

Datengetriebene Systemidentifikation eines Innovation Community Modells

Ertug Olcay

Lehrstuhl für Regelungstechnik, Technische Universität München

Boltzmannstr. 15, 85748 Garching

E-Mail: ertug.olcay@tum.de

Mit dem steigenden globalen Wettbewerb werden Innovationen immer signifikanter für den Markterfolg vieler Unternehmen. Ein wichtiger Begriff dabei ist die Innovation Community, die Innovationsideen entwickeln und realisieren. Die Modellierung eines solchen nicht-physikalischen Systems ist nicht einfach und kann zum Beispiel durch agentenbasierte Modellierung erfolgen. Allerdings sind Agentenbasierte Modelle für systemtheoretische Betrachtungen nicht gut geeignet. Mit Hilfe von Ein-/Ausgangsdaten ist es möglich, ein agentenbasiertes Modell als TS-Fuzzy Modell darzustellen. Hierbei werden komplexe nichtlineare Funktionen durch Trainieren neuronaler Netze durch einfachere lokale lineare Modelle approximiert. In dieser Arbeit wird die Anwendung dieser Methode auf einem Innovation Community Modell aufgezeigt und diskutiert.

1 Einleitung

Innovationen sind entscheidend für den Markterfolg vieler Unternehmen. Ein Innovationsprozess ist ein komplexes dynamisches System, das zeitliche Entwicklungen bestimmter Kenngrößen beinhaltet [1]. Dabei sind vernünftige Gestaltung und Modellierung der Dynamik wichtig, um die Komplexität zu beherrschen und systemtheoretische Analysen durchführen zu können.

Aufgrund des globalen Wettbewerbs gewinnen Innovationsnetzwerke zunehmend an Bedeutung in Innovationsprozessen. In diesem Beitrag wird auf der Grundlage vom Innovation Community Modell [2] eine Systemidentifikation durchgeführt. Dieses Modell weist ein heterogenes Verhalten durch menschliche Akteure auf. Darüber hinaus ist es durch die agentenbasierte Modellierungsweise beschrieben. Die vorliegende Arbeit beschäftigt sich mit der Übertragung solcher komplexen dynamischen Modelle in die TS-Fuzzy Form, weil die TS-Fuzzy Darstellung eine einfachere Stabilitätsanalyse und Reglerauslegung ermöglicht. Dabei kommt die datengetriebene Identifikation von TS-Fuzzy-Modellen in Frage, mit der eine nichtlineare Funktion durch mehrere lokal affine Modelle approximiert wird [3].

Im Folgendem wird zunächst die Grundidee der agentenbasierten Modellierung vorgestellt. Als nächstes werden relevante Begriffe zu einem Innovationsprozess definiert. Im Fokus des vierten Abschnitts steht das Innovation Community Modell. An dieser Stelle wird die Theorie und Funktionsweise des angewandten Identifikationsverfahrens vorgestellt. Danach wird anhand von einem einfachen Beispiel die Modellidentifikation demonstriert. Des Weiteren wird das agentenbasierte Simulationsmodell zu Demonstrationszwecken beliebig um Stelleingriffe erweitert. Abschließend wird das agentenbasierte Innovation Community Modell datengetrieben in ein TS Fuzzy Modell umgewandelt und Simulationsergebnisse präsentiert. Die Analyse beschränkt sich dabei auf deterministische Systeme.

2 Agentenbasierte Modellierung

In den Sozial- und Naturwissenschaften gibt es komplexe Systeme, die aus mehreren miteinander und mit ihrer Umwelt interagierenden Entitäten bestehen. Solche Systeme können mit Hilfe der agentenbasierten Modellierung abgebildet werden. Im Vergleich zu traditionellen Ansätzen wie eine Beschreibung über Differentialgleichungen ermöglicht die agentenbasierte Modellierung, die Heterogenität eines Systems abzubilden und die Verhaltensregeln von Agenten explizit darzustellen [4].

Agenten sind entweder separate Computerprogramme (Prozessor) oder Systemteile, die soziale Akteure darstellt, wie Einzelpersonen, Organisationen oder Unternehmen. Sie können sich gegenseitig Informationen übermitteln. Die agentenbasierte Modellierung ist eine Modellierungsart, mit der die Interaktionen zwischen den Agenten und mit ihrer Umgebung über die Zeit abgebildet werden.

In einem Experiment wird ein Ergebnis einer Behandlung auf ein isoliertes System beobachtet. Allerdings ist die Isolierung bei sozialen Systemen in der Regel unmöglich. In den meisten Sozialwissenschaften ist die Durchführung von Experimenten schwierig. Ein wesentlicher Vorteil der agentenbasierten Modellierung ist, dass sie eine Isolation des menschlichen Systems an virtuellen Modellen ermöglicht. So wird eine transparente Betrachtung von sowohl Mikroebenen (Agentenebene) als auch Makroebenen (z.B. gesellschaftliche Veränderungen) möglich. Solche Modelle nutzt man, um die soziale Welt zu verstehen. Agentenbasierte Modellierung findet in Biologie, Volkswirtschaft, Soziologie und Businessbereich (Produktions- und Fertigungsplanung) Anwendung.

3 Definition des Konzepts Innovation Communities

Heutzutage sind Innovationen die Erfolgsgrundlage vieler Unternehmen. Allerdings bringt der Innovationsprozess eine gewisse Komplexität mit sich. Der Innovationsprozess ist ein komplexes dynamisches System und setzt sich aus interagierenden Teilsystemen zusammen [1]. Außerdem existieren dabei Wechselwirkungen mit anderen Prozessen und Systemen. Die erfolgreiche Beherrschung von Innovationsprozessen spielt für Unternehmen immer eine wichtigere Rolle vor allem im Hinblick auf den Markterfolg und Durchsetzung gegen die Konkurrenz.

Der globale Wettbewerb zwingt Unternehmen bei Forschung und Entwicklung, aber auch im Bereich der Fertigung und des Vertriebs zur Arbeitsteilung und Kooperation. Durch Kooperationen innerhalb von Innovationsnetzwerken werden fehlende Ressourcen und Kompetenzen ergänzt und die Entwicklungsprozesse beschleunigt. Ein wichtiger Begriff in einem Innovationsnetzwerk und im Bereich Innovation Management ist das Konzept der Innovation Communities.

Der Begriff „Innovation Community“ wird unterschiedlich definiert [5]. Unter anderem versteht man darunter ein Kontaktnetzwerk. Dabei handelt es sich um Kontaktplattformen und Netzwerke von Personen, die sich für ein bestimmtes Innovationsfeld interessieren und die Absicht haben, Informationen auszutauschen und geeignete Innovationspartner zu finden.

Darüber hinaus beschreibt der Begriff eine virtuelle Gemeinschaft zur Ideengenerierung. Darunter versteht man virtuelle, durch elektronische Medien gestützte Gemeinschaften zur Generierung und Bewertung von Innovationsideen und Innovationskonzepten. Außerdem sind Innovation Communities als Promotorennetzwerke zur Unterstützung konkreter Innovationsprojekte zu verstehen. Promotor bedeutet in der Ökonomie Förderer eines Innovationsprozesses (z.B. Machtpromotor, Fachpromotor etc.). Bei dieser Definition stehen die Beziehungen einer Gruppe von Innovationspromotoren im Vordergrund, die gemeinsam ein konkretes Innovationsvorhaben vorantreiben [5].

4 Identifikation eines Community Modells mit Hilfe von LOLIMOT

In diesem Abschnitt wird zunächst eine datengetriebene Modellidentifikation mit Hilfe des LOLIMOT-Algorithmus (LOcal-LInear-MODEL-Tree) vorgestellt. Im Anschluss wird ein Zweitanksystem als einfaches Beispiel mit Hilfe des LOLIMOT-Algorithmus identifiziert. Darüber hinaus wird das zu identifizierte Innovation Community Modell vorgestellt und der LOLIMOT-Algorithmus auf dieses Modell angewendet.

4.1 Datengetriebene Modellidentifikation

Prinzipiell gibt es zwei unterschiedliche Wege zur Modellidentifikation. Bei der White-Box-Betrachtungsweise ist die Struktur des Systems bekannt. White-Box-Modelle weisen eine hohe Genauigkeit auf und die Modellparameter haben physikalische Bedeutung. Bei der Black-Box-Betrachtungsweise sind keine Vorkenntnisse über die Systemstruktur vorhanden und die Modellparameter haben keine physikalische Bedeutung [6]. Unter Modellidentifikation wird die Bestimmung der Struktur, der Parameter und des zeitlichen Verhaltens eines Systems verstanden. Die meisten Modelle werden mit Hilfe von physikalischen Gesetzen gebildet. Mathematische Modelle basieren auf Differential- oder Differenzgleichungen. Darüber hinaus ist es möglich, ein System durch gemessene Daten (Ein-/ Ausgangsdaten) mathematisch zu beschreiben [7]. In der Literatur existieren viele verschiedene Methoden zur Identifikation von nichtlinearen Systemen wie Clusterverfahren und künstliche neuronale Netze. Um ein lokal lineares Neuro-Fuzzy-Modell zu bilden, wird hier der LOLIMOT angewendet.

Bei dynamischen Modellen werden zeitlich zurückliegende Werte von Ein- und Ausgängen berücksichtigt. Die Modellordnung wird von der Anzahl der Verzögerungsoperatoren (siehe Glg. 1) bestimmt. Dabei existiert ein Trade-off zwischen Komplexität und Modellordnung. Einerseits ist die hohe Modellordnung besser, um die Dynamik eines Systems zu erfassen. Auf der anderen Seite bringt die hohe (dynamische) Modellordnung hohe Komplexität mit sich und führt zum Overfitting. Ein Teilsystemausgang mit dynamischer Ordnung 2 wird folgendermaßen beschrieben:

$$\hat{y}_i(k) = w_{0,i} + w_{1,i} \cdot y(k-1) + w_{2,i} \cdot y(k-2) + w_{3,i} \cdot u(k-1) + w_{4,i} \cdot u(k-2), \quad (1)$$

wobei k die diskrete Zeit repräsentiert.

Der Ausgang des LOLIMOT-Modells ergibt sich durch Überlappung der gewichteten Ausgänge von lokalen linearen Modellen (LLM). Der globale Ausgang ist wie folgt definiert:

$$\hat{y} = \sum_{i=1}^M \underbrace{(w_{0,i} + w_{1,i}u_1 + \dots + w_{n,i}u_n)}_{\hat{y}_i} \cdot \Phi_i(\mathbf{u}). \quad (2)$$

Dabei sind $w_{0,i}, w_{1,i} \dots w_{n,i}$ die Parameter des i ten linearen Teilmodells. Φ_i ist die normierte Zugehörigkeitsfunktion und sie wird wie folgt definiert:

$$\Phi_i(\mathbf{u}) = \frac{\mu_i(\mathbf{u})}{\sum_{i=1}^M \mu_i(\mathbf{u})} \implies \sum_{i=1}^M \Phi_i = 1, \quad (3)$$

$$\mu(\mathbf{u}) = \exp\left(-\frac{1}{2} \left(\frac{(u_1 - c_{1j})^2}{\sigma_{1j}^2} + \frac{(u_2 - c_{2j})^2}{\sigma_{2j}^2} + \dots + \frac{(u_n - c_{nj})^2}{\sigma_{nj}^2} \right)\right), \quad (4)$$

wobei c_i und σ_i jeweils Zentren und Standardabweichungen von Gauß-Funktionen sind.

Die Abbildung 1 veranschaulicht die Funktionsweise des LOLIMOT-Algorithmus. Der Eingangsraum des zu identifizierenden Systems wird in Hyperquader aufgeteilt. Bei jeder Iteration werden diese Hyperquader in achsenorthogonaler Weise halbiert. Dadurch entstehen die Partitionen, in deren Zentren die Zugehörigkeitsfunktionen (Gauß-Funktionen) positioniert werden. Die Parameter der Teilmodelle werden mit der Methode der kleinsten Quadrate optimiert und festgestellt. Danach wird bestimmt, welches der geteilten Modelle den kleinsten Ausgangsfehler aufweist. Bei dem Modell wird die Partition mit dem größten Prädiktionsfehler (Abbildung 1 blau markiert) festgestellt und bei der nächsten Iteration in alle Achsrichtungen halbiert. Dieser Vorgang wird wiederholt, bis der Algorithmus konvergiert, bzw. bis ein Abbruchkriterium erfüllt wird [8].

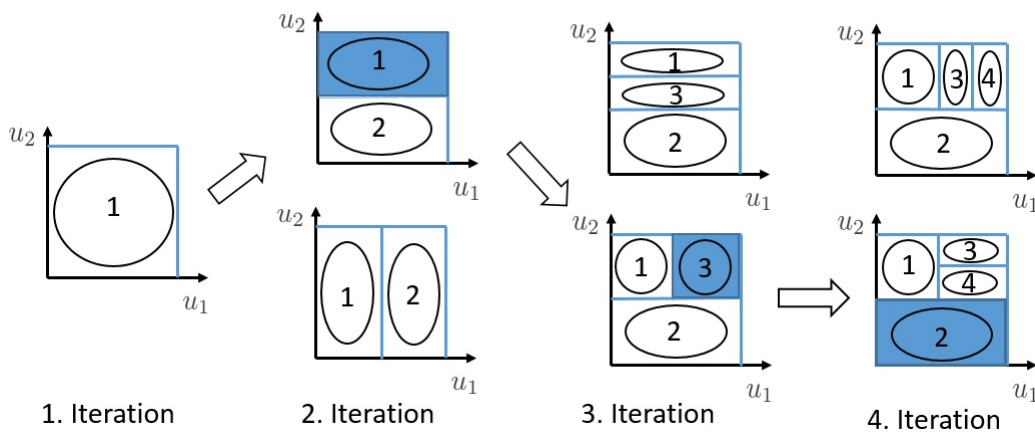


Abbildung 1: Ablauf des LOLIMOT-Algorithmus

Beispiel: Identifikation eines Zweitanksystems

Es handelt sich um zwei baugleiche Wassertanks (siehe Abbildung 2). Die Querschnittsflächen der Auslassventile und die Tankquerschnittsflächen sind identisch.

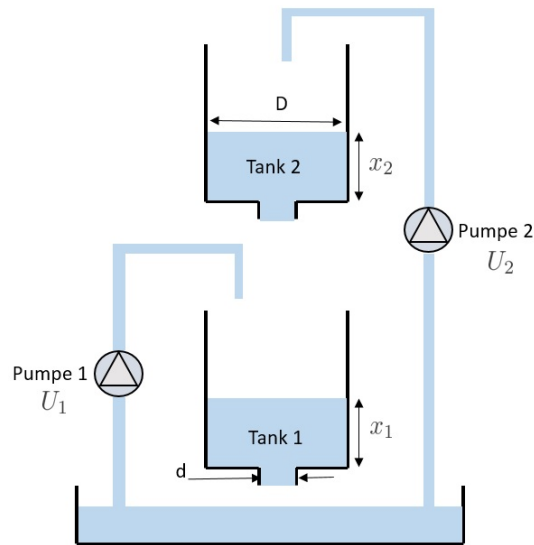


Abbildung 2: Schematische Darstellung des Zweitanksystems

Die Systemgleichungen des Modells lauten:

$$\dot{x}_1(t) = -\alpha\sqrt{x_1(t)} + \alpha\sqrt{x_2(t)} + \beta u_1(t), \quad (5)$$

$$\dot{x}_2(t) = -\alpha\sqrt{x_2(t)} + \beta u_2(t). \quad (6)$$

Dabei entsprechen die Stellgrößen der Spannungen, die die Elektromotoren der Pumpen treiben. Als Ausgang wird die Füllstandhöhe des ersten Tanks $y_1(t) = x_1(t)$ betrachtet.

Bei der datengetriebenen Systemidentifizierung wird davon ausgegangen, dass die Strecke nicht bekannt ist. Mit Hilfe von LMN-Toolbox [9] werden quantitative Abhängigkeiten der Ausgangs- von den Eingangsgrößen abgeleitet. Dabei spielt das Trainingssignal, mit dem das System angeregt wird, eine wichtige Rolle. Um den Algorithmus mit einem geeigneten Anregungssignal zu trainieren, braucht man Wissen über den Einsatzbereich des Modells. Es gibt Standardtestsignale wie gefilterstes weißes Rauschen, Pseudo-Zufallssignale (PRBS), überlagerte Sinussignale und Chirp-Signale [8].

Bei den Untersuchungen wurde für die Spannung der ersten Pumpe u_1 ein Chirp-Signal und für die Spannung der zweiten Pumpe u_2 ein zweiteiliges Chirp-Signal eingegeben, dessen Frequenz zuerst niedriger und danach höher wird. Des Weiteren wird das durch den LOLIMOT erstellte TS-Fuzzy Modell mit den vom Algorithmus unbekanntem Eingangssignalen getestet (siehe Abbildung 3).

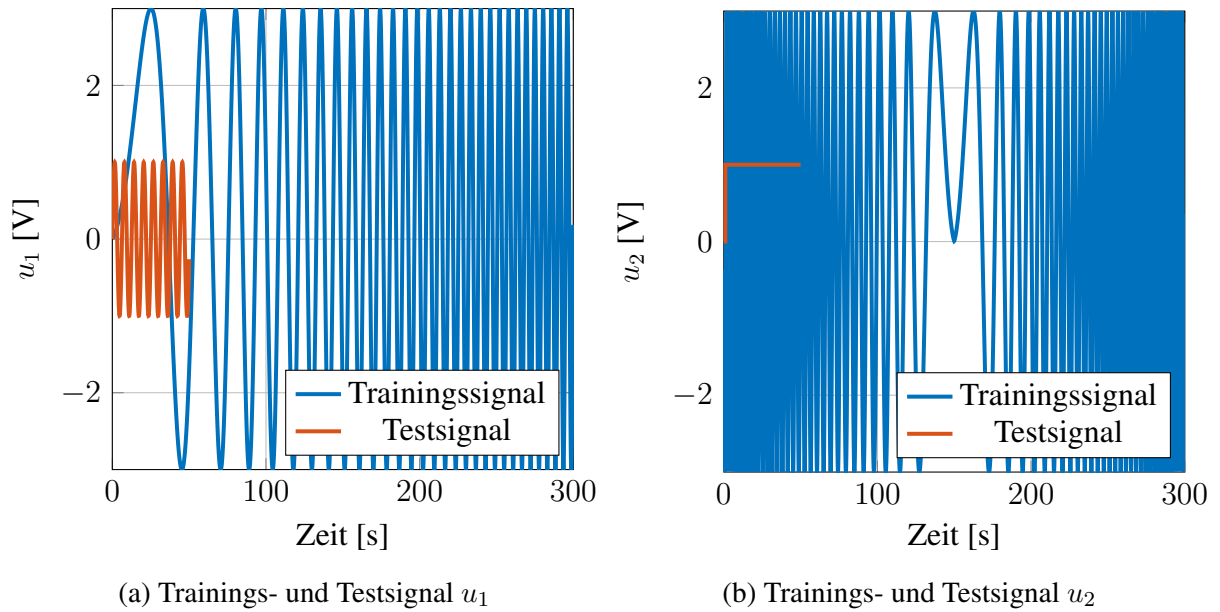


Abbildung 3: Beispiel Signale

Die Abbildung 4 zeigt einen Teil des Zustandsraums des Zweitanksystems. Bei dem Trainingsvorgang ist es prinzipiell das Ziel, alle Zustandsgrößen gut anzuregen, bzw. einen möglichst großen Bereich im Zustandsraum abzudecken. Dabei lässt sich der Zustandsraum durch das Trainingssignal relativ gut abdecken. In der Abbildung 5 wird der Ausgang des Zweitanksystems nach der Anregung mit den Trainings- und Testsignalen gezeigt.

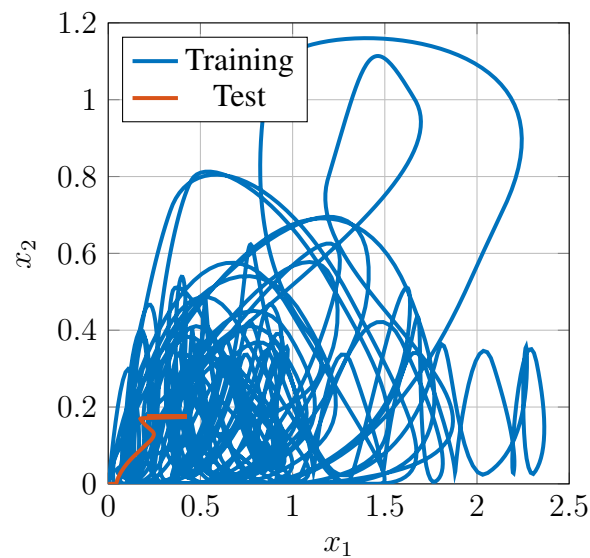


Abbildung 4: Abdeckung des Zustandsraum des Zweitanksystems

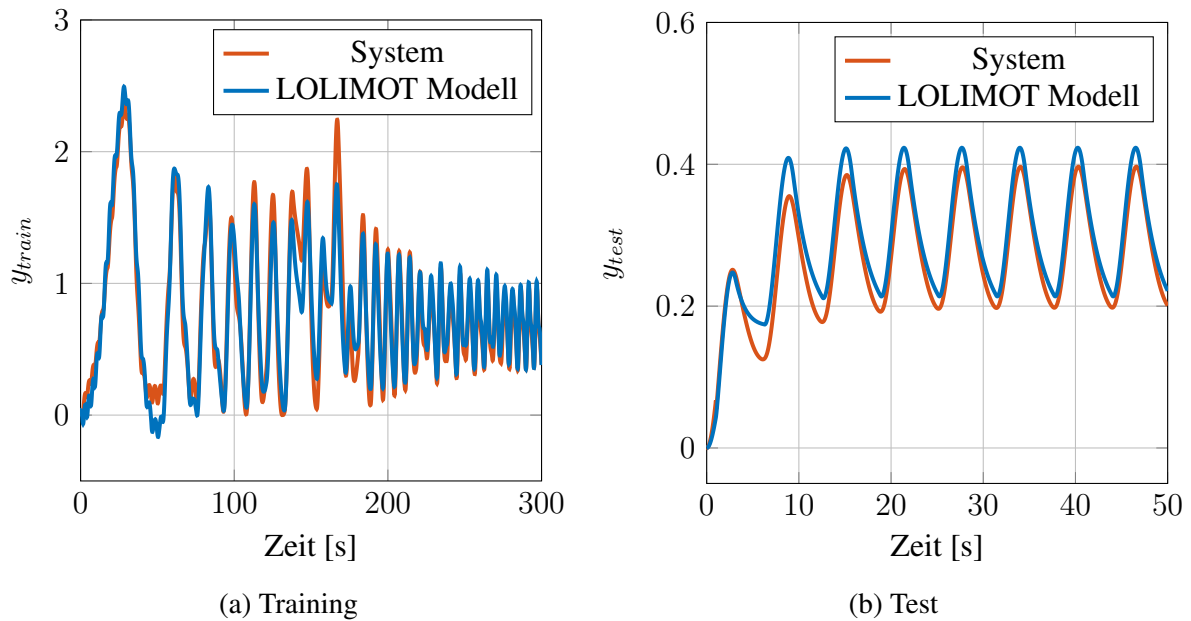


Abbildung 5: Ausgang des Zweitanksystems

4.2 Erweitertes Innovation Community Modell

Das erweiterte Simulationsmodell basiert auf der Arbeit [2], in der ein autonomes Community Modell zu Beobachtungszwecken von komplexen Innovationsverhalten entwickelt wurde. Innovation Communities bestehen aus Teilbereichen, wie z.B. dem öffentlichen und geschlossenen Bereich. Der geschlossener Bereich ist bekannt für Personen mit hoher Innovationsleistung, die effektiver Innovationen durchführen.

In den Innovation Communities befinden sich miteinander interagierende Individuen (Agenten), die unterschiedliche Eigenschaften und Ziele haben. Die Individuen sind Teilsysteme, die unabhängig voneinander betrachtet werden. Die Dynamik eines solchen Systems ist schwer mathematisch zu formulieren. Daher ist die agentenbasierte Modellierung besonders geeignet, um die zeitliche Veränderung eines heterogenen Systems abbilden zu können.

Innerhalb einer Community wird das Wissen zwischen den Community-Mitgliedern geteilt. Durch Wissens- und Erfahrungsaustausch entstehen kumulative Innovationen. Eine Person kann gute Ideen haben. Ohne genügend Wissen kann sie sie aber nicht realisieren. Die Hauptmotivation für die Zusammenarbeit von Mitgliedern ist ihr Mangel an Wissen. Die Arbeit [2] behandelt die Zugangskriterien in den Community-Bereichen und ihre Konsequenzen.

In dem Modell haben alle Agenten, also alle Community-Mitglieder, eine spezifische Aufgabe pro Zeitpunkt. Die Aufgabe ist, Innovationsanforderungen zu erfüllen. Die Agenten befinden sich am Anfang im öffentlichen Community-Bereich. Nach ihrem Erfolg und ihren gesammelten Erfahrungen in Innovation und Sozialisierung treten sie in den geschlossenen Bereich ein. Der Entscheidungsprozess für den Community-Bereich ist als ein rekurrentes Fuzzy-System implementiert, auf den hier nicht eingegangen wird. Jeder Agent besitzt spezifische Innovationsanforderungen a und Fähigkeiten f , die als Vektoren wie unten dargestellt sind.

$$\mathbf{a}_i = [a_{i,1}, a_{i,2}, \dots, a_{i,5}], \quad a_{i,j} \in [0, 10], \quad (7)$$

$$\mathbf{f}_i = [f_{i,1}, f_{i,2}, \dots, f_{i,5}], \quad f_{i,j} \in [0, 10]. \quad (8)$$

Die Elemente des Vektors stehen für die Bereiche, mit denen eine Person sich auskennt und Innovation erzielt. Damit eine Innovation entsteht, muss ein Agent in jedem Bereich genügend Fähigkeitswerte haben, d.h., Innovationsanforderungen müssen in jedem Bereich erfüllt werden, $f_{i,j} \geq a_{i,j}$. Dabei kollaborieren die Agenten, die ihre Innovationsanforderungen allein nicht erfüllen können, mit den anderen Agenten in demselben Community-Bereich. Durch die Zusammenarbeit teilen die Agenten temporär ihre Fähigkeiten. Somit erhalten sie meistens bessere Fähigkeitswerte und höhere Chancen bei der Erfüllung ihrer Innovationsanforderungen. Die temporären Fähigkeiten für einen Agent \mathbf{f}_{coll} entstehen beispielsweise wie folgt:

$$\mathbf{Agent1} : [f_{1,1}, f_{1,2}, \dots, f_{1,5}], \quad (9)$$

$$\mathbf{Agent2} : [f_{2,1}, f_{2,2}, \dots, f_{2,5}], \quad (10)$$

$$\mathbf{Agent3} : [f_{3,1}, f_{3,2}, \dots, f_{3,5}], \quad (11)$$

$$\mathbf{f}_{coll} = \max(f_{1,j}, f_{2,j}, f_{3,j}). \quad (12)$$

Dabei versuchen die anderen Agenten auf die gleiche Weise ihre Innovationsziele zu erreichen. Wenn ein Agent seine Innovationsanforderung erfüllt, wird die Kollaboration aufgelöst. Im Modell kann man noch die Anzahl der Experten definieren. Ein Expert-Agent hat hohe Fähigkeitswerte (7, 8, 9) in zwei beliebigen Bereichen. Die Nicht-Expert-Agenten haben durchschnittliche Fähigkeitswerte (3, 4, 5, 6) in jedem Bereich. Die Suche nach einem Innovationspartner für eine Zusammenarbeit findet zufällig statt und die Kollaborationsanzahl ist auf drei Personen begrenzt. Wenn trotzdem keine Innovation geschaffen wird, wird die aktuelle Kollaboration aufgelöst und ein neues Innovationsziel gesetzt.

Einerseits sind die Innovationen ein wichtiger Erfolgsfaktor, andererseits sind sie mit hohen Kosten verbunden. Eine mögliche Regelungsaufgabe wäre, durch Handlungsempfehlungen die Innovationen in einem bestimmten Bereich zu halten. Das autonome Modell lässt sich daher für regelungstechnische Zwecke erweitern, indem beeinflussbare Größen hinzugefügt werden. An dieser Stelle kommen die Fähigkeiten in Frage, die einmal initialisiert werden und danach fest bleiben. Die Idee bei der Erweiterung um Stellhebel ist, dass die Community-Mitglieder zu jedem Zeitpunkt in jedem Bereich Schulungen bekommen und dadurch ihre Fähigkeiten für einen bestimmten Zeitpunkt erhöhen (siehe Glg. 14). Das hat zur Folge, dass die Anzahl der entstehenden Innovationen gezielt beeinflusst werden kann.

$$\mathbf{u}(t) = [u_1(t), u_2(t), u_3(t), u_4(t), u_5(t)], \quad u_i(t) \in [0, 3], \quad (13)$$

$$\mathbf{f}_i(t) = [f_{i,1} + u_1(t), f_{i,2} + u_2(t), \dots, f_{i,5} + u_5(t)]. \quad (14)$$

Das Community-Modell, das in dieser Arbeit mit Hilfe des LOLIMOT-Algorithmus in ein TS-Fuzzy umgewandelt wird, hat 50 Mitglieder. Davon sind 30% Experten. D.h., sie haben hohe Fähigkeitswerte (7, 8, 9) in zwei beliebigen Bereichen und niedrige Fähigkeitswerte (0, 1, 2) in den restlichen drei beliebigen Bereichen. Die Nicht-Experten haben besitzen beliebige mittelwertige Fähigkeitswerte (3, 4, 5, 6) in 5 Bereichen.

Man unterscheidet zwischen deterministischen und nicht-deterministischen Systemen. Der LOKALIMOT-Algorithmus ist für deterministische Systeme geeignet, weil es ein offline Verfahren ist. Das agentenbasierte Modell wurde mit Pseudozufallszahlen generiert. Daher wird bei jeder Simulation die gleiche Folge von Zahlen geliefert, d.h., es wird davon ausgegangen, dass die Anfangswerte (Fähigkeiten und Innovationsanforderungen) und Interaktionen zwischen den Agenten bekannt sind. Dabei wird als Ausgang der gleitende Mittelwert der Innovationsleistung, die die Anzahl von Innovationen pro Zeitpunkt beschreibt, berücksichtigt. Der Grund dafür ist, dass ein wertkontinuierlicher Ausgang für das Identifikationsverfahren und später für die Regelaufgaben sinnvoll ist. Des Weiteren ist es eine globale Größe, die die gesamte Community betrifft. Der arithmetische Mittelwert entsteht durch die Aufsummierung der Gesamtanzahl der Innovationen pro Zeitpunkt und wird dann durch die Anzahl an vergangene Zeitpunkte geteilt. Dabei konvergiert der Ausgang nach einer gewissen Zeit zu einem bestimmten Wert. Um den Fall zu verhindern, wird dynamisch nur der Mittelwert der Innovationsleistungen von n aufeinanderfolgenden Datenpunkten in Betracht gezogen. Der gleitende Mittelwert lässt sich wie folgt berechnen:

$$m_{MA}^{(n)}(t) = \frac{1}{n} \sum_{i=0}^{n-1} x(t-i). \quad (15)$$

Um das Community Modell zu identifizieren, muss man wie beim Zweitank-System ein geeignetes Trainingssignal eingeben. Die Eingänge in dem Modell sind die Schulungen in fünf Bereichen. Das Trainingssignal besteht aus drei Teilen, bzw. gefiltertes weißes Rauschen, Chirp-Signal und zufällige Step-Funktionen zwischen den Werten 0 und 3. Als Ausgang wird der gleitende Mittelwert der Innovationsleistung der Ordnung zehn $\bar{n}_{MA}^{(10)}$ betrachtet.

Genauso wie bei dem Zweitanksystem ist hier auch die dynamische Modellordnung wichtig. Um die optimale Modellordnung zu bestimmen, wird eine Kreuzvalidierung durchgeführt. Dabei wird der mittlere quadratische Fehler (Normalized root mean square error - NRMSE) von dem Testdatensatz berechnet. Die Abbildung 6 zeigt, dass es bei 18 am kleinsten ist. Eine hohe Anzahl von Eingangsverzögerungen führt zu hoher Modellordnung. Dadurch steigt die Modellkomplexität. Um die Modellkomplexität gering zu halten, wird Modellordnung als 10 gewählt.

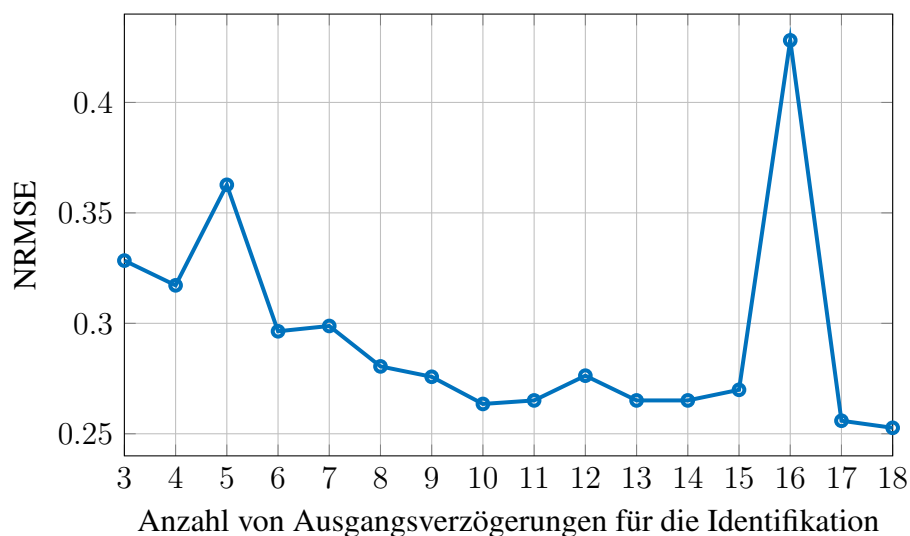


Abbildung 6: Kreuzvalidierung

Die Abbildung 7 zeigt, dass der Ausgangsverlauf von dem Trainingseingang schwer approximiert wird, d.h., die Approximierung von Verläufen mit abrupten Änderungen ist keine einfache Approximierungsaufgabe für den LOLIMOT-Algorithmus. Durch Trainieren des Algorithmus wird das Community Modell mit lokal affinen Modellen genähert.

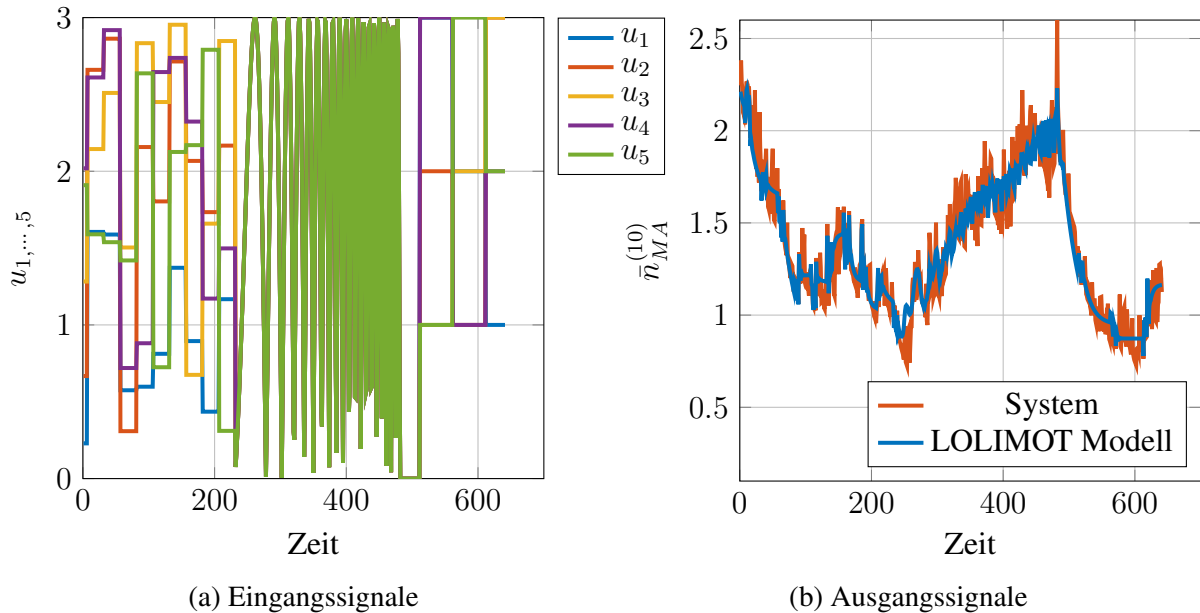


Abbildung 7: Trainingsdaten

Das durch den LOLIMOT identifizierte Modell wurde anschließend mit verschiedenen Step-Signalen (siehe Abbildung 8-a) getestet. Die Abbildung 8-b demonstriert, dass die Approximation trotz dem nicht glatten Ausgangsverlauf des Systems relativ gut ist.

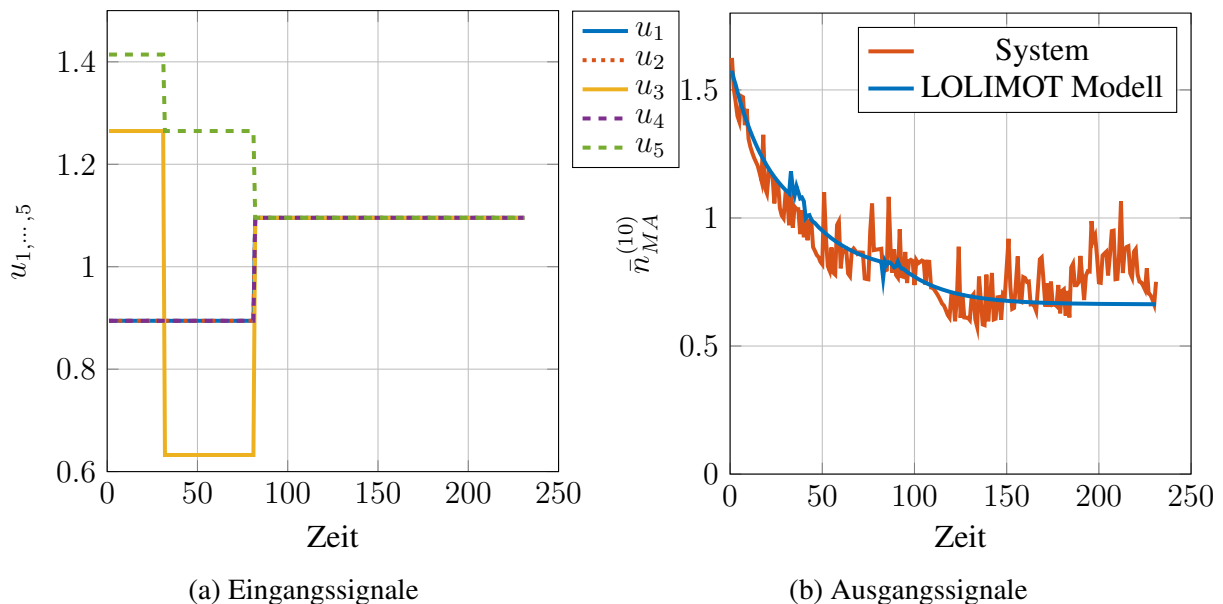


Abbildung 8: Testdaten

5 Zusammenfassung

In dieser Arbeit wird ein Innovation Community Modell vorgestellt und auf dieses Modell wird eine datengetriebene Systemidentifikationsmethode angewendet. Das hier betrachtete Modell ist ein agentenbasiertes Simulationsmodell, das die Entstehung von Innovationen vereinfacht darstellt. Das vorgestellte Identifikationsverfahren LOLIMOT basiert auf der Idee, ein nichtlineares System aus mehreren linearen Modellen stückweise nachzubilden. Dabei spielen Trainingssignale eine wichtige Rolle. Die Anregung möglichst vieler Zustände beim Training ermöglicht es, den Ausgang unbekannter Testsignale besser abzubilden. An dieser Stelle empfiehlt sich eine weitere Untersuchung über die Identifikation des Innovation Community Modells. Es wäre sinnvoll, in dem Modell virtuelle Zustände zu definieren. Darüber hinaus könnte man versuchen, mit besser geeigneten Trainingssignalen diese Zustände abzudecken. Die Ergebnisse und die Grundidee der Arbeit können als Grundlage für die Identifikation und den Reglerentwurf für ein nicht-technisches System dienen.

Danksagung: Dieser Beitrag entstand im Rahmen des Teilprojekts A3 - SFB 768. Der Autor dankt Christian Dengler für die wertvollen Diskussionen.

Literatur

- [1] VOGEL-HEUSER, B. ; LINDEMANN, U. ; REINHART, G.: *Innovationsprozesse zyklensorientiert managen: Verzahnte Entwicklung von Produkt-Service Systemen*. Springer Vieweg, 2014. – ISBN 978-3-662-44931-8
- [2] ZAGGL, M. ; STAHL, B. ; ZHONG, Z.: An Agent-Based Model Using Fuzzy Logic and Qualitative Empirical Data. In: *Proceedings 29th European Conference on Modelling and Simulation* 45 (2015), Nr. 4, S. 50–56. ISBN 978-0-9932440-0-1
- [3] NELLES, O.: Lolimot – Lokale, lineare Modelle zur Identifikation nichtlinearer, dynamischer Systeme. In: *Automatisierungstechnik* 45 (1997), Nr. 4, S. 163–174. <http://dx.doi.org/10.1524/auto.1997.45.4.163>. – DOI 10.1524/auto.1997.45.4.163
- [4] GILBERT, N.: *Agent-Based Models*. SAGE Publications, 2007 (Quantitative Applications in the Social Sciences). – ISBN 978-1412949644
- [5] FICHTER, K. ; BEUCKER, S.: *Innovation Communities – Promotorennetzwerke als Erfolgsfaktor bei radikalen Innovationen*. Fraunhofer IRB Verlag, 2008. – ISBN 978-8167-7525-6
- [6] HOFMANN, S.: *Identifikation von nichtlinearen mechatronischen Systemen auf der Basis von Volterra-Reihen*. Dissertation, Technische Universität München, 2003
- [7] HELLENDORF, N. ; DRIANKOV, D.: *Fuzzy Model Identification*. Springer-Verlag, 1997. – ISBN 3-540-62721-9
- [8] KROLL, A.: *Computational Intelligence*. Oldenbourg Wissenschaftsverlag, 2013. – ISBN 978-3-486-70976-6

- [9] HARTMANN, B. ; EBERT, T. ; FISCHER, T. ; BELZ, J. ; KAMPMANN, G. ; NELLES, O.:
LMNtool - Toolbox zum automatischen Trainieren lokaler Modellnetze. In: 22. *Workshop
Computational Intelligence, Dortmund* (2012)