

Sachbericht zum Verwendungsnachweis Teil 1 Kurzbericht



VIP+ AuRora

Validierung des Innovationspotentials der automatischen
Generierung effizienter Roboterprogramme

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 03VP06730 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin/beim Autor.

Zuwendungsempfänger:

Prof. Dr. rer. nat. Frank Ortmeier
Lehrstuhl für Software Engineering
Institut für Intelligente Kooperierende Systeme
Fakultät für Informatik

Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, 39106 Magdeburg
frank.ortmeier@ovgu.de
Telefon: +49 391 67-58915
Telefax: +49 391 67-12810

Förderkennzeichen:

03VP06730

Projektlaufzeit

01.01.2020 bis 30.06.2023

Berichtszeitraum

01.01.2020 bis 30.06.2023

Datum:

Dezember 2023

Heutzutage werden in der professionellen Automatisierung mit Industrierobotern die Anlagen meist in Simulationsumgebungen geplant und programmiert. Die Programmierung in diesen Umgebungen selbst geschieht jedoch meist wie bei der Online-Programmierung an der Anlage: die Bewegungsbefehle werden manuell von einem erfahrenen Roboterprogrammierer erstellt und simulativ und visuell überprüft. Auch wenn diese Umgebungen teilweise spezielles Prozesswissen (bspw. Lichtbogenschweißen, Sprühen) mit abbilden können, bleibt die Bewegungsprogrammierung selbst immer noch Handarbeit. Die Qualität und Erstellungszeit hängen somit viel von der Qualifikation des Programmierers ab.

Zentrales Ziel von VIP+ AuRora ist es das zu entkoppeln, indem die Teilaufgaben der Roboterprogrammierung durch neue Methoden weitestgehend automatisiert werden und somit die Verbreitung flexibler Roboterautomation ermöglichen.

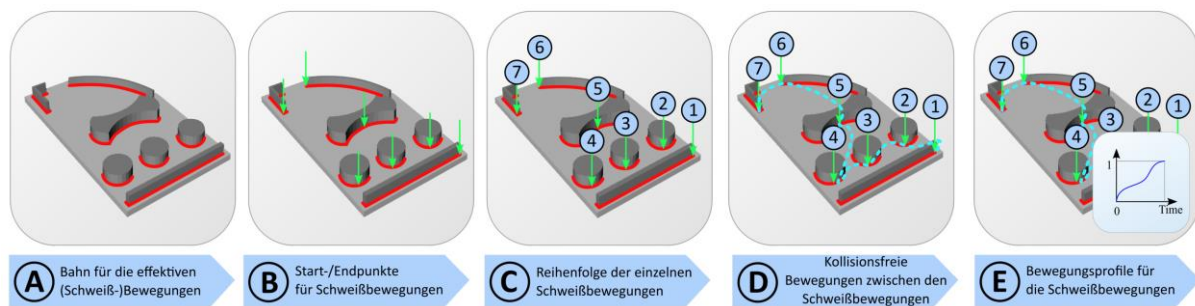


Abbildung 1 Teilaufgaben eines Programmierprozesses

Abbildung 1 zeigt ein typisches Schweißszenario. In diesem Beispiel sollen sieben Metallobjekte auf einer Metallplatte fixiert werden. Um für dieses Problem zu einem Roboterprogramm zu kommen, müssen in der Regel folgende dargestellten Teilaufgaben durchgeführt werden:

1. Definition der Bahn für die effektiven Bewegungen (rote Linien in A)
2. Definition der Start-/Endpunkte für Effektivbewegungen (grüne Pfeile in B)
3. Definition der Reihenfolge der einzelnen Effektivbewegungen (Reihenfolge in C)
4. Definition kollisionsfreier Bewegungen zwischen den Effektivbewegungen (gestrichelte Linien in D)
5. Definition von Bewegungsprofilen für die Effektivbewegungen (E)

Die im Projekt entwickelten Methoden zeigen erfolgreich wie die Teilaufgaben 2 bis 5 automatisiert werden können. Um die Güte der entwickelten Methoden zu testen, wurden auch folgende Validierungsziele für das Projekt definiert:

VZ1.1: 90%ige Zeitreduktion bei Programmen für einfache Szenarien

VZ1.2: 50%ige Zeitreduktion bei Programmen für komplexe Szenarien

VZ1.3: Ergebnis nicht schlechter als das Programm eines Experten

VZ2.1: 33%ige Verringerung der allgemeinen Arbeitstaktzeit (Zeit, die Roboter für die Abarbeitung der einzelnen Effektiv- und Unterstützungsaufgaben benötigt)

VZ2.2: 25%ige Verringerung des Gesamtenergieverbrauchs

VZ2.3: 33%ige Verringerung des Verschleißes

Durch den Wechsel von Prof. Dr.-Ing. Rolf Findeisen an die Technische Universität Darmstadt wurde das Projekt durch den Verbundpartner Technische Universität Darmstadt erweitert und die Validierungsziele wurden somit so angepasst, dass bei VZ2.2 und VZ2.3 die Verbesserungen bezüglich Energieverbrauch und Verschleiß nur mit Hinblick auf eine verbesserte Planung (wie z.B. über die Optimierung der Sequenzen) evaluiert werden soll.

Die Validierungsziele VZ1.1 und VZ1.3 konnten im Rahmen des Projektes erfolgreich validiert werden. VZ1.2 konnte aus technischen Gründen nicht evaluiert werden, da dafür eine Unterstützung von Roboter-systemen mit mehr als 6 Achsen notwendig ist (zum Beispiel ein Roboter auf einer Linear Schinne), was im Rahmen der Projektlaufzeit nicht vorgesehen wurde und dementsprechend nicht vollständig umgesetzt werden konnte.

VZ2.1 konnte für die Testszenarien ebenfalls nicht positiv validiert werden. Im Schweißtestszenario konnte die Taktzeit der Gesamtaufgabe von 91,488s auf 90,408s reduziert werden. Die Dauer der Effektivbewegungen (Schweißen) betrag dabei 83,868s. Die Zeit für die Schweißbewegungen lassen sich Prozessbedingt nicht reduzieren. Die Zeiten für die Unterstützungsbewegungen ließen sich von 7,62s auf 6,54s reduzieren, was einer Reduzierung von 14,2% entspricht. Diese Ergebnisse sind aber Prozess- und Setupabhängig und werden bei z.B. Punktschweißen von 50 Knoten zu viel besseren Verbesserungen führen im Vergleich zu auftragen von 5 geraden Kleberauppen.

VZ2.2 konnte für den evaluierten Roboter nur anhand der Sequenzoptimierung ebenfalls nicht validiert werden. Der Energieverbrauch des untersuchten Kuka KR3 Agilus konnte um 1% reduziert werden. Besonders auffällig dabei war jedoch der erhebliche Energieverbrauch der Robotersteuerung, welcher den Verbrauch des Roboters übersteigt. Für Größere Roboteranlagen kann dieser Wert jedoch erheblich größer ausfallen. Aus Sicht des Projekt-Teams ist jedoch der sinnvollere Weg zur Energieeinsparung von Roboteranlagen die Reduzierung des Verbrauches der Robotersteuerung. Kuka hat mit der KR C5 und C5 micro während der Projektlaufzeit bereits neue Steuerungen auf dem Markt gebracht, welche genau dieses Problem adressieren.

Bei VZ2.3, ähnlich zur VZ2.1, hatte die größte Anteil von Effektivbewegungen (die nicht geändert werden konnten) in Relation zur Unterstützungsbewegungen (die hinsichtlich des Verschleißes optimiert werden konnten) ebenfalls nur zur kleineren Verbesserung geführt.

Auch wenn nicht alle Validierungsziele vollständig erreicht werden konnten, wurden die Forschungsergebnisse in bestehende und in der Industrie bereits eingesetzte Simulations- und Roboterprogrammiersoftware integriert. Roboterprogrammierer werden somit weiterhin in ihrer gewohnten Umgebung arbeiten können, erhalten aber eine signifikante automatisierte Unterstützung, die ihnen bereits für eine Fertigungsaufgabe optimierte Roboterprogramme erstellt.

Sachbericht

zum Verwendungsnachweis

Teil 2



VIP+ AuRora

Validierung des Innovationspotentials der automatischen Generierung effizienter Roboterprogramme

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 03VP06730 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin/beim Autor.

Zuwendungsempfänger:

Prof. Dr. rer. nat. Frank Ortmeier
Lehrstuhl für Software Engineering
Institut für Intelligente Kooperierende Systeme
Fakultät für Informatik

Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, 39106 Magdeburg
frank.ortmeier@ovgu.de
Telefon: +49 391 67-58915
Telefax: +49 391 67-12810

Förderkennzeichen:

03VP06730

Projektlaufzeit

01.01.2020 bis 30.06.2023

Berichtszeitraum

01.01.2020 bis 30.06.2023

Datum:

Dezember 2023

Zusammenfassung:

Zentrales Ziel von VIP+ AuRora war es zu zeigen, dass mittels bereits entwickelter Algorithmen automatisiert hoch-effiziente Roboterprogramme erzeugt und bereits bestehende optimiert werden können. Programme, die bisher manuell von geschultem Personal erstellt werden mussten, können somit automatisch für beliebige Robotersysteme generiert werden. Eine aufwändige experimentelle Validierung und Anpassung bei der manuellen Erstellung von Ablaufplänen und Roboterbewegungen kann vermieden werden. Durch diese große Zeit- und Aufwandsersparnis bei der Programmerzeugung sinken Entwicklungs- und Einstellungskosten für die Anpassung an neue Produktserien. Dies hätte enorme Auswirkungen auf den Kosten/Nutzen-Wert für den Einsatz und die Anschaffung von Industrierobotern, vor allem auch in KMUs, bei der Umsetzung von Produkten in Klein- und Kleinstserien sowie als Einzelstücke.

Durch innovative optimierungsbasierte Planungsverfahren sowie die Ausnutzung weiterer Freiheitsgrade soll aber auch die Qualität bereits vorhandener Roboterprogramme gesteigert werden. Dies bezieht sich auf die Minimierung der Laufzeit, des Energiebedarfs und des Verschleißes der Manipulatoren bei mindestens gleichbleibender Bahngenauigkeit, was wiederum zu einer Kostenersparnis für die Anwender führt. Manuell ließe sich solch eine Optimierung insbesondere bei Größen, die nur schwer vorstellbar sind (beispielsweise der Ruck in Gelenkwinkelkoordinaten), erst nach zahlreichen zeit- und kostenaufwändigen experimentellen Iterationen erreichen.

Projekt Ablauf:

Um diese Ziele zu erreichen hatte die ursprüngliche Planung eine thematische Einteilung in fünf Bereiche mit folgenden Inhalten:

AP I: Anforderungsanalyse

Dieses Arbeitspaket sollte die Analyse aller relevanten Rahmenbedingungen, z.B. technologische Fragen wie die Kompatibilität der Roboterschnittstellen oder Drittanbieter-CAD-Software beinhalten. Auch die Analyse der potentiellen Anwendungsdomänen und eine Kategorisierung der dort auftretenden Aufgabenspezifikation sowie der zur erfolgreichen späteren Verwertung notwendigen Akzeptanzkriterien gehört dazu. Weiterhin sind Anforderungen an Nutzungsschnittstellen und Prozessintegration zu erheben.

AP II: Validieren der technischen Machbarkeit

Kerninhalte dieses AP sind das Erstellen der zur Optimierung notwendigen physikalischen Systemmodelle für die Fallbeispiele sowie die Anwendung der entwickelten Methoden auf diese. Im Ergebnis wird in einer Computersimulation gezeigt, dass die Kernziele in einem simulierten industriellen Kontext erreicht werden können. Dabei werden die simulierten Werte mit dem zuvor von einem professionellen Systemintegrator erstellten Referenzprogramm verglichen. Die Modellerstellung wird in dieser Phase noch weitgehend manuell erfolgen und liefert dadurch wichtiges Feedback zur Entwicklung der praxistauglichen Versuchssoftware.

AP III: Konzeption einer Versuchs- und Demonstrationssoftware

Aufbauend auf AP II soll eine Versuchs- und Demonstrationssoftware erstellt werden. Dies ist Grundlage für die praktische Erprobung und somit auch für die Validierung der Transitionskonzepte.

AP IV: Praktische Erprobung und Evaluation

Aufbauend auf den Ergebnissen aus AP I bis III sollen für mindestens zwei ausgewählte Fallbeispiele praktische Erprobungen durchgeführt werden, um das Validierungsziel VZ1 (90 %ige Zeitreduktion bei normalen Szenarien) zu untersuchen. Dabei wird ausgehend von der Spezifikation der Aufgabe in einer Drittanbieter-CAD-Software der komplette Prozess bis zum Vorliegen des Roboterprogramms evaluiert. Ein weiteres Szenario wird mit Hinblick auf die Untersuchung des Validierungsziels VZ1.2 (50 %ige Zeitreduktion bei komplexere Planungsaufgaben) betrachtet.

Zusätzlich soll, ein durch den Beirat definiertes Szenario vergleichend mit externen Lösungen eines Drittanbieters bewertet werden. Die hierbei generierten Lösungen können dann als Vergleichsgrundlage zu den in diesem Projekt erreichten Ergebnissen dienen und sowohl Nutzen als auch Verbesserungspotenziale aufzeigen.

AP V: Transitionskonzepte

Um die fortlaufenden Arbeiten und Zwischenergebnisse auszuwerten und neue Impulse einzubringen, soll in regelmäßigen Abständen Rücksprache mit dem Innovationsmentor und mit weiteren Domänenexperten gehalten. Die Verwertung des Konzepts wird durch Diskussionen mit Unternehmen, z.B. auf Messen, und durch Business Cases evaluiert.

Die grobe zeitliche Planung sah bereits eine iterativ-inkrementelle Entwicklung vor. Das bedeutet, dass vor allem in den Arbeitspaketen II, III und IV in jeder einzelnen Iteration (vertikale) Prototypen mit einem beschränkten Funktionsumfang geschaffen werden. Diese Planung ist in Abbildung 1 dargestellt.

Es wurden aber schon am Anfang des Projektes Anpassungen vorgenommen. So wurde es gemeinsam mit dem Industriebeirat und im ersten Expertenworkshop festgestellt, dass aufgrund von mangelhaftem Schnittstellenzugriff, alle Programme auf der Herstellerseitig gelieferten Robotersteuerung lauffähig sein müssen. Dies ist notwendig um die Akzeptanz in der potenziellen Verwertung zu gewährleisten, da durch eine hardwarenahe Schnittstelle in der Regel alle Zertifizierungen des Roboters (CE, MACHSchiene-Richtlinie, ISO TS 15066, etc.) ausgehebelt werden. Daraus resultiert, dass die meisten aller Arbeiten zur Taktzeitoptimierung innerhalb einer Offlineprogrammierungsumgebung (mit Simulationsfunktion) durchgeführt werden konnten. Nach der Evaluation der Benutzungskonzepte, wurde festgestellt, dass es nicht möglich ist ausreichend Akzeptanz für die vorgeschlagene Innovation zu erzeugen, ohne dass Anwender in ihren bereits genutzten Simulationsprogrammen eine Überprüfung vornehmen. Zusätzlich neue Simulationssoftware wird überwiegend abgelehnt. Vorhandenes soll genutzt werden. Um die für die Innovation benötigte Akzeptanz zu erzeugen, wurde entschieden die automatisch generierten, optimalen Roboterpfade innerhalb der bei den Anwendern vorhandenen Simulations-Softwares verfügbar zu machen bzw. einzubinden. Dies ist durch die Natur von proprietärer Software jedoch mit erheblichem Mehraufwand verbunden, für die Akzeptanz bei den Anwendern jedoch zwingend. Die Validierung innerhalb der proprietären Software ist auch deshalb notwendig, da nicht sichergestellt werden konnte, ob eine Übertragung der Konzepte Open-Source Lösungen auf entsprechende industrielle Softwarelösungen möglich ist. Aus diesen Gründen wurde gemeinsam mit dem Projektträger entschieden keinen zweiten Roboter zu beschaffen, sondern die Entwicklungen in mindestens zwei Programmierungsumgebungen (OLP) zu integrieren.

	Jahr 1	Jahr 2				Jahr 3				Jahr 4		
	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3
Anforderungsanalyse												
AP I.1 (3 PM)	■											
AP I.2 (5 PM)		■	■									
AP I.3 (4 PM)			■	■								
AP I.4 (3 PM)				■	■							
AP I.5 (3 PM)					■							
AP I.6 (3 PM)						■						
AP I.7 (3 PM)							■					
AP I.8 (3 PM)								■				
Techn. Machbarkeit												
AP II.1 (6 PM)	■	■	■		■	■	■					
AP II.2 (5 PM)			■	■	■		■	■				
AP II.3 (4 PM)				■	■			■	■			
AP II.4 (4 PM)					■				■			
AP II.5 (4 PM)						■				■		
Demonstrationssoftware												
AP III.1 (9 PM)		■	■	■	■							
AP III.2 (9 PM)			■	■	■	■						
AP III.3 (6 PM)				■	■							
AP III.4 (6 PM)					■	■						
AP III.5 (5 PM)						■						
AP III.6 (12 PM)							■	■		■	■	
AP III.7 (12 PM)									■	■	■	
AP III.8 (8 PM)								■	■	■	■	
Praktische Erprobung												
AP IV.1 (6 PM)				■	■		■	■				
AP IV.2 (12 PM)					■		■	■	■			
AP IV.3 (12 PM)							■	■	■	■		
AP IV.4 (7 PM)										■	■	
AP IV.5 (5 PM)							■	■			■	■
AP IV.6 (6 PM)								■	■		■	■
AP IV.7 (3 PM)										■	■	■
Transitionskonzepte												
AP V.1 (4 PM)				■		■		■		■		■
AP V.2 (6 PM)					■		■		■		■	
AP V.3 (6 PM)							■		■		■	
AP V.4 (3 PM)		■				■			■		■	
AP V.5 (4 PM)			■				■			■		■
AP V.6 (4 PM)				■				■			■	

Abbildung 1: Ursprüngliche Zeitliche Planung

Die Beschaffung einer weiteren OLP hat sich jedoch als schwierig herausgestellt. Denn für die Umsetzung des Vorhabens ist ein sehr tiefgehender Eingriff in proprietäre Software nötig, welcher bisher nur VisualComponents (und OEM's) ermöglicht wird. Durch unser Industrienetzwerk ist es jedoch gelungen die Artiminds Robotics GmbH davon zu überzeugen eine entsprechende Schnittstelle in ihrer OLP (Artiminds RPS) für unsere Algorithmen zu schaffen. Durch die Integrationstiefe war hier ein hoher Planungsaufwand gegeben und viele Entscheidung mussten seitens Artiminds von der Führungsebene zugestimmt werden. Durch

diesen zeitaufwändigen Planungsprozess, konnte mit der Umsetzung (seitens Artiminds) erst im August 2022 begonnen werden.

Um das Projektziel der automatischen Generierung von Programmen und Pfaden zu erreichen, mussten weiterhin einige unerwartete konzeptionelle und technische Herausforderungen gelöst werden. Das Roboterprogramme und Pfadbeschreibungen in der Industrierobotik nicht als kontinuierliche Pfadbeschreibung vorliegen, sondern in einer Herstellerspezifischen Beschreibungssprache beschrieben sind (z.B: KRL, RAPID), welche erst von der Robotersteuerung in eine Trajektorie umgewandelt wird. In allen uns bekannten OLPs ist die genaue Trajektorie des Roboters erst nach der Simulation des Kompletten Programmes verfügbar.

Die Simulation wird dann in der Regel durch ein vom Roboterhersteller geliefertes RCS-Modul durchgeführt. Die Verwendung dieser Module wurde durch das Aurora Team evaluiert. Die RCS-Module wurden jedoch für eine Simulation in Echtzeit entwickelt und haben sich daher als zu langsam herausgestellt. Bereits die Generierung eines einfachen Roboterprogrammes würde mit dem Hersteller RCS-Modul über 4 Tage dauern. Daher war es erforderlich an dieser Stelle ein eigenes herstellerunabhängiges Modul zu entwickeln. Da die Start- und Endpunkte einer Robotertrajektorie für die Sequenzoptimierung ohne das Modul nicht berechnet werden können, musste auf die Fertigstellung des Modules gewartet werden.

Auch durch den Corona bedingten Ausfall der meisten Messen und Netzwerk Veranstaltungen war es schwierig das Industrienetzwerk weiter auszubauen. Die Kaltakquise von Industriekontakten durch E-Mail und Telefon war leider nicht nachhaltig und hat nicht zu Erfolgen geführt. Besonders bei der Suche nach dem passendem Systemintegratoren waren wir nicht erfolgreich, was die Durchführung der geplanten Benchmark-Challenge beeinflusste. Weiterhin wurde das Projekt durch den Wechsel von Prof. Dr.-Ing. Rolf Findeisen an die Technische Universität Darmstadt durch den Verbundpartner Technische Universität Darmstadt (TUD) erweitert und die Arbeitspakete zwischen beiden Arbeitsgruppen verteilt. Zusätzlich dazu gab es auch inhaltliche Anpassungen. Generell gilt es bei der Verteilung, dass die Arbeitsgruppe Findeisen an der TUD den Einfluss einer iterativen Regelung auf die Optimierung untersucht, wodurch die Arbeitsgruppe Ortmeier an der OvGU die Verbesserungen mit Hinblick auf eine verbesserte Pfadplanung (z.B. über die Optimierung der Sequenzen). Das Team arbeitet somit in Kooperation mit der Arbeitsgruppe aus TUD zusammen um dort entwickelte modelprädiktive Trajektorienplaner zu integrieren, um für vorgegebenen Pfad Energieoptimale Trajektorien berechnen, welche in der Folge dann bei der Planung einbezogen werden können.

Als Konsequenz wurde eine kostenneutrale Verlängerung beantragt und auch genehmigt. In der Retrospektive stellt sich der Zeitplan jetzt wie im Abbildung 2 dargestellt dar.

	2020				2021				2022				2023	
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2
Anforderungsanalyse AP I.1 (3 PM) AP I.2 (3 PM) AP I.3 (4 PM) AP I.4 (3 PM) AP I.6 (2 PM) AP I.7 (2 PM)	■													
Techn. Machbarkeit AP II.2 (3 PM) AP II.3 (2 PM)			■	■		■	■							
Demonstrationssoftware AP III.1 (8 PM) AP III.2 (8 PM) AP III.3 (6 PM) AP III.4 (6 PM) AP III.5 (5 PM) AP III.7 (7 PM) AP III.8 (4 PM)			■	■	■	■	■	■	■	■	■	■	■	■
Praktische Erprobung AP IV.1 (6 PM) AP IV.2 (6 PM) AP IV.3 (6 PM) AP IV.4 (4 PM) AP IV.5 (5 PM) AP IV.6 (6 PM) AP IV.7 (2 PM)					■	■	■	■	■	■	■	■	■	■
Transitionskonzepte AP V.1 (4 PM) AP V.2 (3 PM) AP V.3 (3 PM) AP V.4 (2 PM) AP V.5 (2 PM) AP V.6 (3 PM)		■			■	■	■	■	■	■	■	■	■	■

Abbildung 2: Ablauf des Projektes inkl. kostenneutraler Verlängerung

Projekt Ergebnisse:

Es wurden Analysen durchgeführt, zur Auswahl des richtigen Robotersystems, betrachtet wurden dabei vor allem die Verbreitung des Robotertyps in der Industrie, das Einsatzgebiet und vorhandene Software Schnittstellen. Ausgewählt wurde dabei ein KR 3 AGILUS, weil Kuka im deutschen Raum einen hohen Marktanteil hat und dieser dieselbe Steuerungssoftware nutzt wie größere Robotermodelle. Die Auswahl ist bewusst nicht auf einen kooperativen Leichtbauroboter gefallen, da diese zum einen nicht über eine solch hohe Verbreitung verfügen, als auch normalerweise nicht für die im Projekt relevanten Prozesse eingesetzt werden und eine andere Steuerungssoftware verwenden. Die Wahl fiel auf den KR3, da dieser aufgrund seiner Größe und der Verfügbarkeit als fertige Roboterzelle für den Einsatz im den Vorhanden Räumlichkeiten am einfachsten einsetzbar ist. Da größere Robotermodel dieselbe Steuerung nutzen sind die Ergebnisse einfach auf größere Modelle mit ähnlichem Aufbau zu übertragen (z.B. „Kuka KR 120“ oder „Kuka KR 1000 Titan“).

Bei der Erstellung der Softwarearchitektur wurden folgende grundlegende strategische Entscheidungen getroffen:

- Die Software wird in zwei Module aufgeteilt:
 - Algorithmisches Backend in C++, enthält die eigentlichen Optimierungsalgorithmen
 - Benutzerschnittstelle zur Eingabe des Roboterprogrammes
- Die Benutzerschnittstelle wird nicht als eigenständige Software entwickelt, sondern als Plugin in bestehende Offline Programmierumgebungen integriert

Die Aufteilung der Software hat initial zu einem erheblichen Mehraufwand geführt, erleichterte später jedoch erheblich das Vorhaben die späteren Berechnungen in die Cloud auszulagern und erhöht massiv die Wiederverwendbarkeit der Softwarekomponenten, da sie nur über eine genau zu spezifizierende Schnittstelle anzusprechen sind.

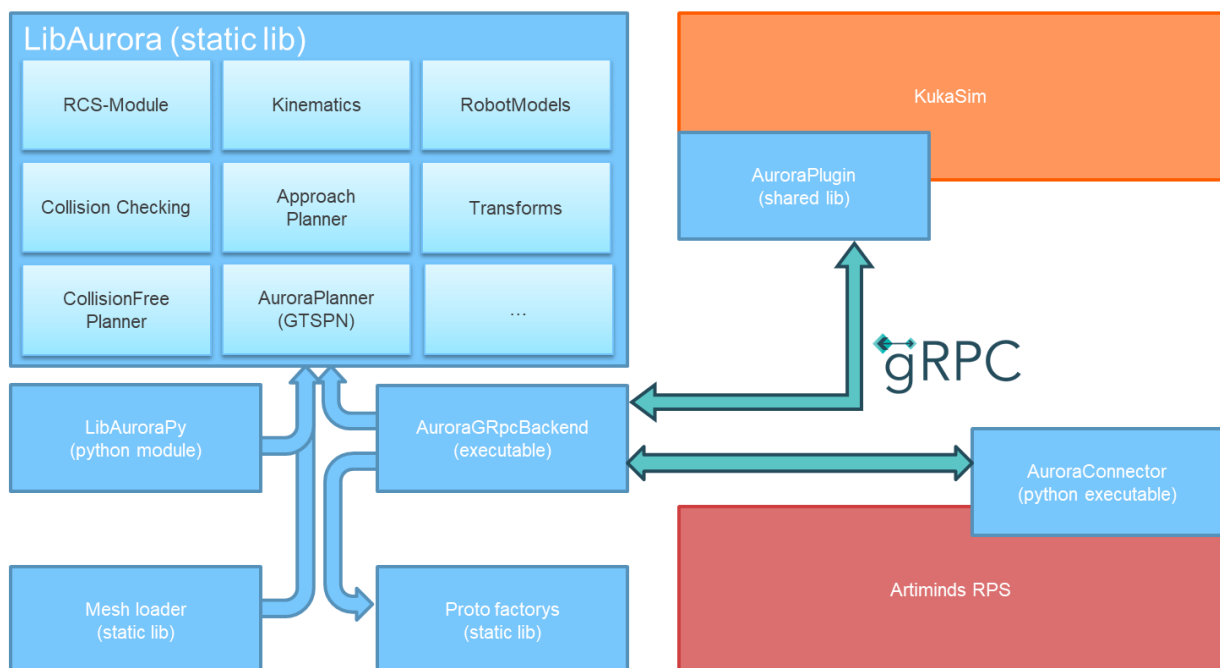


Abbildung 3: Software Architektur

Sehr schnell zu Beginn des Projektes wurde im Rahmen der Expertentreffen festgestellt, dass üblich Roboterprogramme nicht über genug Information verfügen um die geplanten Algorithmen zur Sequenzoptimierung anzuwenden. Bisherige Roboterprogramme sind recht einfach aufgebaut und bestehen meist nur aus einer Abfolge abzuarbeitender Befehle. Das heißt, dass Informationen über die Geometrie des zu bearbeitenden Objekts im finalen Roboterprogramm nicht mehr enthalten sind. Das diese Informationen für die Optimierung des Programmes nötig sind, wurde ein Konzept erarbeitet, welches es ermöglicht, das Roboterprogramm um semantische Zusatzinformationen zu erweitern. Das Konzept führt ein neues Primitiv in Roboterprogramme ein, welche Kontur getauft wurde. Eine Kontur gruppiert letztlich eine Menge von Roboterbefehlen, welche bei der Ausführung die Trajektorie des Roboterwerkzeuges beschreiben. Eine Kontur kann außerdem durch zusätzliches Prozesswissen Parametrisiert werden. Wie z.B. Geschwindigkeit auf der Kontur oder die Bewegungsrichtung auf der Kontur (im oder gegen den Uhrzeigersinn). Das Konzept der

Konturen wurde im weiteren Projektverlauf um die Komponenten Entry, Body, Exit erweitert.

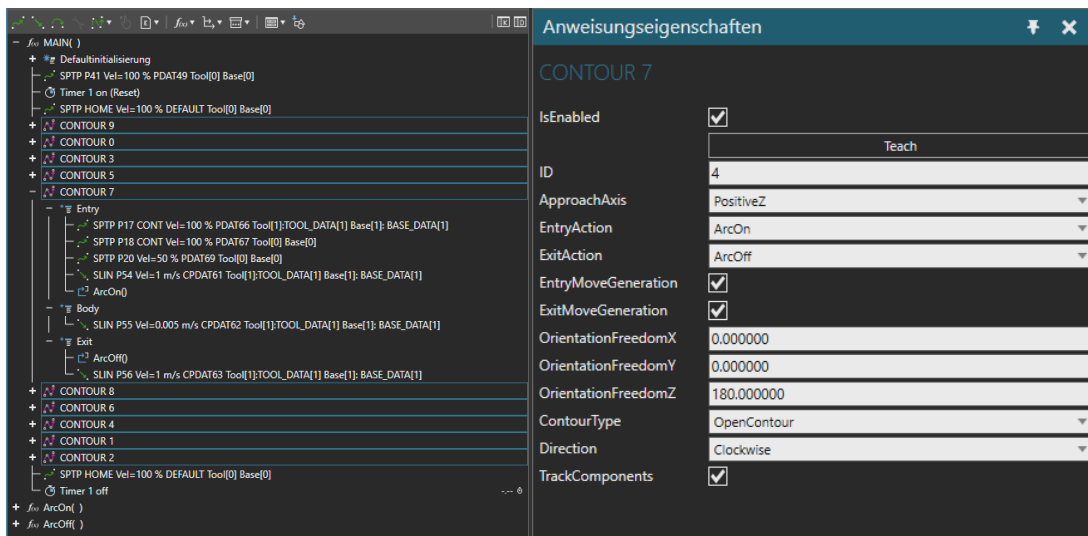
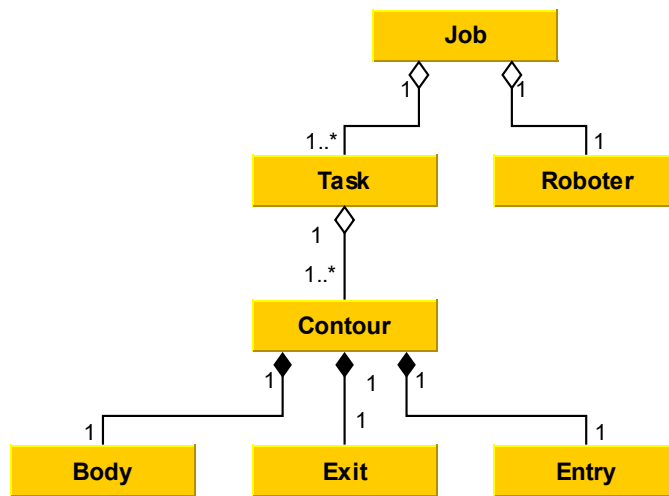


Abbildung 4: Eingefügter Konturkonzept und deren Implementierung

Nachdem der erste Prototyp der Algorithmen implementiert war, musste das Projektteam leider feststellen, dass durch einige starke Vereinfachungen in der Modellierung des Problems in der vorangegangenen Promotionsarbeit, die Algorithmen nur mit einer sehr eingeschränkten Auswahl von Prozessen verwendbar sind. So wurde bei den einzelnen Teilaufgaben der Sequenzoptimierung davon ausgegangen, dass die Start- und Endkonfiguration identisch sind, dass ist jedoch nur der Fall für Prozesse, wo die Rotation des Werkzeuges um die Stoß-Achse nicht relevant ist. Beispielprozesse dafür wären Punktschweißen, Fräsen oder Bohren. Für einen Großteil aller anderen Prozesse muss die Endkonfiguration auf dem Pfad erst berechnet werden. Alle von uns gefundenen Veröffentlichungen zu diesem Thema haben dasselbe Problem und beschränken sich deshalb meist auf die oben genannten Prozesse. Um für eine Startkonfiguration eine Endkonfiguration zu berechnen muss das Roboterprogramm, welche die Teilaufgabe beschreibt, simuliert werden. Für die Simulation von Roboterprogrammen gibt es seit den 90igern das RCS (Robot Controller Simulation) Interface. Diese wird von den meisten großen Roboterherstellern unterstützt, erfordert aber für jeden Hersteller ein proprietäres RCS-Modul. Leider ist es uns

nicht gelungen dieses zu beschaffen. Die RCS Module werden zwar von den Roboterherstellern entwickelt, da diese eine modifizierte Variante der Software auf der Robotersteuerung sind. Allerdings werden die Module ausschließlich durch Siemens als Teil von Process Simulate lizenziert. Kuka liefert sein RCS Modul außerdem als Bestandteil der KukaSim OLP Umgebung mit aus, welches ohne KukaSim allerdings nicht lauffähig ist. Nach dem das Modul analysiert wurde, mussten wir allerdings feststellen, dass das Modul für unseren Zweck viel zu langsam ist, da die Simulation innerhalb des Optimierungsalgorithmus mehrere zehntausende Male berechnet werden muss. Das Projektteam hat ebenfalls keine alternative Möglichkeit in Form von frei verfügbarer Software gefunden, welche die Simulation von Roboterprogrammen erlaubt. Daher wurde eine eigene Lösung, im folgenden Aurora-RCS genannt, geschaffen, welche die nötigen Berechnungen ausführt. Dazu wurde im ersten Schritt das Verhalten einer Robotersteuerung nach Lehrbuch implementiert und in einem weiteren Schritt die Kuka spezifischen Eigenheiten der Herstellersoftware übernommen. Dazu zählen das Handling von Singularitäten und Geschwindigkeitsprofilen, sowie die Berechnung der Rotationen. Das Softwaredesign wurde bewusst so umgesetzt, dass eine einfache Anpassung an andere Hersteller möglich ist. Das Projektteam hat auch mit auf KI basierenden Methoden experimentiert, die Idee wurde aber aufgrund von fehlenden Kompetenzen im Projektteam und mangelnden Kapazitäten der Kollegen nicht weiterverfolgt. Das Aurora-RCS-Modul unterstützte somit Linear-, Kreis- und Punkt zu Punkt-Bewegungen. Abbildung 5 zeigt zum Beispiel den Vergleich zwischen generierten Kreisförmigen Bewegung und mit dem Roboter aufgenommenen Daten für verschiedene Arten der Rotation.

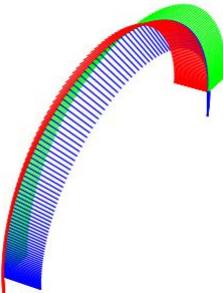
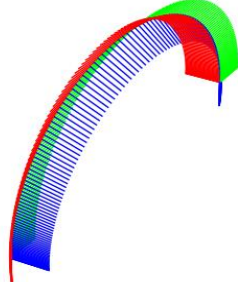
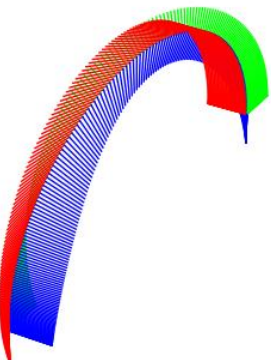
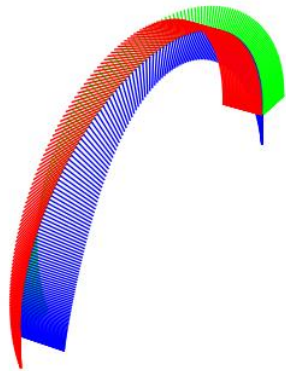
	Kuka-RCS	Aurora-RCS
Kreis-Bewegung, Rotation Basis bezogen		
Kreis-Bewegung, Rotation Pfad bezogen		

Abbildung 5: Rotation Interpolation Vergleich

Parallel zu dieser Entwicklung wurde auch kollisionsfreie Bahnplanung implementiert. Die größte Herausforderung hier war es innerhalb der OLP Umgebung Zugriff auf die 3D-Objekte zu bekommen und die komplette 3D-Szene der OLP Umgebung innerhalb unserer Software zu rekonstruieren (Abbildung 6), da dies zwar technisch möglich ist, aber Seitens des Herstellers nicht unterstützt wird.

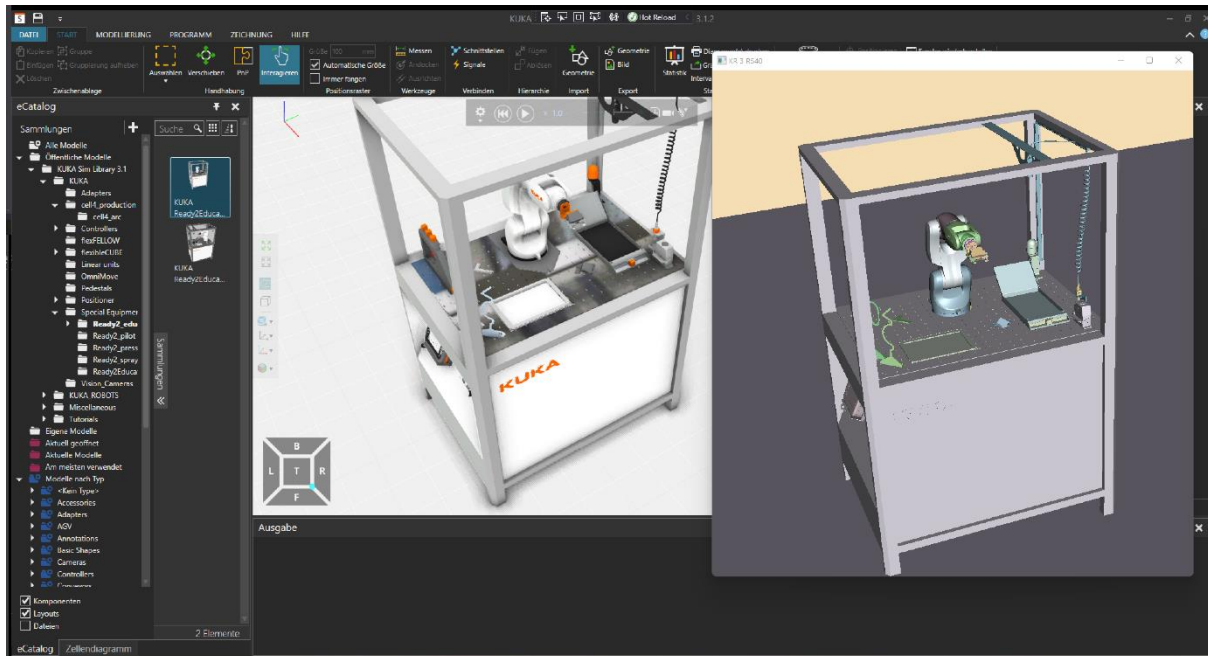


Abbildung 6: Rekonstruierte 3D-Szene

Mit Hilfe dieser Implementierungen war es Mitte 2022 erstmals möglich, einfache Szenen zuverlässig hinsichtlich der Zykluszeit zu optimieren und gleichzeitig Kollisionsfreiheit zu gewährleisten. Basierend auf diesem Stand wurde auch ein Demonstrator für die Automatica 2022 erstellt.

Der technische Stand der Automatica hat jedoch Probleme mit komplexeren Szenarien, da die Berechnung für eine einfache Anwendungen noch mehrere Minuten dauerte und die gefundenen Pfade teilweise von schlechter Qualität waren, was für die Taktzeitoptimierung kritisch ist. Das Problem konnte aber mithilfe der Parallelisierung der Pfadplanung und Wiederverwendung von Kollisionsinformationen bei Veränderung der 3D-Szene gelöst werden.

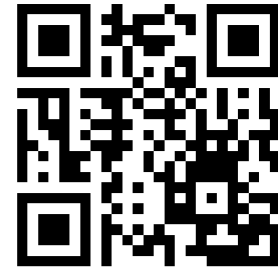
Da die Softwarearchitektur von Aurora von Anfang an für den Einsatz in der Cloud konzipiert wurde, konnte der Pfadplaner mit wenigen Anpassungen auch auf einem Server des Universitätsrechenzentrums ausgeführt werden. Die Evaluierung zeigte, dass die entwickelte Lösung problemlos auf 128 Cores skaliert und damit um den Faktor 128 schneller ist als die dem Projektteam bekannten Open Source Lösungen.

Aufgrund der hohen Performance des Planers wurde die Funktionalität zusätzlich als Stand-Alone-Funktion implementiert und der OLP-Schnittstelle zur Verfügung gestellt. Das Projektteam erhielt sehr positives Feedback aus dem Industrienetzwerk und es wurden Herausforderungen gestellt, in denen das Projektteam zeigen konnte, dass die Lösung nicht nur besser als aktuelle Open Source Lösungen, sondern auch besser als kommerzielle Lösungen ist:

AuRora-ABB¹

ABB (vergleich)²

VisualComponents (Aurora)³



Um komplexere Szenarien mit zusätzlichen Achsen abbilden zu können, wurde weiterhin eine zusätzliche Kinematik implementiert, die es erlaubt, beliebige Mechaniken abzubilden (z.B.: Roboter auf einer Linearschiene oder ein Tüscharnier). Die bisherige Kinematik war auf den Einsatz von industrietypischen Spherical-Wrist-Robotern beschränkt, jedoch sehr effizient in der Berechnung.

Für die taktzeitoptimierte Generierung der Programme war es auch notwendig, die Spline-Funktionalität der Roboter nutzen zu können. Da auch für die Spline-Bewegungen die Kollisionsfreiheit gewährleistet sein muss, war es notwendig, diese ebenfalls simulieren zu können. Aus diesem Grund wurde die Spline Funktionalität des Roboters nachgebildet (Abbildung 7).

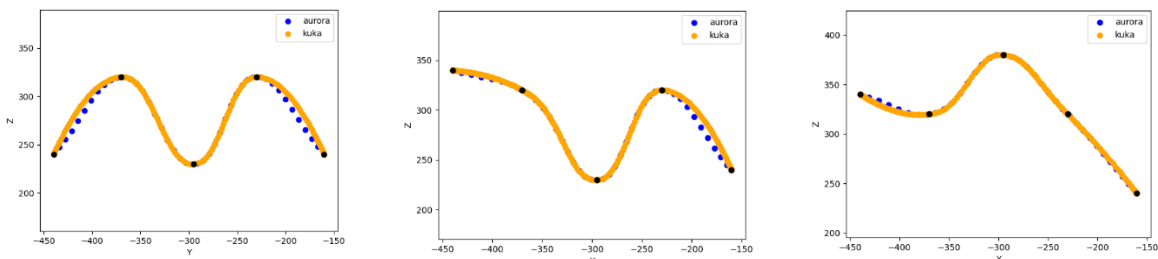


Abbildung 7: Vergleich Spline-Bewegung

Das Aurora-RCS Modul wurde weiterhin um eine Blendingfunktion erweitert (Abbildung 8), womit die RCS Modul Implementierung abgeschlossen wurde.

Folgende Validierungsziele wurden für das Projekt definiert:

- VZ1.1:** 90%ige Zeitreduktion bei Programmen für einfache Szenarien
- VZ1.2:** 50%ige Zeitreduktion bei Programmen für komplexe Szenarien
- VZ1.3:** Ergebnis nicht schlechter als das Programm eines Experten

- VZ2.1:** 33%ige Verringerung der allgemeinen Arbeitstaktzeit (Zeit, die Roboter für die Abarbeitung der einzelnen Effektiv- und Unterstützungsaufgaben benötigt)

¹https://youtu.be/LykikaL_Fdw

²<https://youtu.be/h3D0hbKQq8E>

³<https://youtu.be/2i7luORwpDg>

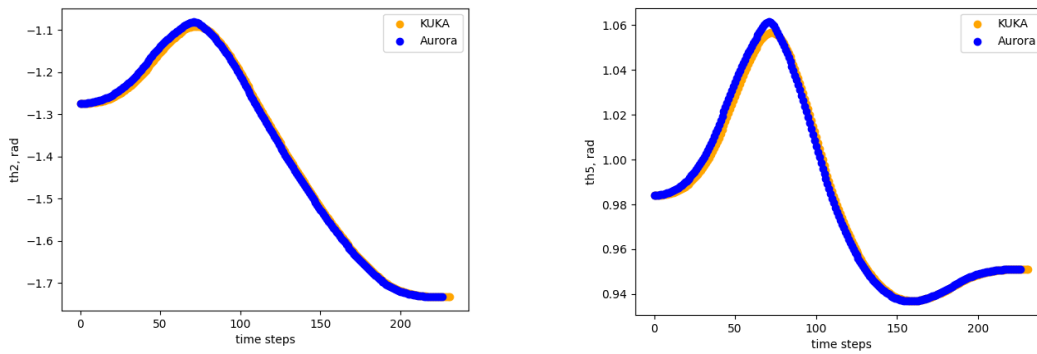


Abbildung 8: Vergleich Blending

VZ2.2: 25%ige Verringerung des Gesamtenergieverbrauchs

VZ2.3: 33%ige Verringerung des Verschleißes

Durch die schon erwähnte Erweiterung des Projektes durch den Verbundpartner Technische Universität Darmstadt wurden auch die Validierungsziele angepasst, indem es bei VZ2.2 und VZ2.3 die Verbesserungen bezüglich Energieverbrauch und Verschleiß mit Hinblick nur auf eine verbesserte Planung (wie z.B. über die Optimierung der Sequenzen) evaluiert werden soll.

Die Validierungsziele VZ1.1 und VZ1.3 konnten erfolgreich validiert werden. VZ1.2 konnte aus technischen Gründen nicht Komplet evaluiert werden, da dafür eine Unterstützung von Roboter-systemen mit mehr als 6 Achsen notwendig ist (zum Beispiel ein Roboter auf einer Linear Schinne). Der Software-Demonstrator konnte zwar solche Szenarien abbilden, die durch Externen-Achsen hinzugefügte Freiheitsgrade wurden aber nicht für die Optimierung benutzt. Diese Komplexität Erhöhung benötigte eine markante Erweiterung der Algorithmen, was im Rahmen der Projektlaufzeit nicht umgesetzt werden konnte.

VZ2.1 konnte für die Testszenarien ebenfalls nicht positiv validiert werden. Im Schweißtestszenario konnte die Taktzeit der Gesamtaufgabe von 91,488s auf 90,408s reduziert werden. Die Dauer der Effektivbewegungen (Schweißen) betrag dabei 83,868s. Die Zeit für die Schweißbewegungen lassen sich Prozessbedingt nicht reduzieren. Die Zeiten für die Unterstützungsbewegungen ließen sich von 7,62s auf 6,54s reduzieren, was einer Reduzierung von 14,2% entspricht. Diese Ergebnisse sind aber Prozess- und Setupabhängig und werden bei z.B. Punktschweißen von 50 Knoten zu viel besseren Verbesserungen führen im Vergleich zu auftragen von 5 geraden Kleberaupen.

VZ2.2 konnte für den evaluierten Roboter nur anhand der Sequenzoptimierung ebenfalls nicht validiert werden. Der Energieverbrauch des untersuchten Kuka KR3 Agilus konnte um 1% reduziert werden. Besonders auffällig dabei war jedoch der erhebliche Energieverbrauch der Robotersteuerung, welcher den Verbrauch des Roboters übersteigt. Für Größere Roboteranlagen kann dieser Wert jedoch erheblich größer ausfallen. Aus Sicht des Projekt-Teams ist jedoch der sinnvollere Weg zur Energieeinsparung von Roboteranlagen die Reduzierung des Verbrauches der Robotersteuerung. Kuka hat mit der KR C5 und C5 micro während der Projektlaufzeit bereits neue Steuerungen auf dem Markt gebracht, welche genau dieses Problem adressieren.

Bei VZ2.3, ähnlich zur VZ2.1, hatte die größte Anteil von Effektivbewegungen (die nicht geändert werden konnten) in Relation zur Unterstützungen (die hinsichtlich des Verschleißes optimiert werden konnten) ebenfalls zur kleineren Verbesserung geführt.

Auch wenn nicht alle Ziele vollständig erreicht werden konnten, wurden die Forschungsergebnisse in bestehende und in der Industrie bereits eingesetzte Simulations- und Roboterprogrammiersoftware integriert. Roboterprogrammierer werden somit weiterhin in ihrer gewohnten Umgebung arbeiten können, erhalten aber eine signifikante automatisierte Unterstützung, die ihnen bereits für eine Fertigungsaufgabe optimierte Roboterprogramme erstellt.

Verwertbarkeit des Ergebnisses

Im Zuge des Projektes wurden im Zuge der durchgeführten Marktanalyse einige potentielle Geschäftsmodelle identifiziert:

- 1) Direkte Lizenzierung der Technologie an OLP-Anbieter
- 2) Plugin Entwicklung für OLP und eigene Vermarktung des Plugins an Endnutzer
- 3) Plugin / Lizenzierung mit Cloudservice

Diese Modelle schließen sich nicht gegenseitig aus und können, abhängig von den zukünftigen Beziehungen zu dem OLP-Anbieter parallel verfolgt werden. Es ist auch möglich, einzelne Komponenten der Technologie herauszulösen und als separates Produkt zu vermarkten (z.B.: Kollisionsfreie Bahnplanung, Aurora-RCS). Dies kann jedoch nur als Zusatzeinnahme betrachtet werden, da es hier bereits sehr viele Wettbewerber am Markt gibt. Eine direkte Lizenzierung wird vor allem von dem großen OLP-Anbieter angestrebt (z.B. KUKA / VisualComponents / ABB).

Um den im Projekt erzeugten Software Prototypen weiter zu entwickeln, wurde ein Antrag zum EXIST-Forschungstransfer zum 31.01.2023 eingereicht. Dabei hat ein großer Teil der Netzwerkpartner des Projektes Ihre Unterstützung des Vorhabens erklärt. Das Team konnte auch die EXIST-Juri überzeugen und somit wurde die Förderung ab September 2023 mit der Laufzeit von 18 Monate genehmigt. Die Ausgründung ist daher voraussichtlich im Q2 2024 geplant.

Beschaffungen:

Die wichtigste Hardware und Software Beschaffungen während des Projektes sind in Tabelle 1 dargestellt.

Der Roboter wurde für den Demonstrator in der Verbindung mit der entwickelten Software beschafft. Vor allem bei Messen-Besuch (Hannover Messe, Automatika und Transfermesse Sachsen-Anhalt) konnte man damit die Gültigkeit von erzeugten Programmen am realen Roboter nachweisen. Außerdem wurden zahlreichen Testdaten mit dem Roboter aufgenommen, um den Aurora-RCS Modul herzustellen.

Im Rahmen des Forschungsprojekts sollen Methoden an zwei Robotersystemen unterschiedlicher Roboterhersteller validiert werden. Langfristiges Ziel ist ein roboterherstellerunabhängiger Einsatz. Da jedes Robotersystem über eine herstellereigenspezifische Programmiersprache programmiert wird, ist jeweils eine geeignete

Schnittstelle zwischen der AuRora-Software und dem Roboter erforderlich, um die Funktion der Software in der Praxis am Robotersystem durchzuführen. Die zweite industrielle OLP-Umgebung (Artiminds RPS) wurde beschafft, weil es im Gegenteil zu KukaSim Roboter von verschiedenem Hersteller unterstützt und auch ein komplett anderes Software-Architektur hat. Dementsprechend wurden bei der Ausschreibung VisualComponents oder darauf basierende OLP Umgebungen (z.B. Delfoi oder Octopuz) explizit ausgeschlossen. Beim Erstellen von notwendigen Schnittstellen und der Spezifikation der Datenformate konnte nachgewiesen werden, dass die entwickelte Lösung flexibel genug ist und in jeder OLP-Umgebung effizient integriert werden kann auch wenn damit verbundene Mehraufwand seitens OLP-Hersteller gibt.

Rechnungsdatum	Zahlungsempfänger	Rechnungsgegenstand	Rechnungsbetrag
20.10.2020	KUKA Augsburg	Roboter	40.461,96 €
22.07.2022	ArtiMinds Robotics GmbH Karlsruhe	Roboterlizenz	17.040,80 €
28.07.2022	ArtiMinds Robotics GmbH Karlsruhe	Softwareentwicklungsprojekt	19.082,84 €
29.08.2022	ArtiMinds Robotics GmbH Karlsruhe	Softwareentwicklungsprojekt 2. Teilrechnung	54.363,96 €

Tabelle 1: Wichtigste Beschaffungen

Voraussichtlicher Nutzen und Erfolge:

Das Projekt hat sich als erfolgreich herausgestellt. Bei den Auftritten auf den Messen und anderen Netzwerk-Veranstaltungen wurde seitens Industrie immer ein großes Interesse an der entwickelten Lösung gezeigt. Der Grund liegt bei dem in industrielle Software integrierten Softwareprototypen, die veranschaulicht die Vorteile der Algorithmik, zeigen konnten. Aufbauend auf die in Projekt identifizierten Geschäftsmodelle und durchgeführte Markanalyse, wurde eine wirtschaftliche Verwertung der erzielten Ergebnisse geplant.

Und es wurde ein Antrag zum EXIST-Forschungstransfer zum 31.01.2023 eingereicht. Dabei hat ein großer Teil der Netzwerkpartner des Projektes Ihre Unterstützung des Vorhabens erklärt (darunter nicht nur KMUs sondern auch VisualComponents und KUKA). Das Team konnte auch die EXIST-Juri mit Ergebnissen überzeugen und somit wurde die Förderung ab September 2023 mit der Laufzeit von 18 Monate genehmigt. Die im Rahmen der EXIST vorgesehene Ausgründung ist daher voraussichtlich im Q2 2024 geplant. Das Team geht davon aus das die Software in der kommenden zwei Jahre marktreif sein kann und somit in vielen Bereichen die Automatisierung voranbringen wird.

Das Team hat sich auch mit den Forschungs- und Entwicklungsergebnissen um den IQ Innovationspreis Mitteldeutschland 2023 im Cluster: Automotive beworben und hier auch Sieg holen konnte. Die Experten-Jury konnte die Vorteile des schnelleren und effizienteren

Entwicklungsprozesses nicht nur für den Fahrzeugbau Markt erkennen, sondern die größere Potenziale für weiteren Automatisierung Bereichen sehen. Dieser Sieg führte auch zum auftreten auf den ACOD Kongress bei Porsche in Leipzig, was viele nützliche Kontakte aus der Automotive Industrie gebracht hat.

Veröffentlichungen:

Während des Projektlaufzeitens wurden folgende Veröffentlichungen der Ergebnisse durchgeführt:

Konferenzbeiträge:

Schillreff N., Scholle J., Kirchheim K., Ortmeier F. (2022) High Speed RCS for Robot Task Sequencing Optimization. In: VDE, (Hrsg.): 54th International Symposium on Robotics (ISR Europe), 2022, ISBN: 978-3-8007-5891-3

Abschlussarbeiten:

Kühne, M. (2020) Automated Collision-free Tasksequencing for Industrial Robots. Universitätsplatz 2, 39106 Magdeburg: Otto-von-Guericke-University Magdeburg.

Brahmann, A. (2021) Ableitung eines Roboterprogramms für die Fertigung von Abdichtnähten in der Automobilindustrie auf Basis von Punktdaten gegebener Abdichtnahtmodelle. Universitätsplatz 2, 39106 Magdeburg: Otto-von-Guericke-University Magdeburg.

Naujoks, M. (2023) Analyse der automatischen Parametrierung eines dynamischen Robotermodells anhand von CAD-Daten zur Berechnung von Drehmoment beschränkten Robotertrajektorien. Universitätsplatz 2, 39106 Magdeburg: Otto-von-Guericke-University Magdeburg.

Chen, Y. (2023) Redundancy optimization of an industrial robot with up to 3DOF external axes. Universitätsplatz 2, 39106 Magdeburg: Otto-von-Guericke-University Magdeburg.