}_l - f_l| < |o_l - f_l|. \quad (8)$$

Damit wird also die Abweichung der Regelbasis von der Ebene lokal an der Stelle  $l$  kleiner, und somit auch global. Für den Gesamtfehler ergibt sich entsprechend eine beschränkte, monoton fallende Folge. Und eine solche Folge muss konvergieren.

Die hier beschriebenen Schritte lassen sich graphisch veranschaulichen. Dazu zeigt Abbildung 2 eine eindimensionale Beispielfunktion. Die Stützstellen liegen an den auf der horizontalen Achse durchnummerierten Stellen. Die Abbildung zeigt dann die Wirkung des mittelnden Templates auf die Stützstelle, die jeweils den Maximalindex hat. Man kann deutlich erkennen, dass die Funktion immer flacher wird und sich einer gedachten Geraden durch die Randstützstellen annähert.

### 3 Diskussion

Die wiederholte lokale Anwendung des SILKE-Ansatzes mit einem mittelnden Template leitet den Lernvorgang also an, nicht jeder Lernvorgabe beliebig zu folgen, sondern dabei eine möglichst hohe Glattheit und Linearität einzuhalten. Der Grenzwert dieses Anleitungsvorgangs ist dabei zu jedem Zeitpunkt eine  $n$ -dimensionale Ebene. Dieser Konvergenzvorgang wird durch inkrementelle Lernvorgaben beeinflusst, so dass er verzögert oder auch beschleunigt werden kann. Die dynamische Wechselwirkung zwischen der Reaktion der Umwelt und den Lernvorgaben bestimmt abhängig von der Steuerung über die Anpassungsrate  $\alpha$  insgesamt also die sich ausprägende Regelbasis.

Nachdem sich der SILKE-Ansatz mit einem mittelnden Template bereits empirisch in Simulationen und realen Anwendungen bewiesen hat, konnte in dieser Arbeit mit formalen Methoden gezeigt werden, dass er unabhängig vom unterliegenden System den Lernvorgang steuern kann. Die Abhängigkeit des Grenzwertes von der Wechselwirkung mit dem Lernvorgang erhält dem Systemdesigner dadurch die Möglichkeit, mittels der Anpassungsrate das Verhältnis zwischen Plastizität und Stabilität einzustellen. Das Verhältnis

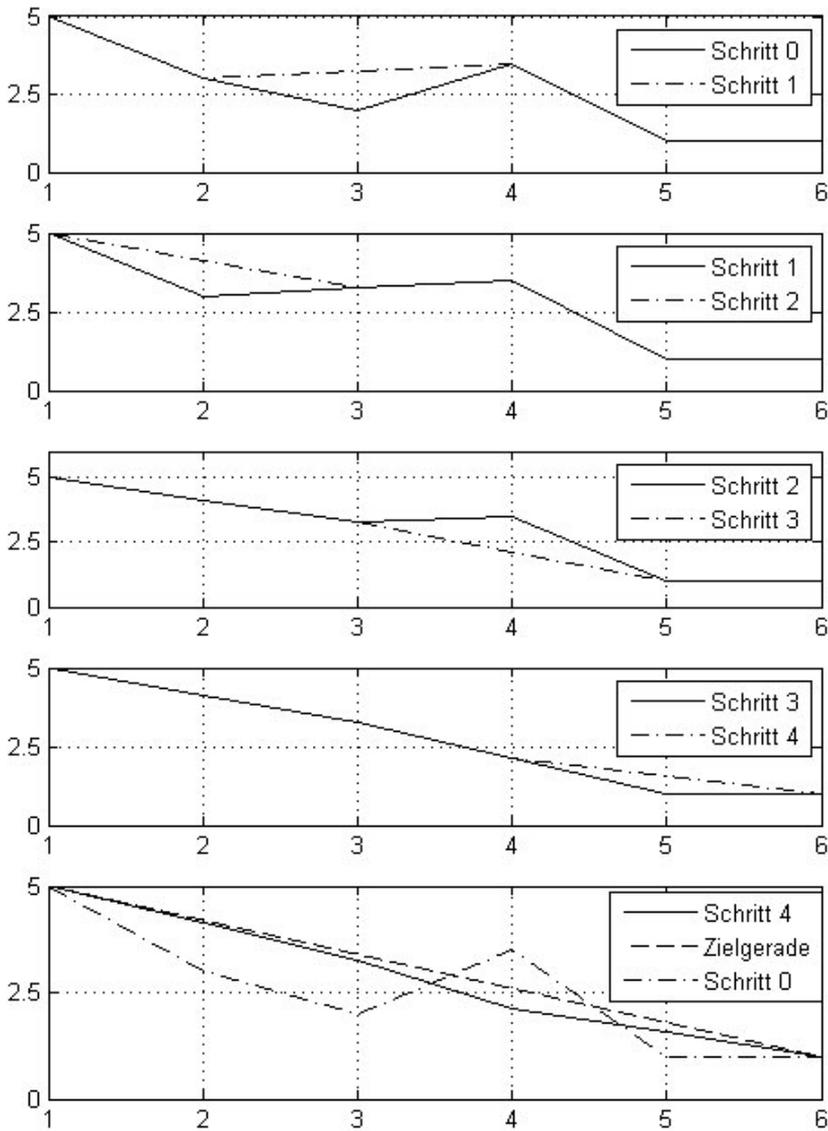


Abbildung 2: Anwendung des mittelnden Templates auf eine Beispielfunktion. Man erkennt, wie sich der Abstand der Funktion von einer gedachten Geraden schrittweise verringert.

kann sogar online und in Abhängigkeit von äußeren Bedingungen dynamisch verändert werden. Dies ermöglicht ein Systemdesign, welches das Stabilitäts-Plastizitäts-Dilemma adaptiver Systeme für viele Situationen beherrschbar macht.

Die strenge Formalisierung des SILKE-Ansatzes erlaubt es in Zukunft, die Grenzwert-Eigenschaften anderer Templates zu untersuchen (z. B. für Monotonie und Konvexität) und systematisch neue Templates zu suchen.

## Literatur

- [BH07] W. Brockmann und A. Horst. Stabilizing the Convergence of Online-Learning in Neuro-Fuzzy Systems by an Immune System-inspired Approach. *IEEE Int. Conf. Fuzzy Systems*, Seiten 1–6, 2007.
- [BPR07] F. Bauer, S. Pereverzev und L. Rosasco. On regularization algorithms in learning theory. *J. Complex.*, 23(1):52–72, 2007.
- [BR08] W. Brockmann und N. Rosemann. Instantaneous Anomaly Detection in Online Learning Fuzzy-Systems. *3rd Int. Workshop on Genetic and Evolving Fuzzy Systems, Bommerholz, Germany*, Seiten 23–28, 2008.
- [EHN96] H.W. Engl, M. Hanke und A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers, 1996.
- [FP06] J. Farrell und M. Polycarpou. *Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*. Wiley, 2006.
- [MSvdMW04] C. Müller-Schloer, C. von der Malsburg und R.P. Würt. Organic Computing. *Informatik-Spektrum*, 27(4):332–336, 2004.
- [RB07] N. Rosemann und W. Brockmann. Concept for Controlled Self-optimization in Online Learning Neuro-fuzzy Systems. *30th Ger. Conf. AI, KI 2007*, Springer LNCS 4667:498–501, 2007.
- [RMB<sup>+</sup>06] U. Richter, M. Mnif, J. Branke, C. Müller-Schloer und H. Schmeck. Towards a generic observer/controller architecture for Organic Computing. *Informatik*, Seiten 112–119, 2006.
- [Tik43] A. Tikhonov. On the stability of inverse problems. *CR (Dokl.) Acad. Sci. URSS, n. Ser.*, 39:176–179, 1943.