Sprachsteuerung für einen industriellen Leichtbauroboter

Thomas Wachsmuth, Christian Thormann, Alexander Winkler
Hochschule Mittweida, Technikumplatz 17, 09648 Mittweida

In diesem Beitrag wird die Sprachsteuerung eines Leichtbauroboters vorgestellt. Im Alltag findet man viele Beispiele zur Sprachsteuerung. Darum sollte dieser Ansatz auch in der industriellen Automatisierung etabliert werden, wie z.B. in der Robotik. Leichtbauroboter eignen sich aufgrund ihrer intuitiven Programmierung sehr gut für solche Aufgaben. Zudem enthalten sie üblicherweise Sicherheitsfunktionen in Form von Geschwindigkeits- oder Kraftbeschränkungen was sie auch in Kombination mit der Spracherkennung sicherer macht. Für die Implementierung der Roboter-Sprachsteuerung musste eine geeignete Spracherkennungssoftware ausgewählt werden. Hierfür wurden verschiedene Systeme verglichen. Bei der Sprachsteuerung des Roboters wurde zwischen dem Teach-In mit Sprachbefehlen und der Bewegungssteuerung mittels Sprache unterschieden. Beide Konzepte wurden an einem Demonstrator umgesetzt und die Algorithmen in Versuchen verifiziert. Anschließend wurden das Teach-In mit Sprachbefehlen mit dem per Handbediengerät in Bezug auf die Arbeitszeit verglichen.

1. Einführung

Im Allgemeinen besteht die Programmierung industrieller Roboter aus einer Kombination von Teach-In und textueller Programmierung. Teach-In bedeutet, dass der Bediener den Manipulator zu allen Positionen bewegt, die für die Aufgabe notwendig sind. Dabei kann der Manipulator z.B. über die Tastatur oder per Joystick gesteuert. Nach dem Teach-In muss die Programmierung textbasiert vervollständigt werden. Hierfür gibt es viele verschieden Roboterprogrammiersprachen. Ein Überblick kann in [1] gefunden werden.

Seit einigen Jahren geht der Trend hin zu sogenannten Leichtbaurobotern. Das bedeutet, dass das Verhältnis von Traglast zu Eigenmasse nahezu 1:1 beträgt, im Gegensatz zu konventionellen Industrierobotern mit nahezu 1:10. Leichtbauroboter wurden ursprünglich für die Raumfahrt entwickelt. Allerdings führte die geringere Masse auch zu neuen Anwendungsgebieten, wie z.B. der Mensch-Roboter Interaktion, welche zunehmend an Bedeutung gewinnt [2]. Aus diesem Grund können gegenwärtig viele Leichtbauroboter von unterschiedlichen Herstellern am Markt gefunden werden.

Häufig können diese Leichtbauroboter zum komfortablen und intuitiven Teach-In vom Bediener handgeführt werden. Für diesen Zweck werden oftmals die Motorströme analysiert oder es kommen Gelenkmomentsensoren zum Einsatz.

Eine weit verbreitete Gruppe von Leichtbaurobotern ist die UR-Serie von Universal Robots. Diese verfügt über Energie- und Kraftbegrenzungsfunktionen, welche den Betrieb des Roboters auch ohne trennenden Schutzeinrichtungen erlaubt. Wie bereits erwähnt, kann das Teach-In auch per Handführen erfolgen. Danach muss man das Programm jedoch noch vervollständigen, wobei die entsprechenden Anweisungen auf dem Touchscreen des Handbediengerätes ausgewählt werden [3]. Nach unserem Verständnis passt diese Vorgehensweise jedoch nicht zum Teach-In per Handführen. Außerdem sollte der Roboter auch im Automatikmodus nicht ausschließlich mittels Kontrollpanel bedient werden können. Bewegt sich der Roboter z.B. im kraftgeregelten Modus [4] und der Bediener greift in den Prozess ein, so wird der Roboter die Kraft weiter erhöhen, wodurch auch der Bediener mehr Kraft aufwendet. Dies ist der Beginn einer Interaktion, welche der Roboter gewinnen wird. In einem solchen Fall könnte ein instinktiver „Stopp“-Ruf des Bedieners den Roboter unter Kontrolle bringen.

Aus den genannten Gründen scheint es notwendig zu sein, alternative Programmier- und Bedienkonzepte zu betrachten. Dazu existiert eine Vielzahl von Arbeiten. Dabei wird z.T. zwischen Experten und Amateuren unterschieden. Im Fall der Experten, würde es hilfreich sein, eine genormte Programmiersprache zu haben. Jedoch ist der Versuch einer Normung, mittels der standardisierten Sprache IRL [5], in der Vergangenheit gescheitert. Für den Fall das auch Nicht-Experten in die Roboterprogrammierung und Bedienung einbezogen werden sollen, spielt die Intuitivität eine wichtige Rolle. Praktische Ansätze können den Roboter durch Berührung, Gesten, Sprache oder 3D-Tracker steuern.

Dieser Beitrag ist im Weiteren wie folgt aufgebaut. Das folgende Kapitel gibt einen Überblick über den aktuellen Stand der Technik in Bezug auf Spracherkennungssoftware. In Abschnitt 3 stellen wir alternative Ansätze zum Teach-In und zum Bedienen eines UR3e-Leichtbauroboters über Sprache vor. Das neue Teach-In-Konzept wird in Abschnitt 4 untersucht. Zu diesem Zweck wird die benötigte Zeit für das Einlernen mit der Dauer herkömmlicher Programmierkonzepte verglichen. Eine Zusammenfassung der Ergebnisse wird in Abschnitt 5 gegeben.

2. Systeme zur Spracherkennung

Eine Art der intuitiven Programmierung und Steuerung ist per Sprache. Sprachsteuerung hat sich schon im Alltag etabliert. Ebenso auf dem Feld der Roboter, wo man bereits diverse Ansätze finden kann. Allerdings sind die

meisten davon für mobile Roboter vorgesehen. Im Gegensatz zu anderen Konzepten, arbeitet die Sprachsteuerung ohne spezielle Hardware. Nur ein Mikrofon wird benötigt, welches in den meisten modernen Kommunikationsgeräten integriert ist, wie im Tablet PC, Laptop oder Handy. Die Sprachsteuerung arbeitet auch unter ungünstigen Lichtverhältnissen und in der Dunkelheit, was für spezielle Befehle, wie den Not-Halt, erforderlich ist. Ein weiterer Vorteil liegt darin, dass der Bediener währenddessen auch etwas Anderes mit seinen Händen machen kann. So dürfte es möglich sein den Roboterarm per Hand zu führen und die Teach-In-Positionen per Sprachbefehl zu speichern.

Zur Realisierung einer Robotersteuerung per Sprache wurden die aktuell relevanten Sprachanalysesysteme recherchiert und in Tabelle 1 aufgeführt.

Sprachanalyse-Software / Eigenschaften	Online	Offline	kostenfreier Download	kostenfreie Nutzung	Entwickler-Schnittstelle
Apps (Microsoft, Android, MAC)	X	X	X	X	
Amazon Web Service Lex	X			X	X
CMU Spinx		X	X	X	X
Google Cloud Speech API	X			X	X
Houndify API	X				X
IBM Whatson	X				
Julius (bis 2014)		X	X	X	X
Nuance		X			
Mozilla CommonVoice		X	X	X	
Mozilla DeepSpeech		X	X	X	X
MS Cortana	X			X*	X*
MS Speech Assistant		X*	X*	X*	X*
Picovoice		X	X		X
Porcupine (HotWordDetection)		X	X		X
Snips Flow	X	X	X		
Snowboy (HotWordDetection)	X	X	X		X
wit.ai	X		X	X	X

Gewünschte Eigenschaft
X* Abhängig von Windows-Lizenz

Tabelle 1: Vergleich aktueller Spracherkennungssoftware

Dabei kann grundsätzlich zwischen Onlinesprachanalyse, deren Verarbeitung in der Cloud stattfindet, und der Offlinesprachanalyse unterschieden werden. Es gibt auch hybride Varianten, welche zum Anlernen den Sprachalgorithmus und die Rechenkapazitäten der Cloud nutzen, ihre eigentliche Arbeit jedoch offline ausführen können. Die bedeutendsten drei Sprachanalysesysteme sind die von Google, Microsoft und IBM. Sie bieten, durch starke Rechenserver, sehr schnelle Analysezeiten mit geringen Fehlerraten. Dieses Cloudcomputing setzt allerdings eine Onlineverbindung der Maschine voraus. Weiterhin ist die Nutzung dieser künstlichen Intelligenz kostenpflichtig und wird je nach Anzahl der transkribierten Wörter oder der eingewählten Zeit abgerechnet.

Im Zuge der vierten industriellen Revolution hält das Internet of Things derzeit Einzug in die Produktion und die Onlinekommunikation der Maschinen nimmt zu. Es gibt jedoch einige Aspekte, wie Betriebskosten, Produktionsgeheimnisse oder eine sensible IT-Sicherheitsinfrastruktur, die für einen cloudbasierten Maschinenbetrieb als kritisch anzusehen sind. Hier bietet eine Offline-Variante Vorteile.

Die Offlinesprachanalyse ist jedoch weniger komfortabel als die cloudbasierte Variante, da ein lokaler Algorithmus mit beschränktem Datensatz erst die Stimme des Bedieners erlernen muss. Die intensive Nutzung der Offline-Spracherkennung verringert diesen Nachteil, da sich der Algorithmus kontinuierlich verbessert, die Produktionsdaten im Unternehmen verbleiben und kostenlos sind. Wenn allerdings die Offlinesprachsteuerung für dieses Projekt geeignet ist, dann sollte auch die Onlinesprachsteuerung funktionieren. Deswegen wird im vorliegenden Szenario eine Offline-Variante verwendet.

Fast alle Offline-Sprachanalysesysteme sind kostenfrei benutzbar, aber nur etwa 60 % der betrachteten Systeme sind auch kostenlos zu erwerben und nur etwa 1/3 bieten eine Entwicklerschnittstelle, um sie in eigenen Anwendungen zu implementieren. Aufgrund der weiten Verbreitung des Betriebssystems Microsoft in Unternehmen und den zugehörigen Entwicklungspaketen wurde der Microsoft Speech Assistent zum Aufbau eines Demonstrators ausgewählt.

3. Funktionsmuster einer Sprachsteuerung für Industrielle Leichtbau-Roboter

Wie bereits erwähnt, verwenden wir den Microsoft Sprachassistenten für die Sprachanalyse. Dieser ist in jedem aktuellen Windows Betriebssystem vorhanden und vielfältig verwendbar. Die Spracheingabe erfolgt über ein einfaches USB-Mikrofon. Der Sprachassistent des Betriebssystems stellt die offline analysierten Wörter den verschiedenen Anwendungen auf dem PC zur weiteren Verarbeitung zur Verfügung. Je nach Sprachqualität muss der Offline-Algorithmus vom Benutzer trainiert werden. Wenn das Betriebssystem neue Sprachdaten empfängt, sendet es diese Daten in unserem Fall an MATLAB. Dies werdet die Daten aus und sendet entsprechende Befehlsfolgen an die Steuerung des Roboters. Die Kommunikation zwischen PC und Robotersteuerung wurde mittels Ethernet hergestellt.

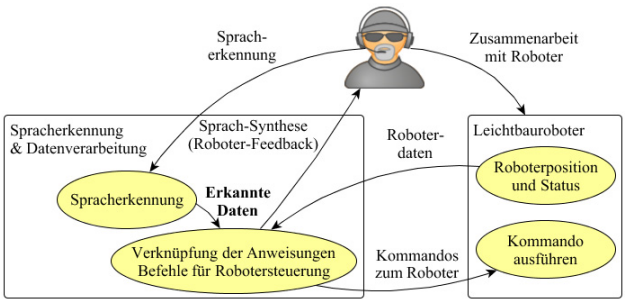

Abbildung 1: Konzept der Sprachsteuerung

Die Kommunikationsstruktur ist in der Abbildung 1 zu sehen. Für die Interaktion mit dem Roboter wird zwischen sprachgesteuerten Teach-In und sprachgesteuerten Bewegungen des Roboters unterschieden.

3.1. Teach-In mittels Sprachbefehl

Während des Teach-In, wird der Roboter per Hand zur Teach-In-Position geführt. Wenn diese erreicht ist, sagt der Programmierer den Namen dieser Position. Dieser wird durch die Sprachanalyse übersetzt und als Variablenname gespeichert. Diese Variable kann nun um Eigenschaften, wie Bewegungsart, Geschwindigkeit oder Beschleunigung erweitert werden. Gleichzeitig werden die Positionsdaten des Roboters ausgelesen und im genannten Variablennamen gespeichert. Dabei entsteht eine Datenstruktur, die für beliebig viele Teach-In-Punkte erweitert werden kann. Die so angelernten Positionen können dem Roboter offline vor Programmstart übermittelt werden oder die Bahn auch online kontinuierlich beeinflussen. Die Abbildung 2 zeigt den Programmablauf.

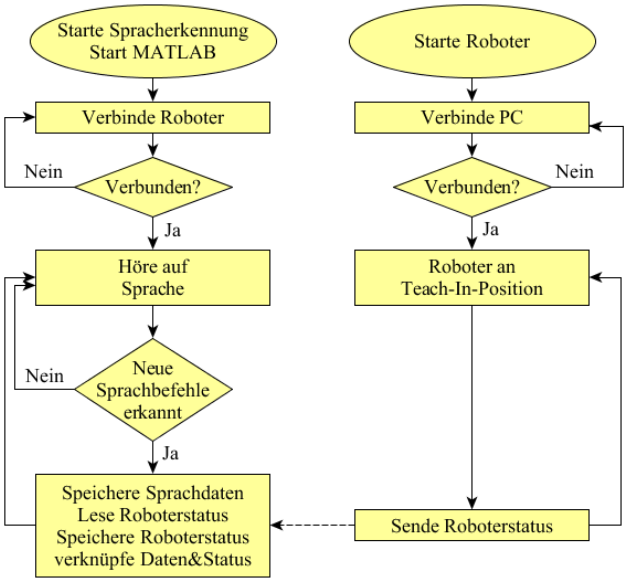

Abbildung 2: Ablaufdiagramm Teach-In mittels Sprache

3.2. Steuerung des Roboters mittels Sprache

Im Weiteren soll der Roboter während des Betriebs durch sprachliche Interaktion in bestimmte Steuerungszustände versetzt werden. Es soll z.B. möglich sein, ihn im laufenden Betrieb durch die Wörter „Stopp“ oder „Halt“ in einen Sicherheitszustand zu versetzen. Außerdem soll die Bewegung des Roboters durch gesprochenen Befehle gesteuert werden. Dazu analysiert das Betriebssystem fortwährend die aufgezeichneten Umgebungsgeräusche. Sobald das Signalwort identifiziert wird, sendet der PC einen entsprechenden Befehl an die Robotersteuerung. Dadurch wird die Zusammenarbeit zwischen Mensch und Roboter wesentlich erleichtert und die Sicherheit insbesondere bei der kollaborativen Mensch-Roboter Interaktion erhöht. Die Abbildung 3 zeigt das Funktionsprinzip der Sprachsteuerung.

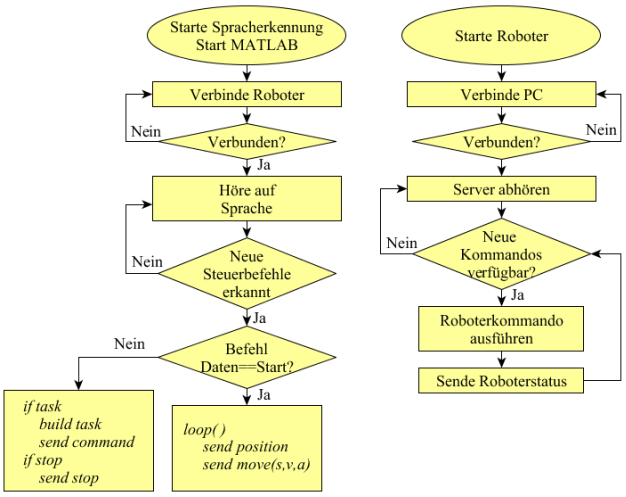

Abbildung 3: Ablaufdiagramm der Sprachsteuerung

4. Vergleich zwischen neuem und konventionellem Teach-In

Universal Robots bewirbt seine Roboter mit einem einfachen und schnellen Programmierverfahren. Tatsächlich erleichtert der vorgefertigte Befehlssatz die Programmierung, sodass diese sehr schnell erfolgen kann. Unsere Absicht war es, die vorgefertigten *URScript*-Befehle mit dem Teach-In per Sprache zu vergleichen. Der aufgebaute Demonstrator ist in Abbildung 4 zu sehen. Dieser besteht aus dem Roboter UR3e mit einem elektrischen Greifer. Dieser hält ein Werkzeug in Form eines Stiftes. Die Bahnpunkte sind auf einem kleinen Tisch markiert.

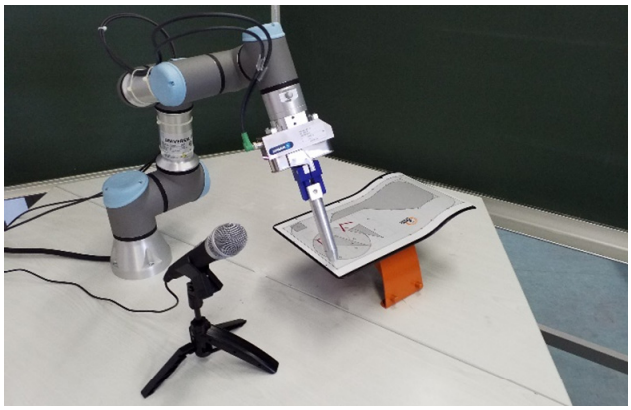

Abbildung 4: Versuchsaufbau zum Vergleichen der Teach-In-Konzepte

Die Bewegung des Roboters zu den Punkten hängt von den Fähigkeiten des Bedieners ab und wird im Vergleich nicht betrachtet. Es wird nur die Programmiermethode berücksichtigt. Um beide Teach-In-Verfahren zu vergleichen, wurden je 10 Punkte aufgenommen, zuerst über herkömmliches Teach-In und dann über Sprach-Teach-In. Des Weiteren wurde das Teach-In mit einem anderen Roboter vom Typ KUKA KR6 Agilus durchgeführt. Beim konventionelle Teach-In wurden bei beiden Robotern jeweils die vorgefertigten Funktionen mit Standardeinstellungen verwendet, um die Teach-In-Zeit zu minimieren. Dies beinhaltet auch das automatisches Beschriften und

Inkrementieren der Variablen. Die Abbildungen 5 und 6 zeigen die Bedienoberflächen der Roboter. Das Teach-In per Sprache erzeugt dieselben Bewegungseinstellungen, jedoch mit individuellen Beschriftungen.

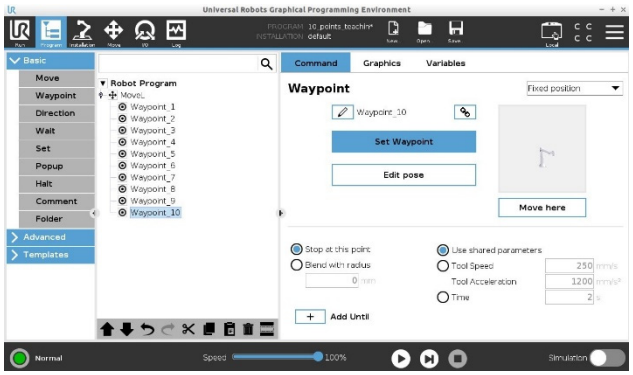

Abbildung 5: Bedienoberfläche des UR3e-Panels

Abbildung 6: Ausschnitt der Oberfläche des KUKA-Bedienfeldes

Position	Teach-Zeit in Sekunden			
	Universal Robots UR3e			KUKA AGILUS 6
	Sprach-Teach-In		Teach-In Bedienpanel	Teach-In Bedienpanel
	total	korr.	total	total
1	9,07	5,57	10,35	15,60
2	8,57	5,07	8,25	13,67
3	6,94	3,44	8,65	12,47
4	7,86	4,36	5,83	26,70
5	6,94	3,44	7,88	12,56
6	5,80	2,30	6,22	9,70
7	6,02	2,52	8,17	10,27
8	6,40	2,90	5,56	10,80
9	5,82	2,32	7,53	9,36
10	8,50	5,00	7,84	11,38

Tabelle 2: Vergleich der verschiedenen Teach-In-Konzepte

Die Tabelle 2 zeigt die angelernten Positionen und die dazugehörigen Teach-In-Zeiten. Wie zu erkennen ist, benötigt das Einlernen beim KUKA die längste Zeit. Möglicherweise hängt dies von der Empfindlichkeit und Latenz des Bedienfeldes ab. Das konventionelle Tech-In des UR3e ist dahingegen fast doppelt so schnell. Die Messung des UR3e-Sprach-Teach-In zeigt nahezu die gleichen Werte wie das herkömmliche UR3e-Teach-In. Die Datenverarbeitung unseres Systems beinhaltet jedoch eine Latenz von 3,5 Sekunden ohne Spracherkennung. Die an die Verarbeitungslatenz angepasste Teach-In-Zeit ist in der Spalte „total“ zu sehen und zeigt, dass das Teach-In per Sprache genau betrachtet doppelt so schnell ist wie das herkömmliche Einlernen. Im Vergleich zum verwendeten konventionellen Industrieroboter wurde sogar nur ein Viertel der Zeit benötigt. Ein weiterer Vorteil würden sich bei der Verwendung von aussagekräftigeren und somit komplexeren Positionsnamen ergeben, indem z.B. Schreibfehler vermieden werden.

5. Fazit

In diesem Beitrag wurde ein Konzept zur intuitiven Bedienung eines Leichtbauroboters per Sprache vorgestellt. Es besteht aus Sprach-Teach-In und sprachbasierter Bewegungssteuerung. Für die Spracherkennung wurde der Microsoft Sprachassistent gewählt. Die Offline-Sprachsteuerung erfordert etwas Zeitaufwand, um den Algorithmus zu trainieren. Während der Tests wurde festgestellt, dass dabei einige Probleme auftreten, z.B. Kollisionen mit anderen Programmen wie Mozilla Firefox. Hinzu kommt eine hohe Auslastung des Betriebssystems. Daher sollten zukünftig alternative Offline-Algorithmen in Betracht gezogen werden.

Die Untersuchung der Sprachsteuerung für industrielle Leichtbauroboter hat einige Vorteile für die Zusammenarbeit zwischen Mensch und Roboter gezeigt. Wie erwartet war das Sprach-Teach-In schneller als die klassische Methode. Darüber hinaus können Sicherheit und Produktivität durch diese Schnittstelle erhöht werden. Es ist möglich, den Roboter mittels eines Ausrufs in einer riskanten Situation anzuhalten. Ebenso kann der Roboter intuitiver gesteuert und bewegt werden. Es ist auch denkbar, den Roboter so in gefährlichen Umgebungen fernzusteuern.

Literaturverzeichnis

[1] S. Yang, X. Mao, B. Ge, and S. Yang, *The roadmap and challenges of robot programming languages*, in Proc. of IEEE Int. Conf. on Systems, Man, and Cybernetics, vol. 19, 2015, pp. 328–333.

[2] B. Finkemeyer, *Towards safe human-robot collaboration*, in Proc. of IEEE Int. Conf. on Methods and Models in Automation and Robotics (MMAR), 2017, pp. 883–888.

[3] Universal Robots A/S, *PolyScope Manual*, 2018.

[4] Universal Robots A/S, *The URScript Programming Language*, 2018.

[5] Deutsches Institut für Normung, *DIN 66312, Industrial Robotic Language (IRL)*, 1996.

Danksagung

Diese Maßnahme wird mitfinanziert mit Steuermitteln auf Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

STAATSMINISTERIUM FÜR WISSENSCHAFT UND KUNST | Freistaat SACHSEN