

Systemkomplexität in der Automation beherrschen

Intelligente Assistenzsysteme unterstützen den Menschen

Die Integration der Informations- und Kommunikationstechnologien (IKT) und neue Anforderungen der Produktionstechnik erhöhen die Komplexität der Automation und überfordern den Menschen, womit die Arbeitseffektivität sinkt. Um dieser Situation zu begegnen, werden in diesem Beitrag intelligente Assistenzsysteme hinterfragt, die die Menschen bei fortschreitender Systemkomplexität während der Inbetriebnahme und beim Betrieb unterstützen und nicht weiter von wertschöpfenden Tätigkeiten abhalten sollen. Am Beispiel der Lemgoer Modellfabrik als Forschungsplattform werden typische Assistenzfunktionen für die Selbstkonfiguration, Selbstdiagnose und Selbstoptimierung in der Automation vorgestellt sowie deren notwendige Fähigkeiten analysiert.

SCHLAGWÖRTER Assistenzsysteme / Intelligente technische Systeme / Beherrschung der steigenden Komplexität

Intelligent Assistance Systems to control the Complexity of industrial Automation Systems

Increasing integration of ICT and challenging requirements to production systems increase the complexity of automation systems continuously. This leads to an excessive demand of the involved people with the result of a decreasing work efficiency. To counter this situation, in this paper the usage of intelligent assistant systems is described, to support human users of automation systems to handle increasing system complexity and to avoid non-value adding activities. With the Lemgo smart factory as an research and demonstration platform typical assistance functions for self-configuration, self-diagnosis and self-optimization of industrial automation systems are described and its requirements are analyzed. Important research questions in this field are addressed.

KEYWORDS assistance systems / intelligent technical systems / control the increasing system c omplexity

Steigende Anforderungen aus der Produktionstechnik und die fortschreitende IKT-Integration führen zu einem erheblichen Anstieg der Komplexität der Automatisierungssysteme [1]. Heute werden im Maschinenbau schon bis zu 50 Prozent des gesamten Entwicklungsaufwands investiert, um die notwendige Software zu erstellen [2]. Forschungsrichtungen, wie zum Beispiel das Internet der Dinge, Industrie 4.0 oder Cyber Physical Systems (CPS), werden die Systemkomplexität aufgrund der fortschreitenden IKT-Integration und der Vernetzung der Systeme noch verstärken [3].

Diese Ausgangssituation löst eine zunehmende Überforderung der Menschen aus, die mit diesen Systemen, zum Beispiel für die Aufgabenfelder Programmierung, Bedienung, Instandhaltung und Service, arbeiten. Das senkt die Arbeitseffektivität der Benutzer von Automatisierungssystemen im Entwurf, im Aufbau und in der Wartung von Anlagen. Die Folge: lange Inbetriebnahmezeiten im Anlagenhochlauf, hohe Fehleranfälligkeit durch verbleibende Hard- und Softwarefehler und damit verbunden die Gefahr von Stillstandzeiten sowie ein häufig nicht ressourcenoptimierter Betrieb der automatisierten Anlagen. Ähnliche Problemstellungen sind auch aus anderen hochtechnisierten Bereichen bekannt; die Gesellschaft für Informatik (GI) sieht dringenden Handlungsbedarf zur besseren Beherrschbarkeit komplexer technischer Systeme [4].

Dieser steigenden Komplexität der Automatisierungssysteme lässt sich auf zweierlei Weise begegnen:

- 1 | Reduktion der Komplexität der Automatisierungssysteme durch Reduzierung der enthaltenen Systemelemente und deren Relationen. Dieser Weg ist in Anbetracht der steigenden Anforderungen nicht vielversprechend.
- 2 | Die wachsende Systemkomplexität zuzulassen, aber gleichzeitig deren Wirkung auf den Menschen durch neuartige Assistenzsysteme als weniger kompliziert erscheinen zu lassen.

Als Lösung bietet sich der Einsatz von intelligenten Assistenzsystemen an. Der Begriff intelligent wird dabei aus dem Forschungsgebiet der Künstlichen Intelligenz und der Kognition entnommen und auf die industrielle Automation angewendet. Intelligenz bezeichnet dabei nach dem amerikanischen Forscher Marvin Lee Minsky die Fähigkeit von Assistenzsystemen, „Aufgaben zu lösen, zu deren Lösung Intelligenz notwendig ist, wenn sie vom Menschen durchgeführt werden“. Intelligente Assistenzsysteme sollen also zunehmend Aufgaben von menschlichen Experten übernehmen, um diese zu unterstützen oder zu entlasten.

Das Grundprinzip ist in Bild 1 zu sehen: Informationen über das technische System und dessen aktuellen Zustand werden von der Automatisierungstechnik erfasst (linke Seite von Bild 1) und müssen vom Anwender verstanden werden (rechte Seite von Bild 1), zum Beispiel, um Fehler oder ein Optimierungspotenzial zu erkennen. Da diese Informationen immer komplexer werden, sollen Assistenzsysteme dem Anwender helfen, die Informationen zu interpretieren und zu nutzen (Mitte Bild 1). Intelligente Assistenzsysteme verstecken die Systemkomplexität vor dem Anwender und erlauben eine abstraktere, das heißt menschenzentrierte, Sicht auf die Systeme. In diesem Beitrag geht es vor allen Dingen um Assistenzsysteme, die zur Laufzeit einer automatisierten Maschine oder Anlage zum Einsatz kommen und nicht um unterstützende Systeme in der Planungs- und Entwurfsphase.

Die Produktionstechnik ist seit Längerem ein wichtiges Anwendungsgebiet für intelligente oder kognitive technische Systeme, wobei die Intelligenz beziehungsweise Kognition dabei zumeist in den Automatisierungssystemen umgesetzt wird. Der gemeinsame Nenner dieser Arbeiten ist nach [8]: „[...] the application of Artificial Intelligence (AI) methods and techniques.“

Der Schwerpunkt der meisten Arbeiten in diesem Gebiet liegt auf Planungs- und Optimierungsschritten, zumeist auf Ebene der Manufacturing-Execution-Systeme (MES) und vor allem unter Verwendung von symbolischem Wissen über die Produkte und die Produktions-

technik [7, 8]. Basis sind im Allgemeinen kognitive Referenzarchitekturen wie SOAR [7], oft unter Erweiterung um Architekturen mit mehrschichtigen, aufeinander aufbauenden kognitiven Fähigkeiten [9,10]. Der Lernaspekt zielt in der Mehrheit auf Parameteroptimierungen oder Regleranpassungen [11], das Lernen von symbolischem Wissen ist eher selten anzutreffen [15]. Warum gerade das Lernen ein offenes Problem darstellt, erklärt zum Beispiel Brachman [12]: „Learning in the context of a full-blown cognitive system is not a unitary thing. There are many types of learning – whether or not they are based on some common mechanism, I can't tell you – but skill learning and language learning and discovering patterns in data and learning to build things are all different.“

Generell fehlt eine Kategorisierung der in der Automation benötigten kognitiven Fähigkeit inklusive der Abbildung auf die entsprechenden Anwendungsfälle und Herausforderungen in der Automation. Zur Veranschaulichung eignen sich entsprechende Forschungs- und Demonstrationsplattformen, wie beispielsweise die Smartfactory^{KL} des Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI) in Kaiserslautern [5] oder die gemeinsam von Fraunhofer und der Hochschule Ostwestfalen-Lippe betriebene Lemgoer Modellfabrik [6].

1. KONFIGURATIONSASSISTENTEN UND ADAPTIVITÄT

Während moderne Produktionsanlagen aus Sicht des Maschinenbaus oft schon modular aufgebaut sind und dadurch schnelle Rekonfigurationen prinzipiell unterstützen, stellen die statisch programmierten Automatisierungssysteme den Engpass dar.

So müssen bei einem Anlagenumbau Steuerungsprogramme umprogrammiert, Netzwerke neu konfiguriert und Leitsysteme angepasst werden. Diese Schritte erfolgen häufig direkt an der Maschine oder Anlage und kosten Zeit und bergen das Risiko von Fehlern. Intelligente Konfigurationsassistenten sollen diese Aufgaben im Sinne eines Plug-and-Play oder Plug-and-Produce (PnP) in Zukunft weitestgehend automatisch durchführen und damit die Inbetriebnahme- und Umbauphasen von Anlagen signifikant verkürzen (Bild 3).

Assistenzfunktionen im Bereich der Selbstkonfiguration sind ein Beispiel für intelligente technische Systeme mit der Fähigkeit zur Adaptivität: Adaptive Systeme sind in der Lage, Lösungsstrategien in sich ändernden Umgebungen selbstständig zu entwickeln. Dies steht im Gegensatz zum aktuellen Stand der Technik, so werden zum Beispiel in der Automation Lösungsstrategien noch fest in Form von Steuerungsprogrammen vom Menschen vorgegeben.

So besteht der Hauptunterschied zwischen der konventionellen Automation und den intelligenten Konfigurationsassistenten in der Verwendung von explizitem Domänenwissen durch die Assistenzsysteme (siehe auch Bild 2). Solch explizites, formalisiertes Domänenwissen, zum Beispiel über die Maschine oder Anlage, über das Automatisierungssystem oder über physikalische Hintergründe, ist von Rechnern verarbeitbar. Hierzu werden Methoden des maschinellen Schließens ver-

wendet [13]. So können die Algorithmen mit deskriptiven Zielvorgaben (wie „Produziere ein Produkt in einer vorgegebenen Güte und Zeit“) arbeiten, anstatt auf heute übliche statische, durch den Menschen vorgegebene Handlungsabläufe (beispielsweise „bewege Roboter auf Position A, starte Transportband für 5 sec,“) angewiesen zu sein. Es werden also statische, lösungsorientierte Handlungsvorgaben durch abstrakte ergebnisorientierte Zielvorgaben ersetzt. Das erhöht die Flexibilität der Automatisierungssysteme und verringert den Aufwand beim Menschen.

Die Konfigurationsunterstützung umfasst dabei zwei Schritte:

- 1 | Zuerst muss die Planung des angestrebten Produktionsablaufes auf MES-Ebene (MES: Manufacturing-Execution-System) durchgeführt werden [9,11]. Der Produktionsablauf legt fest, in welcher zeitlichen Reihenfolge die Einzelmaschinen arbeiten, um das Endprodukt zu erzeugen. Aktuell werden Beschreibungen der zur Verfügung stehenden Ressourcen/Geräte und die Zielvorgabe als Modell genutzt. Algorithmisch kommen zumeist Suchverfahren, maschinelles Schließen und Optimierungsverfahren zum Einsatz.
- 2 | Im nächsten Schritt identifiziert das Automatisierungsmodul, bestehend aus Steuerung, Feldgeräten und so weiter, selbstständig seine Rolle im Gesamtablauf der Produktion inklusive aller benötigter Kommunikationsbeziehungen für die notwendige Systemintegration [22, 23, 28].

Für die Integration von Anlagenmodulen auf Feld- und Steuerungsebene (Schritt 2) sind im Wesentlichen drei Aufgaben zu leisten:

- 1 | Herstellen einer grundsätzlichen Konnektivität durch Realisierung eines Ad-hoc-Kommunikationskanals zwischen einer Managementinstanz für die PnP-Funktionalität und dem neuen Anlagenmodul (Bild 3).
- 2 | Infrastruktur zur Herstellung einer Interaktionsfähigkeit zwischen dem Automatisierungsmodul und der Umgebung (Middleware).
- 3 | Selbstbeschreibung der Funktionalität des Anlagenmoduls (Semantik).

Zu 1: In der Automation sind Echtzeit-Ethernetsysteme (RTE) Stand der Technik. Bei jeder Anpassung der Netzkonfiguration aufgrund von Anlagenumbauten sind heute manuelle Konfigurations- und Planungsschritte notwendig, die einer Selbstkonfiguration im Wege stehen. Heutige RTE-Systeme bieten neben der zeitkritischen Prozessdatenübertragung auch einen parallel nutzbaren IP-basierten Kommunikationskanal an, der sich für den Austausch der Informationen zur Selbstkonfiguration anbietet [20].

Für diesen IP-basierten Kommunikationskanal ist zunächst die Adressvergabe und das Discovery zu lösen. Hierzu stehen auf OSI-Schicht 2 standardisierte Proto-

kolle, wie LLDP (IEEE 802.1 AB) oder Vorschläge für effiziente Autokonfigurationsverfahren auf Basis topologiebasierter MAC-Adressen [19] zur Verfügung. Auf der Netzwerkschicht sind Verfahren, wie Auto-IP oder DHCP bekannt.

Zu 2: Für die Realisierung der notwendigen Interaktionsfähigkeit bietet sich der Einsatz einer Middleware an, die eine Abstraktion von der eingesetzten Kommunikationstechnik ermöglicht und zum Beispiel über Web Services (wie OPC-UA, DPWS) standardisierte Dienste für die Erkundung der Umgebung und den Nachrichtenaustausch bietet [29, 30, 40]. Über diese Infrastruktur werden auch alle Parameter für die Konfiguration des RTE-Protokollstapels übertragen. Hierdurch kann die eigentliche Prozessdatenkommunikation mit ihren Echtzeitanforderungen über das RTE erfolgen. Diese Verfahren werden derzeit zum Beispiel in dem EU-Projekt IOT@Work [21] an der Lemgoer Modellfabrik auch vor dem Hintergrund der notwendigen IT-Sicherheit untersucht und erprobt. Der Lösungsansatz einer Trennung der Übertragung der Echtzeitdaten und der für die Selbstkonfiguration notwendigen Informationen ist mit allen RTEs möglich, die eine IP-Kommunikation zulassen, die nicht projektiert werden muss. Am Beispiel des RTE Profinet ließ sich die Tragfähigkeit nachweisen [25].

Zu 3: Für eine vollständige Selbstkonfiguration im hier dargestellten Sinne muss jedes Anlagenmodul nach dem Umbau identifizieren, welche Signale es anderen Anlagenmodulen, übergeordneten Steuerungen oder dem Leitsystem zur Verfügung stellen muss beziehungsweise welche es aus seiner Umgebung braucht. Dafür ist eine Definition der Semantik aller Signale mittels einer maschinell auswertbaren, funktionsbeschreibenden Anlagen-Modulbeschreibung notwendig. Auf der Feld- und Steuerungsebene stellt dieser Schritt noch eine offene Forschungsfrage dar [22]. Generell existieren hierzu zwei Ansätze:

- 1 | Eine Identifikation der Signale über eindeutige Namen setzt eine zentrale Planung oder eine Standardisierung der Automatisierungssysteme voraus und würde damit direkt dem Ziel der Wandlungsfähigkeit, das heißt der Unabhängigkeit bei der Entwicklung eines Anlagenmoduls, widersprechen. Daher stellen Lastra und Delamer in [31] die Frage: „How to enable two devices with no previous knowledge on each other's type, conceived using different paradigms and interaction models but still with complementary skill sets, to interact autonomously?“
- 2 | Eine Definition der Semantik aller Signale mittels eines maschinell auswertbaren, funktionsbeschreibenden Informationsmodells könnte die Funktion eines Signals im Kontext der Anlage modellieren. Ein Konfigurationsservice in der Steuerung des Anlagenmoduls könnte dann anhand dieses Informationsmodells und eines Schlussfolgerungsalgorithmus identifizieren, welche Signale es mit anderen Modulen austauschen muss. Dieser Ansatz entspricht der allgemeinen Idee von intelligenten technischen Systemen: Wissen wird explizit und damit vom Computer verarbeitbar gemacht, das heißt es entstehen adaptive und flexible Systeme.

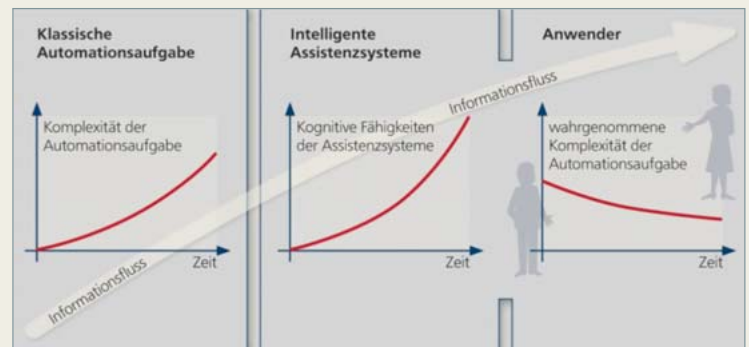


BILD 1: Assistenzsysteme unterstützen den Benutzer künftig beim Umgang mit komplexen Automationsystemen.

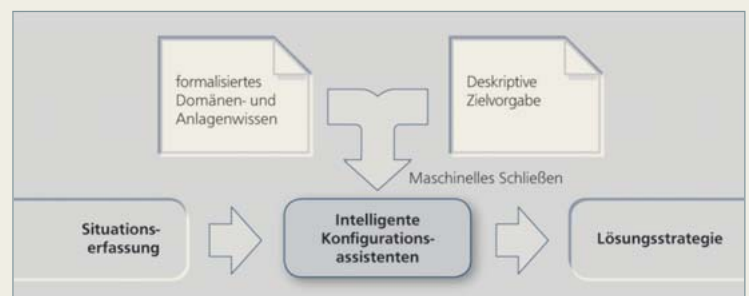


BILD 2: Das Grundprinzip adaptiver Systeme



BILD 3: Automatische Systemintegration eines Anlagenmoduls in der Lemgoer Modellfabrik

Herausforderung beim zweiten Ansatz ist die Modellierung der Semantik in Form von Informationsmodellen. Diese müssen zum einen ausdrucksstark genug sein, um die notwendige Semantik abzubilden. Andererseits müssen effiziente Schlussfolgerungsmechanismen für den Formalismus existieren. Eine allgemeine Übersicht zu Informationsmodellen findet sich in [32]. In [31] werden zum Beispiel Semantic Web Services und Web Ontology Language (OWL) als Lösung vorgeschlagen. Loskyll et al. verwenden Semantic Annotations for Web Services Description Language (SAWSDL) und OWL [33]. Einen interessanten Ansatz stellt die Verwendung von OPC-UA-kompatiblen Informationsmodellen in Kombination mit OWL dar [34]. Generell besteht also ein erster Trend bezüglich der zu verwendenden Formalismen für die Informationsmodelle, während zu den notwendigen Inhalten der Modelle und der Schlussfolgerungsmechanismen noch offene Fragen existieren.

2. DIAGNOSEASSISTENTEN UND DIE LERNFÄHIGKEIT

Intelligente Diagnoseassistenten helfen dem Benutzer, frühzeitig Probleme und Verschleiß (Anomalieerkennung) zu erkennen und entsprechende Fehlerursachen (Diagnose) zu identifizieren und bei der Anlagenreparatur. Diese drei Schritte bauen dabei üblicherweise aufeinander auf.

Generell ist in der Produktionstechnik die Zuverlässigkeit der Maschinen und Anlagen ein wichtiges Ziel. Produktionsausfall durch Anlagenstillstände führt zu hohen Kosten. Gerade in verteilten Produktionsanlagen und in vernetzten Automatisierungssystemen ist die Fehlersuche aber sehr aufwendig, da der Ort eines Fehlersymptoms nicht gleich dem Ort der Fehlerursache sein muss. Das Bedien- und Instandhaltungspersonal steht daher bei auftretenden Fehlern unter hohem Zeit- und Erfolgsdruck, um die Anlage wieder anzufahren. Hier können Diagnoseassistenten helfen (siehe Bild 4).

Grundlage der Diagnoseassistenten ist die Verfügbarkeit von formalisiertem Wissen über den automatisierten Produktionsprozess, zumeist in Form von Modellen. Diese Modelle werden zur Erkennung von Anomalien und von Fehlerursachen verwendet. Alle erkannten Probleme werden dem Fachpersonal über geeignete Mensch-Maschine-Interaktionstechnologien (Leitsysteme, mobile Plattformen) mitgeteilt.

Solche Diagnoseassistenten werden aus zwei Gründen bislang wenig in der Industrie eingesetzt: Zum einen ist die Modellerstellung arbeitsintensiv und kann oft nur von Experten ausgeführt werden. Zum anderen ist der Genauigkeit von manuell erstellten Modellen oft eine Grenze gesetzt: Anlagen verändern sich häufig, zum Beispiel durch Verschleißprozesse, durch Umwelteinflüsse oder durch Umbauten.

Hier bietet das maschinelle Lernen einen Ausweg: Durch Beobachtung des Prozesses in Echtzeit kann das Modell und damit das notwendige Diagnosewissen maschinell erlernt werden. Grundlage hierfür ist die Verfügbarkeit der Prozessdaten, die zum Betriebszeitpunkt in ausreichender Menge erfasst werden können.

Solche Assistenzfunktionen sind ein Beispiel für intelligente technische Systeme basierend auf einer Lernfähigkeit: Wie in Bild 5 zu sehen, erlernt der Diagnoseassistent, basierend auf der erfassten Anlagensituation (Situationserfassung), Wissen über das Anlagenverhalten in Normal- und Fehlersituationen und über die Wirkzusammenhänge in der Anlage.

Mittels dieses gelernten Wissens analysiert der Diagnoseassistent das Anlagenverhalten im Betrieb und erkennt Anomalien und Fehlerursachen, die dann dem Benutzer mitgeteilt werden.

Das Lernen von Modellen zur Anomalieerkennung und Diagnose ist ein typisches Anwendungsgebiet des maschinellen Lernens. Wie in Bild 6 zu sehen, wird generell zwischen dem Lernen von Klassifikatoren für phänomenologische Diagnoseansätze und dem Lernen von Modellen für modellbasierte Diagnoseansätze unterschieden.

- 1 | Phänomenologische Ansätze klassifizieren Beobachtungen direkt als korrekt oder inkorrekt beziehungsweise klassifizieren sie gemäß der Fehlerursachen.
- 2 | Modellbasierte Ansätze ermitteln zunächst Anomalien durch einen Vergleich eines Modells des Normalverhaltens mit aktuellen Beobachtungen. Fehlerursachen werden anschließend durch ein Suchverfahren ermittelt, wobei solange Hypothesen für die Fehlerursachen in das Modell integriert werden, bis Prognose und Beobachtung wieder übereinstimmen.

Das Lernen für phänomenologische Ansätze kann also auf das Lernen von Klassifikatoren zurückgeführt werden. Typisch für phänomenologische Ansätze ist die statische Behandlung des wichtigen Faktors Zeit: Im Allgemeinen wird zunächst ein statischer Eigenschaftsvektor berechnet, der dann als Eingabe der Klassifikationsfunktion verwendet wird. Beispiele sind Funktionsapproximationsansätze (wie Neuronale Netze, Regression, Fuzzy-basierte Ansätze), Support-Vektor-Maschinen, fallbasierte Ansätze oder regelbasierte Methoden (zum Beispiel induktive Lernverfahren oder das Lernen von Entscheidungsbäumen).

Neben der für technische Systeme unzureichenden Behandlung des Faktors Zeit haben phänomenologische Ansätze ein grundsätzliches Problem: Sie schliessen gegen die Kausalität des ursprünglichen Systems, das heißt sie schließen von Symptomen/Beobachtungen auf Anomalien und Fehlerursachen. Für komplexe Anlagen mit vielen Abhängigkeiten zwischen Komponenten und komplizierten Wirkzusammenhängen führt dies zu diversen Problemen: (1.) Die Klassifikationsalgorithmen müssen viele Signale inklusive ihrer Historie im Eigenschaftsvektor berücksichtigen. (2.) Die Klassifikationsansätze müssen eine hohe Anzahl von Wertkombinationen im Eigenschaftsvektor abbilden, dies verhindert letztendlich die Lernbarkeit dieser Modelle. Aus diesen Gründen sind phänomenologische Ansätze zwar aus Sicht der Lernbarkeit dankbare Ansätze, für komplexe technische Systeme aber oft ungeeignet.

Das Lernen von modellbasierten Ansätzen ist zunächst grundsätzlich schwieriger. Es muss ein vollständiges

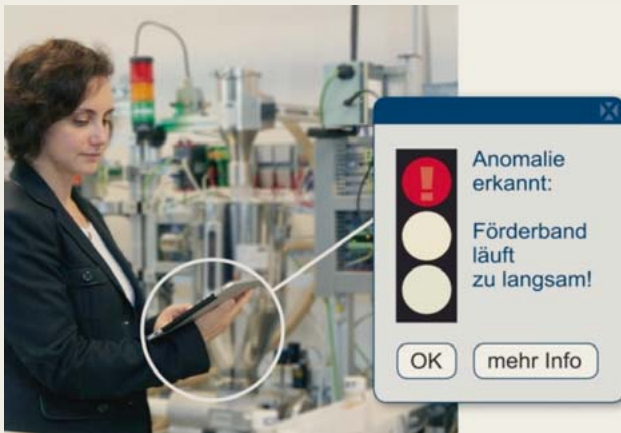


BILD 4: Unterstützung des Benutzers bei der Fehlersuche durch Diagnoseassistenten



BILD 5: Maschinelles Lernen als Grundlage für Diagnose-assistenten

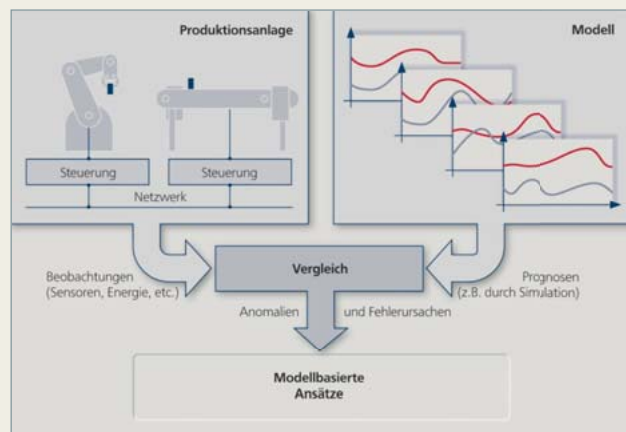
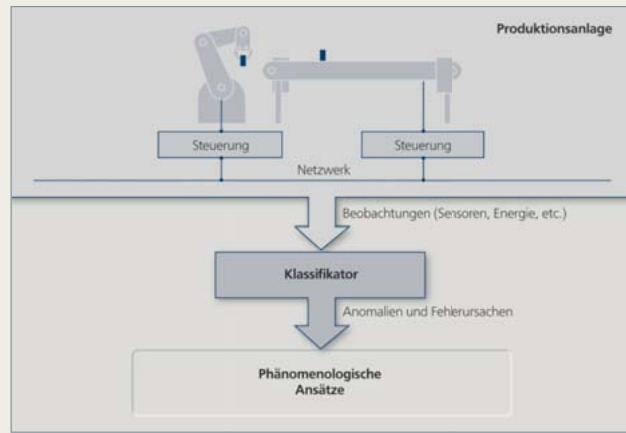


BILD 6: Phänomenologische (oben) und modellbasierte Ansätze (unten) zur Anomalieerkennung und Diagnose

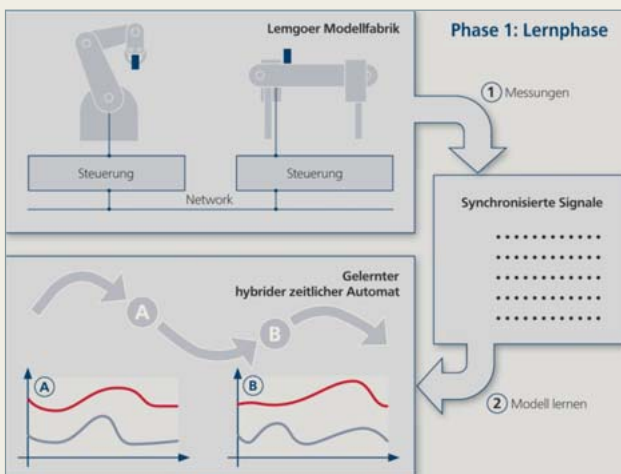


BILD 7: Lernphase des Assistenten zur Anomalieerkennung

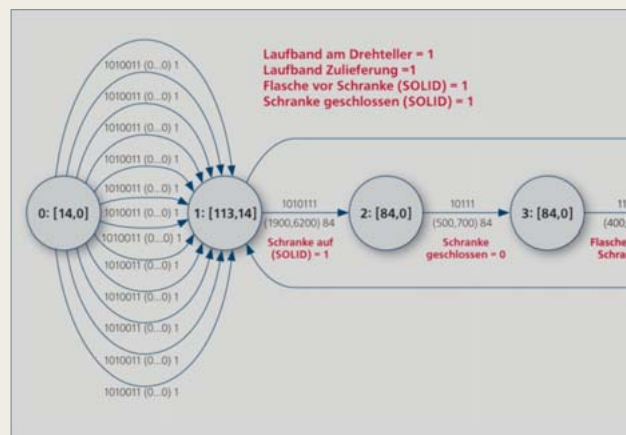


BILD 8: Ausschnitt eines gelernten hybriden zeitlichen Automaten für den Abfüllprozess der Lemgoer Modellfabrik

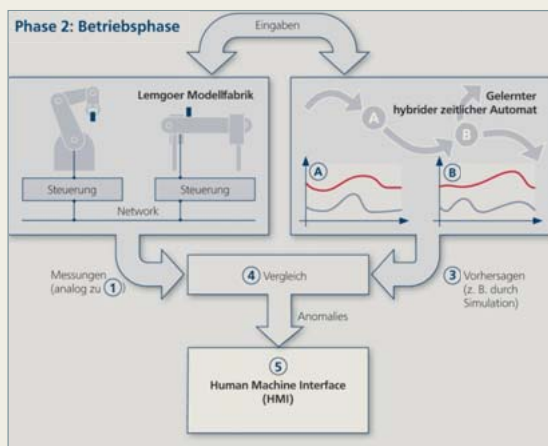


BILD 9: Diagnoseassistent aus den Projekten Initial und AVA [16, 17]



BILD 10: Energieoptimierter Betrieb eines Hochregallagers in der Lemgoer Modellfabrik

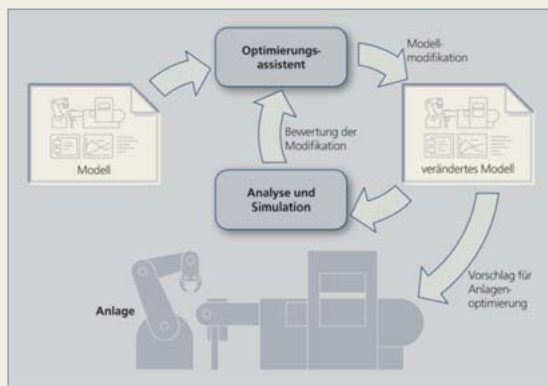


BILD 11: Prinzip der Optimierungsassistenten

Modell des Systemverhaltens gelernt werden und dies nur durch Beobachtung des Systems. Auf der anderen Seite sind solche Modelle aber in der Lage, das Zeitverhalten genau abzubilden (zum Beispiel in Form von Differenzialgleichungen) und eine große Kombination von Wirkzusammenhängen kompakt zu erfassen (beispielsweise in Form von endlichen Automaten).

Bisherige Arbeiten zum Modelllernen haben sich auf die Parametrisierung von vorgegebenen Differenzialgleichungssystemen [35], auf das Lernen von Regeln für diskrete Ereignisse [36, 38] oder auf die Ermittlung von lokalen Zusammenhängen zwischen kontinuierlichen Variablen eines Teilsystems [37] konzentriert. Für komplexe technische Systeme liegt daher die aktuelle Herausforderung auf dem Lernen von Modellen mit den folgenden Eigenschaften:

- 1 | Es sollte möglichst wenig A-priori-Wissen zum Lernen notwendig sein. Optimal ist eine Beschränkung auf die Erfassung von asynchronen Teilsystemen. Daraus folgt, dass eine reine Modellparametrisierung unzureichend ist.
- 2 | Das Zeitverhalten sollte explizit modelliert sein.
- 3 | Es sollten hybride Systeme, das heißt Systeme mit Wert-diskreten und Wert-kontinuierlichen Variablen, erfasst werden. Hierbei ist zu beachten, dass Wert-diskrete Variablen oft radikale Änderungen des Systemverhaltens (Mode-Wechsel, [39]) nach sich ziehen. Beispiele sind das Öffnen eines Ventils oder das Einschalten eines Antriebs.

In den Projekten Initial und AVA wurden entsprechende Ansätze an der Lemgoer Modellfabrik und an realen Industrieanlagen entwickelt und validiert [16, 17]. Der Fokus lag dabei auf der Erkennung von Anomalien und auf dem Lernen von Modellen des Normalverhaltens mit einer expliziten Erfassung des Zeitverhaltens für hybride Systeme.

Der in diesen Projekten entwickelte Diagnoseassistent unterscheidet generell zwei unterschiedliche Phasen.

In der Lernphase (Bild 7) werden zunächst alle Daten der Anlage durch Messungen von Prozesssignalen gesammelt (Schritt 1). Herausforderungen sind hierbei die Erfassung der Daten in heterogenen, verteilten Automationssystemen und die zeitliche Synchronisation der Messstellen. Aus diesen Daten wird in Schritt 2 ein Modell des Normalverhaltens der Anlage gelernt.

Generell kann nicht das komplette Modell des Normalverhaltens erlernt werden, wo immer möglich wird daher auf A-priori-Wissen zurückgegriffen. Dieses A-priori-Wissen umfasst zumeist Strukturinformationen der Anlage und der Automationslösung, minimal müssen zumindest asynchrone Teilsysteme ausgewiesen werden. Beispiele für solches a-priori-Wissen sind vorhandene Anlagenmodelle in Form von AutomationML-Dateien [14] oder in Form von CAD-Modellen.

Das Verhalten der einzelnen, synchronen Teilsysteme wird in Form eines hybriden, zeitbehafteten Automaten gelernt [15]. Solche Automaten bilden das Normalverhalten (das heißt das fehlerfreie Verhalten) der Anlage als Abfolge von Zuständen ab, wobei innerhalb eines

Zustandes ein einfacher zeitlicher und funktionaler Zusammenhang zwischen den Signalen besteht, zum Beispiel beschreibbar durch gewöhnliche Differenzialgleichungen oder durch ein neuronales Netz. Bei dem Auftreten eines Ereignisses, wie Einschalten einer Pumpe oder Schalten einer Lichtschranke, wechselt der Automat innerhalb einer definierten Zeitspanne in einen neuen Zustand.

Bild 8 zeigt einen solchen gelernten Automaten für die Abfüllstation der Lemgoer Modellfabrik: Erkennbar sind Grundzustände (Kreise in der Abbildung) bei der Abfüllung von Schüttgut und die Zustandsübergänge, das heißt die Transitionen. An jeder Transition sind die auslösenden Ereignisse (Prozesssignale) und die Zeitspanne seit dem letzten Ereignis annotiert. Solche Automaten haben auch den Vorteil, dass Experten sie gut verstehen und somit zu Verifikationszwecken mit dem Verhalten der Anlage vergleichen können (rote Annotationen in Bild 8).

Der Diagnoseassistent kann nun in der Betriebsphase (Bild 9) das von dem gelernten Automaten prognostizierte Verhalten (Schritt 3) mit dem beobachteten Verhalten der realen Anlage vergleichen und so Anomalien erkennen (Schritt 4). Diese werden dem Benutzer über geeignete Mensch-Maschine-Interaktionstechnologien mitgeteilt (Schritt 5).

Die in [15, 17] entwickelten Algorithmen sind in der Lage, das Normalverhalten hybrider Systeme korrekt zu erlernen und so falsches Zeitverhalten aufgrund von Verschleiß, suboptimaler Energieverbräuche oder unerwarteter Prozesssignale zu erkennen und dem Benutzer frühzeitig Hinweise auf Probleme zu liefern.

3. OPTIMIERUNGSASSISTENTEN UND DIE PROGNOSEFÄHIGKEIT

Ein weiteres Anwendungsfeld sind Assistenzsysteme für die Selbstoptimierung. Sie helfen dem Benutzer dabei, die Anlagenleistung und Effizienz kontinuierlich zu analysieren, zu verbessern und einen möglichst optimalen Betriebspunkt anzustreben (Bild 10).

Hierdurch lässt sich beispielsweise der Energieverbrauch von produktionstechnischen Anlagen optimieren. Um neben der Grundfunktion auch eine energieoptimierte Betriebsführung durchführen zu können, muss ein Rechnermodell der Maschine oder Anlage aus energie- und automatisierungstechnischer Sicht vorhanden sein. Algorithmen der Selbstoptimierung übernehmen nun auf Basis dieses Modells wiederkehrend und in Echtzeit Aufgaben des Programmierers [18], indem sie das Ablaufverhalten im Modell kontinuierlich derart anpassen, dass zum einen die Grundfunktion gewährleistet bleibt und gleichzeitig die gesetzten Energieziele erfüllt werden können.

Solche Assistenzfunktionen sind ein Beispiel für intelligente technische Systeme mit einer Prognosefähigkeit. Wie in Bild 11 zu sehen ist, basiert jede automatische Optimierung eines technischen Systems auf einer Prognosefähigkeit. Nur durch die hypothetische Analyse der Auswirkung von Änderungen im Ablaufverhalten

durch Simulation der angepassten Modelle kann mittels einer gesteuerten Suche im Optimierungsraum eine gute Anlagenkonfiguration ermittelt werden.

ZUSAMMENFASSUNG

Künftige Automatisierungssysteme müssen sich selbstständig vernetzen, diagnostizieren und optimal anpassen. Hierfür existieren derzeit viele Teil-, aber noch keine ganzheitlichen Lösungen in der Automation. Technische Grundlage der vorgestellten intelligenten Assistenten sind neben einer durchgängigen Vernetzung die explizite, rechnerverarbeitbare Modellierung des Wissens der automatisierten Prozesse sowie entsprechende wissensbasierte Algorithmen zur Selbstkonfiguration, Selbstdiagnose und Selbstoptimierung.

Derzeit fehlen Modellformalismen und Semantikinformationen, die dann mittelfristig (1.) die Grundlage für das maschinelle Schließen von Anlageninformationen bilden (zum Beispiel für die Konfigurationsassistenten), die (2.) das Lernen der Modelle unterstützen (beispielsweise für die Diagnoseassistenten) und die (3.) eine Prognose des Systemverhaltens erlauben (zum Beispiel für die Optimierungsassistenten). Langfristig ermöglichen diese Grundlagen die Entwicklung der Anomalie- und Diagnosealgorithmen, der Methoden zur Anlagensynthese und der Verfahren zur Anlagenoptimierung.

Dieser Beitrag zeigt mit den intelligenten Assistenzsystemen ein Handlungsfeld auf, dessen Forschungsfragen von der automatisierungstechnischen Forschungswelt aufgegriffen werden sollten, um die zunehmende Komplexität der Automatisierungssysteme beherrschbar zu machen und damit die technologisch führende Position der deutschen Automation zu sichern und auszubauen.

MANUSKRIPTEINGANG
05.05.2012

Im Peer-Review-Verfahren begutachtet

DANKSAGUNG

Das Projekt Initial wird im Rahmen der Förderlinie „Hightech-NRW“ vom Ministerium für Innovation, Wissenschaft, Forschung und Technologie des Landes Nordrhein-Westfalen (Förderkennzeichen z0903ht015a) gefördert. Das Projekt AVA wird vom Bundesministerium für Bildung und Forschung unter dem Förderkennzeichen 17N1211 gefördert.

Die Bereiche Selbstkonfiguration, Selbstdiagnose und Selbstoptimierung von technischen Systemen sind ein Schwerpunkt des BMBF-Spitzenclusters „Intelligente Technische Systeme Ostwestfalen-Lippe it's OWL“, in dem das Fraunhofer-Anwendungszentrum für Industrielle Automation (IOSB-INA) in Lemgo eine der beteiligten Forschungseinrichtungen ist.

REFERENZEN

- [1] VDMA 2012: Management Summary zur Tendenzbefragung 2012 – Bedeutung der Informations- und Automatisierungstechnik in den Produkten des Maschinen- und Anlagenbaus, 2012. (<http://www.vdma.org/>)
- [2] BMBF: Software in Maschinen und Produkten, 2009. (<http://www.produktionsforschung.de>)
- [3] acatech (Hrsg.): agendaCPS – Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion, Dezember 2011
- [4] Gesellschaft für Informatik (GI): 38. Jahrestagung „INFORMATIK 2008 – Beherrschbare Systeme – dank Informatik“ (<http://www.informatik2008.de/371.html>)
- [5] Smartfactory am DFKI (<http://www.smartfactory-kl.de>)
- [6] Lemgoer Modellfabrik im Centrum Industrial IT, Lemgo, (<http://www.smartfactory-owl.de>)
- [7] Behnen, D.; Brecher, C.; Hammann, G.; Herfs, W.; Koeppel, R.; Pauly, D.; Possel-Dölken, F.; Verl, A.: Potenziale kognitiver Systeme in der Produktionstechnik. In: Wettbewerbsfaktor Produktionstechnik: Aachener Perspektiven, S. 101–123. Shaker, Aachen 2011
- [8] Zaeh, M.; Lau, C.; Wiesbeck, M.; Ostgathe, M.; Vogl, W.: Towards the cognitive factory. In: Tagungsband Int. Conf. Changeable, Agile, Reconfigurable and Virtual Production (CARV), S. 2–16. CARV CD Proceedings, 2007
- [9] Putzer, H.; Onken, R.: COSA – A Generic Cognitive System Architecture Based on a Cognitive Model of Human Behavior. *Cognition Technology and Work*, 5(2), S. 140–151, 2003
- [10] Frank, U.; Giese, H.; Klein, F.; Oberschelp, O.; Schmidt, A.; Schulz, B.; Vöcking, H.; Witting, K.: Selbstoptimierende Systeme des Maschinenbaus – Definitionen und Konzepte. HNI-Verlagsschriftenreihe, Paderborn 2004
- [11] Beetz, M.; Buss, M.; Wollherr, D.: Cognitive Technical Systems – What Is the Role of Artificial Intelligence? In: Conference on Artificial Intelligence (KI 2007), S. 19–42. Springer, Berlin 2007
- [12] Brachman, R.: Systems That Know What They're Doing. *IEEE Intelligent Systems*, 17(6), S. 67–71, 2002
- [13] Zaeh, M. F.; Beetz, M.; Shea, K.; Reinhart, G.; Bender, K.; Lau, C.; Ostgathe, M.; Vogl, W.; Wiesbeck, M.; Engelhard, M.; Ertelt, C.; Rühr, T.; Friedrich, M.; Herle, S.: The Cognitive Factory. In: *Changeable and Reconfigurable Manufacturing Systems*, S. 355–371. Springer, 2009
- [14] Faltinski, S.; Niggemann, O.; Moriz, N.; Mankowski, A.: AutomationML: From Data Exchange to System Planning and Simulation. In: *IEEE International Conference on Industrial Technology (ICIT 2012)*, S. 378–383. IEEE, Piscataway 2012.
- [15] Niggemann, O.; Stein, B.; Vodenčarevič, A.; Maier, A.; Kleine Büning, H.: Learning Behavior Models for Hybrid Timed Systems. To be published in: *Twenty-Sixth Conference on Artificial Intelligence (AAAI-12)*. AAAI Press, Toronto 2012
- [16] Faltinski, S.; Flatt, H.; Pethig, F.; Kroll, B.; Vodenčarevič, A.; Maier, A.; Niggemann, O.: Detecting Anomalous Energy Consumptions in Distributed Manufacturing Systems. To be published in: *IEEE International Conference on Industrial Informatics (INDIN 2012)*. IEEE Conference Publications, Beijing 2012
- [17] Vodenčarevič, A.; Kleine Büning, H.; Niggemann, O.; Maier, A.: Identifying Behavior Models for Process Plants. In: 16th IEEE International Conference on Emerging Technologies and Factory Automation ETFA'2011, Toulouse, France, 2011
- [18] Possel-Dölken, J.; Brecher, C.; Herfs, W.; Kempf, T.: Kognitive Automatisierung – Szenarien und Ansätze für die selbstoptimierende Steuerung von Abläufen. *atp – Automatisierungstechnische Praxis*, 49(12), S. 50–60, 2007
- [19] Imtiaz, J.; Jasperneite, J.; Weber, K.; Lessmann, G.; Goetz, F.: A Novel Method for Auto Configuration of Realtime Ethernet Networks. In: *13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA2008)*, S. 861–868, Hamburg, Germany, 2008
- [20] Abel, M.; Lechler, A.; Schlechtendahl, J.; Klemm, P.: Serviceorientierte Architektur für die Konfiguration und Inbetriebnahme von Produktionsmaschinen – Universal Plug and Produce. In: *8. Fachkolloquium für Angewandte Automatisierungstechnik in Lehre und Entwicklung an Fachhochschulen (AALE)*, S. 111–120. Oldenbourg Industrieverlag, Göppingen 2011.
- [21] EU-Projekt IOT@Work (<http://www.iot-atwork.de>)
- [22] Faltinski, S.; Wienke, M.; Niggemann, O.; Jasperneite, J.: mlNA – Eine echtzeitfähige Middleware für die Industrieautomation zur Realisierung semantikkbasierter Wandelbarkeit. In: *2. Jahreskolloquium Kommunikation in der Automation (KommA 2011)*, Magdeburg, Sep 2011
- [23] Hodek, S.; Schlick, J.: Ad hoc field device integration using device profiles, concepts for automated configuration and web service technologies: Plug&Play field device integration concepts for industrial production processes. In: *Proceedings 9th International Multi-Conference on Systems, Signals and Devices (SSD)*, S. 1–6. IEEE, Chemnitz 2012
- [24] Loskyll, M.; Schlick, J.; Hodek, S.; Ollinger, L.; Gerber, T.; Pirvu, B.: Semantic service discovery and orchestration for manufacturing processes. In: *Proceedings 16th IEEE Conference on Emerging Technologies & Factory Automation*, S. 543–550. IEEE, 2011.
- [25] Dürkop, L.; Trsek, H.; Jasperneite, J.; Wisniewski, L.: Towards Autoconfiguration of Industrial Automation Systems: A Case Study Using PROFINET IO. In: *Proceedings 17th International Conference on Emerging Technology & Factory Automation (ETFA)*. IEEE, 2012.
- [26] Ebel, M.; Okon, M.; Baumann, M.: ProduFlexil: Flexible Produktion mit SOA-Architektur und Plug-and-Work-Mechanismus. Technical Report, Fraunhofer Institut für Informations- und Datenverarbeitung (IITB), 2007
- [27] Jammes, F.; Smit, H.: Service-oriented paradigms in industrial automation. In: *IEEE Transaction on Industrial Informatics*, 1(1), S. 62–70, 2005
- [28] Colombo, A.: SOCRADES: Steps Towards the Factory of the Future. In: *Projects Magazine EU*, Vol(12), S. 18–21, 2009
- [29] Martinez Lastra, J.; Delamer, I.: Semantic Web Services in Factory Automation: Fundamental Insights and Research Roadmap. In: *IEEE Transactions on Industrial Informatics*, 2(1), S. 1–11, 2006
- [30] Mahnke, W.; Gösling, A.; Graube, M.; Urbas, L.: Information Modeling for Middleware in Automation. In: *Proceedings 16th IEEE International Conference on Emerging Technologies and Factory Automation*, S. 1–7. IEEE, 2010