Safety nicht ohne Security in der kollaborativen Robotik

Benjamin Breiling · Bernhard Dieber · Bernhard Reiterer
Andreas Schlotzhauer · Sebastian Taurer

Joanneum Research
vorname.nachname@joanneum.at

Zusammenfassung

Die neueste Generation von Industrierobotern wurde speziell für die Zusammenarbeit zwischen Mensch und Roboter entwickelt. Dabei kann ein kollaborativer Roboter mit einem Menschen gemeinsam an einer Aufgabe arbeiten, muss aber gleichzeitig dessen Unversehrtheit zu jedem Zeitpunkt gewährleisten können. Dazu werden eine Vielzahl an Sicherheitsmechanismen eingesetzt, welche die physische Sicherheit des Robotersystems sicherstellen. Werden diese Mechanismen zugunsten von Flexibilität und Intelligenz in Software realisiert, erhöht dies jedoch gleichzeitig die Anfälligkeit für Angriffe, welche bei üblicher Software im Allgemeinen zum Tragen kommen. Dies bedingt in weiterer Folge, dass physische Sicherheit (Safety) und informationstechnische Sicherheit (Security) beim Design eines derartigen kollaborativen Robotersystems unbedingt gemeinsam betrachtet werden müssen. Dazu demonstrieren wir im Zuge dieses Beitrags wie Safety durch entsprechende Softwarekomponenten intelligenter reagieren kann, welche Angriffsszenarien sich aus diesem Vorgehen ergeben und wie ein kollaboratives Robotersystem durch das Zusammenspiel von Safety und Security sicher aber dennoch flexibel implementiert werden kann.

1 Einführung

In klassischen Anwendungsszenarien für Industrieroboter waren diese bisher hinter Zäunen, Lichtschranken oder anderen Absperrungen platziert, um die physische Sicherheit (Safety) der sich im Umfeld befindlichen Menschen sicherzustellen. Allerdings drängt nun eine neue Generation von Robotern auf den Markt, welche speziell für die Arbeit im engen Kontakt mit dem Menschen konstruiert sind. Bei der Mensch-Roboter-Kollaboration (MRK) arbeiten Mensch und Roboter zur gleichen Zeit an einer gemeinsamen Aufgabe, wodurch die klassischen Sicherheitsmaßnahmen nicht angewendet werden können und physischer Kontakt zwischen Mensch und Roboter möglich bzw. gewollt ist. Um physische Sicherheit dennoch zu gewährleisten, müssen potenzielle Gefahren im Rahmen einer Risikoanalyse und -minderung erkannt und behandelt werden. Dabei hat sich ein in der Norm ISO-12100:2010 [ISOb] festgelegtes mehrstufiges Verfahren zur Risikominimierung etabliert. Im ersten Schritt wird versucht, die MRK-Anwendung durch konstruktive Maßnahmen (leichte Bauweise, große Kollisionsflächen, nachgiebige Strukturen) sicher zu gestalten. Daraufhin werden die Risiken durch technische Maßnahmen wie entsprechende Regelung/Steuerung, Begrenzung der Bewegungsparameter bzw. interne und externe Sensorik weiter minimiert. Erst zum Schluss werden zusätzliche organisatorische Maßnahmen wie Schutzbekleidung und Zugangsberechtigungen in Betracht gezogen. Zurzeit führen Roboter auch in industriellen MRK-Anwendungen klar definierte und repetiti-

P. Schartner · A. Baumann (Hrsg.) · D•A•CH Security 2017 · syssec (2017) 409-420.

ve Aufgaben durch. Um neue Einsatzgebiete zu erschließen und flexiblere Arbeitsabläufe zu ermöglichen, werden zukünftig jedoch immer intelligentere Systeme benötigt, die dynamisch auf ihre Umwelt reagieren können. Mit dieser Dynamik spannt sich ein Raum von möglichen Szenarien und Abläufen auf, der in vielen Fällen nicht durch eine einzige Risikobeurteilung zu erfassen ist. Technische Sicherheitsmaßnahmen, etwa im Bereich der Fusionierung und Auswertung von Sensordaten, werden zunehmend in Software realisiert, um Rekonfigurierbarkeit und Flexibilität zu erreichen. Diese Motive bewirken im Bereich der Ablaufsteuerung, dass starre Programmierung nicht mehr praktikabel ist und verstärkt auf Aufgabenplanungskomponenten aus dem Bereich der künstlichen Intelligenz (KI) gesetzt wird, welche aus einem Wissensmodell des relevanten wahrgenommenen Ausschnitts der Welt und den zu erreichenden Zielen einen Plan von Aktionen ermitteln, die das Robotersystem dann ausführen soll.

Dabei müssen aber dieselben Sicherheitsstandards wie bei herkömmlicher Software angewendet werden, zumal sich die Industrie durch die immer stärkere Vernetzung und die Verwendung von Cloud-Services nun einer höheren Anfälligkeit für Cyber-Attacken gegenüber sieht. Demnach dürfen Vertraulichkeit, Integrität und Authentizität von Informationen in MRK-Anwendungen nicht mehr der Verfügbarkeit untergeordnet werden. Wir schlagen daher vor, Safety und Security als ein gesamtheitliches Thema, das nicht aufgeteilt werden sollte, zu betrachten. Demnach müssen potenzielle Angriffsszenarien in der Risikoanalyse mitberücksichtigt und entsprechende Maßnahmen bezüglich Security ergriffen werden. Im vorliegenden Artikel wollen wir nun die Querverbindung zur physischen Sicherheit näher beschreiben, die Sicherheitsmaßnahmen und ihre Wirksamkeit sowie adäquate Schutzmechanismen auf informationstechnischer Ebene präsentieren. Im Rest dieses Artikels beschreiben wir Safety von Robotern in Abschnitt 2 und Security (Angriffsszenarien und Lösungsansätze) in Abschnitt 3. Wir präsentieren Angriff auf die Safety-Implementierung in einer MRK-Anwendung in Abschnitt 4 und schließen mit Abschnitt 5.

2 Safety – Physische Sicherheit

Das oberste Ziel einer Sicherheitsstrategie für Robotersysteme ist die Unversehrtheit des Menschen. Für unseren Ansatz zur Erreichung dieses Ziels betrachten wir einige Schlüsselaspekte unter dem Gesichtspunkt der Safety: die Behandlung von Risiken, insbesondere im Sinne der relevanten Normen, die sicherheitsgerichtete maschinelle Wahrnehmung und darauf aufgebaut die Aufgaben- und Bewegungsplanung für das Robotersystem.

2.1 Standardkonforme Sicherheit und Risikominderung

Im Zuge der Entwicklung und Verbreitung des Schlüsselelements der MRK, der sensitiven Manipulatoren, sind bereits viele Veröffentlichungen zur sicheren physischen Kollaboration entstanden. Eine potentielle Gefahrensituation in MRK-Anwendungen ist der Zusammenstoß mit dem Menschen, welcher unter anderem an der Universität Mainz im Auftrag des Instituts für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung (IFA) untersucht wurde. Zur Sicherstellung und Beurteilung der Unversehrtheit des Menschen wurden Grenzwerte der biomechanischen Belastungen empirisch ermittelt, welche den Beginn der Schmerzwahrnehmung definieren und durch die wirkende Kraft und den wirkenden Druck während einer Kollision beschrieben sind. Die Ergebnisse sind teilweise in [BGIA11] veröffentlicht. Das Fraunhofer IFF in Magdeburg führte ebenfalls empirische Studien zur Kollision mit dem menschlichen Körper durch um Schmerz- und Verletzungsgrenzen zu bestimmen. Die Arbeiten [HASHR+11] sowie [PHSAS11] behandeln die Kollision mit scharfen Gegenständen bzw. definierten Oberflächen

und [KiSB17] untersucht den Einfluss von verschiedenen Parametern auf die biomechanischen Belastungen während einer Kollision.

Wenn eine MRK-Anwendung in der Europäischen Union in Verkehr gebracht werden soll, so ist die Maschinenrichtlinie [MR-06] heranzuziehen. Um deren Einhaltung nachzuweisen, sind unter anderem die im folgenden Absatz genannten Normen bzw. deren nationale Entsprechungen relevant.

Um physische Sicherheit zu gewährleisten, müssen potentielle Gefahren im Rahmen einer Risikoanalyse und -minderung erkannt und behandelt werden. Die relevanten Normen in diesem Bereich sind die ISO 10218-1/2:2011 [ISOa], welche vorrangig auf der Norm ISO 12100:2010 [ISOb] aufbaut sowie die ergänzende technische Spezifikation ISO/TS 15066:2016 [ISOd]. In der ISO/TS 15066:2016 werden unter anderem biomechanische Grenzwerte für den menschlichen Kontakt mit einem Roboter sowie ein Verfahren für deren Verifikation vorgeschrieben. Wenn programmierbare Hardware oder elektronische Schaltungen für die Ausführung von sicherheitsrelevanten Funktionen zur Risikominderung eingesetzt werden, so muss die Norm [ISOc] oder [IEC] Beachtung finden. Besonders in diesem Punkt wird die Querverbindung zur Systemsicherheit deutlich, und eine Beurteilung der Wirksamkeit der Maßnahmen ist ohne die Verifikation und Validierung der ausführenden Software und Elektronik nicht möglich. Die korrekte und sichere Funktion von risikomindernden Maßnahmen wie eine nachgiebige Regelung, die Begrenzung der Geschwindigkeit oder nur die Überwachung der Position des Manipulators setzen voraus, dass die dafür implementierten Algorithmen korrekte Ergebnisse liefern und nicht manipuliert werden können. Es werden in den relevanten Normen zwar hohe Anforderungen an die Verfügbarkeit der verwendenden Elektronik gestellt (PLr=d, Cat. 3, siehe [ISOd]), die Vertraulichkeit, Integrität und Authentizität der Ergebnisse wird dagegen nicht ausreichend behandelt.

Die genannten Standards gehen stets von einer klar definierten Aufgabe aus, die eine Roboteranwendung zu erfüllen hat. Ein flexibler Einsatz von intelligenten Robotern erfordert eine neue Sichtweise. Neben der Behandlung aller denkbaren Risiken vor dem Betrieb der Anlage muss eine Beurteilung von Risiken auch während dem Einsatz möglich sein. Dazu muss der verwendete Roboter in der Lage sein die erfassten Daten aus der Umwelt zu interpretieren um das Risiko für den Benutzer einschätzen zu können. Ein unter anderem in [PoZR17] beschriebener Ansatz um Risiken dynamisch zu beurteilen ist ein Sicherheitsfeld, welches eine örtliche Verteilung des Risikos ermöglicht. Diese Verteilung ist abhängig von der Bewegung einer Gefahrenquelle relativ zum Benutzer und kann in der Bahnplanung Verwendung finden um Kollisionen zu vermeiden. Dieser Ansatz berücksichtigt allerdings nicht, welches Objekt die Gefährdung verursacht und betrachtet weder ob ein Objekt nachgiebig oder scharfkantig ist. Für eine intelligente dynamische Risikobeurteilung muss der Roboter auch solche Eigenschaften erkennen. In [BeTW15] wird vorgeschlagen Eigenschaften und Regeln zur Handhabung von identifizierten Objekten in einer Wissensbasis, in Form einer öffentlichen Datenbank, verfügbar zu machen, was stark zur Sicherheit beitragen kann.

Die beschriebenen Ansätze sind ausschließlich in Software realisiert. Um sie einsetzten zu können ist eine unbedingte Betrachtung der korrekten und sicheren Funktionsweise der Algorithmen nötig.

2.2 Sicherheitsgerichtete Perzeption

Für ein kollaboratives Robotersystem sind aus mehreren Gründen umfassende maschinelle Wahrnehmungsfähigkeiten erforderlich. Neben den Anforderungen, die sich für Aufgabenplanung, Robotersteuerung, Interaktion mit Menschen und Manipulation von Objekten ergeben, sind verschiedene Sensoren und die weitere Verarbeitung der aus ihnen gewonnenen Daten ein zentraler Baustein für die physische Sicherheit. Die Fähigkeit des Systems, das eigene Verhalten auf Basis von Sensordaten so anzupassen, dass Kollisionen vermieden werden, stellt sowohl eine risikomindernde Maßnahme im Sinne der sicherheitstechnischen Standardkonformität als auch ein Mittel zur dauerhaften Bewältigung einiger Arten von Gefahren dar, die sich im Zuge der Risikoreduktion nicht gänzlich ausschließen lassen.

Eine geeignete Perzeptionsinfrastruktur für Kollaboration kann in zwei Schichten aufgeteilt werden, wobei die untere Schicht aus Sensoren, die obere aus der Sensordaten-Fusion besteht. Zur Einstufung als sicher ist, neben der grundsätzlichen Eignung der gewählten Komponenten für ihre Aufgabe und deren sinnvollen Anordnung, auf beiden Schichten Redundanz erforderlich. Für die Sensoren bedeutet dies, dass auf verschiedene Technologien zurückgegriffen wird – naheliegend sind Laser-Scanner, Time-of-Flight-(ToF)-, Thermal- und RGB-Kameras, Lichtschranken, Mikrofone etc. – und dass diese möglichst mehrfach vorhanden sind. Im Sinne der Ausfallsicherheit ist die Sensorfusion ebenso zumindest zweifach ausgeführt und in der Lage, den Ausfall der jeweils anderen Komponente sowie der Sensoren zu erkennen. Sicherheitsrelevante Informationen müssen zumindest für einfache Reaktionen wie Verlangsamung oder Halt des Roboters über redundante, zuverlässige Kanäle weitergeleitet werden können. Der Weg, über den die fusionierten Daten zur normalen Aufgabenabarbeitung in die Entscheidungskomponenten des Systems gelangen, ist weniger kritisch, zumeist auch in Hinsicht auf zeitliche Nähe. Aufwändigere Datenaufbereitungsmethoden, etwa komplexe Detektionsalgorithmen des maschinellen Sehens, sind für sicherheitsrelevante Zwecke ungeeignet, können aber sehr wohl auf Basis gemeinsam genutzter Sensordaten für Entscheidungen des Systems dienen.

Eine bei uns in Entwicklung befindliche sicherheitsgerichtete Sensor-Fusionsmethode ermittelt einen Safety-Statuswert, der durch die Farben Grün, Gelb und Rot ausdrückt, wie hoch aktuell die Kollisionsgefahr abhängig von detektierten menschlichen Bewegungen in konfigurierbaren Zonen ist. Die Sensordatenfusion ist so gestaltet, dass die Vermeidung von False Positives der Sicherheit untergeordnet ist und bei widersprüchlichen Daten von Sensoren, deren jeweils erfasste Bereiche sich überschneiden, im Zweifelsfall im Sinne der Kollisionsvermeidung entschieden wird. Zusätzliche Sensoren können zur Auflösung solcher Unsicherheiten beitragen. Eine spezielle Herausforderung ist auch durch das Eindringen von Bewegungen des Roboters in überwachte Bereiche gegeben. Diese könnte prinzipiell durch Einbeziehung der im System bekannten bzw. berechenbaren Positionen der Teile des Roboters gelöst werden, allerdings kaum auf sicherheitskonforme Weise, etwa wegen Verzögerungen in der Datenverfügbarkeit oder der Gefahr von Ungenauigkeiten. Hinzu kommt, dass der Roboter auch Teile des menschlichen Körpers für die Sensoren verdecken kann. Die Minimierung von False Positives sollte daher durch eine möglichst optimierte Anordnung des Arbeitsbereichs samt der Sensoren mit Augenmerk auf die Sicherheit erwirkt werden. In laufenden weiterführenden Arbeiten bringen wir spezielle Annäherungssensoren direkt am Roboter an.

Verwandte Arbeiten wie [RAHN+12] und [ScWa13] gehen in Richtung von vereinfachter 3D-Rekonstruktion auf Basis verschiedenartiger Kameras, was auch in unserem Ansatz der Datenfusion abgedeckt ist. Neben der einfachen Erweiterbarkeit der eingesetzten Sensoren sowohl

in ihrer Art als auch in der Anzahl ist einer unserer Beiträge der Fokus auf Sensorhardware, die selbst als sicher gelten kann und die nicht typischerweise für die Detektion von Menschen ausgelegt ist.

2.3 Planen für sichere Kollaboration

Im Bereich des Planens für Robotersysteme ist Systemsicherheit ein gänzlich vernachlässigtes Thema. Kollaboration findet in Grundzügen Beachtung. In manchen Arbeiten, etwa [KLYW15] und [BCOU16], ist diese auf die sich ergebende zeitliche Unsicherheit menschlicher Aufgabenausführung fokussiert, in anderen, z.B. [PaPM16], auf die Kollisionsvermeidung in der Bewegungsplanung, auch angesichts der Unkontrollierbarkeit menschlicher Handlungen. Dies gipfelt in [WDCK16], einer pessimistischen Modellierung des Menschen als Verfolger, dem es zu entkommen gilt. Während zwar in Sicherheitsfragen suboptimale Fälle hohe Aufmerksamkeit verdienen, ist eine solche antagonismusbasierte Sichtweise ungeeignet für weitere Entwicklungen mit dem Ziel, einen Mehrwert aus der gemeinsamen Aufgabenerledigung durch ein gemischtes Mensch-Roboter-Team zu gewinnen. Ein fundamental anders ausgelegter Roboter-Planungsansatz wird in [Köc16] vorgestellt und vereinigt diverse Problemlösungs- und Interaktionstechniken anhand einer umfangreichen formalen Sprache, die auch menschliche Verhaltenseigenschaften und Vorlieben abdeckt. Ein aktueller Überblick zu offenen Safety-Forschungsfragen der KI für die Mensch-Roboter-Interaktion [FrZi16] identifiziert drei Kategorien von Mängeln, die in ihrer Kombination ungelöst sind. Diese betreffen die Berücksichtigung von Menschen in der Modellierung, das Schaffen von wechselseitigem Verständnis und den vorausblickenden Umgang mit Fehlschlägen.

Der innere Planungsvorgang basiert in typischen Ansätzen des klassischen Planens auf formalen Domänenbeschreibungen gemäß der Sprachfamilie PDDL [FoLo03, GHLS$^+$09]. Kollaboration erfordert neuartige Praktiken bei der Domänenmodellierung. In PDDL würde man etwa diverse menschliche Faktoren als Fakten und Funktionen abbilden oder Kommunikationsschritte als Aktionen beschreiben. Safety-Aspekte können auf ähnliche Weise in die Domäne einfließen. Durch die Übertragung des zuvor erwähnten Sicherheitsstatuswerts in das Planungssystem sind wir in der Lage, Aktionen beim Planen abhängig von der Sicherheitssituation zu erlauben bzw. zu verbieten oder Aktionen unterschiedlich zu parametrisieren, z.B. maximale Robotergeschwindigkeit oder zu vermeidende Teile des Arbeitsbereichs als Faktoren für die nachgeschaltete Bewegungsplanung. Die Exekutive, welche den Plan in die Tat umsetzt, muss die Fähigkeit besitzen, auf festgestellte Abweichungen von Annahmen, die beim Planen getroffen wurden, geeignet zu reagieren. Dies umfasst das Finden eines neuen Plans ausgehend von der aktuellen Situation und dessen In-Gang-Setzung. Eine musterhafte Umsetzung dieses Prinzips ist in dem Roboter-Planungsframework ROSPlan [CFLM$^+$15] zu finden. Handelt es sich bei der Abweichung um einen strengeren Sicherheitsstatus, ist allerdings neben dem neuerlichen Planen auch das unmittelbare Auslösen einer Verlangsamung oder eines Halts des Roboters notwendig. Jenseits der Domänenmodellierung sind bei uns weiterführende Ideen in Ausarbeitung, die physische Sicherheit auch bei der Gestaltung und Einbindung des Planungsvorgangs an sich zu berücksichtigen, etwa durch die ständige Verfügbarkeit eines Reserve-Plans für den Fall des Eintretens eines sicherheitsrelevanten Ereignisses, auf den dann rasch gewechselt werden kann.

Ein Planungssystem ist durch seine zentrale Rolle zwischen sensorischen und agierenden Systemteilen einerseits sowie durch seine interne Zusammensetzung aus mehreren Komponenten andererseits an zahlreichen Kommunikationskanälen in sendender und/oder empfangender Rolle beteiligt. Mit ROSPlan als typisches Beispiel auf der Infrastruktur des Robot Operating

System (ROS) [QCGF+09] bedeutet das eine Vielzahl von Topics und Services, auf welchen die Richtigkeit der Wissensgewinnung, der inneren Abläufe und der Aktionsumsetzung der Planungskomponente beruhen.

3 Security – Sicherheit vor Angriffen

Bisherige Arbeiten zur Sicherheit in Robotersystemen zielen häufig primär auf den korrekten und verlässlichen Ablauf der auszuführenden Tätigkeit bzw. des auszuführenden Programmes ab. Allerdings ist die Implementierung von IT-Security unabdingbar, wenn es darum geht, Verlässlichkeit und Korrektheit in diesem Zusammenhang zu gewährleisten. Die Tatsache, dass Safety-Funktionen zunehmend auch in Software realisiert werden, macht diese auch durch Cyber-Angriffe verwundbar. Wie in Abschnitt 2.3 bereits beschrieben, ist ein Planungssystem für kollaborative Robotik auf die Informationen von einer Vielzahl von Sensoren angewiesen. Des Weiteren steuert ein derartiges System die Aktionen, die von dem Robotersystem in weiterer Folge ausgeführt werden. Dabei ist die Sicherheit der im gesamten Robotersystem kommunizierten Daten im Sinne von Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität (CIA+) stets zu gewährleisten, um den Safety-Anforderungen auch gerecht werden zu können. Während der Bedarf nach Integrität, Verfügbarkeit und Authentizität klar erkennbar ist, kann über Vertraulichkeit im Zusammenhang mit Safety durchaus diskutiert werden. Hierbei ist allerdings zu bedenken, dass allein das Wissen über die im gesamten Robotersystem transferierte Information die Ausführung eines Angriffs auf ein Safety-Modul erheblich erleichtern kann. Am Beispiel des Robot Operating System (ROS) [QCGF+09] werden wir in Abschnitt 4 zeigen wie die physische Sicherheit in einem ROS-basierten Robotersystem angegriffen werden kann. Dazu stellen wir in diesem Abschnitt das entsprechende Angriffsszenario auf ROS vor, um in weiterer Folge Gegenmaßnahmen zu diskutieren.

3.1 Angriffsszenarien auf ROS

ROS, als Middleware für Roboter-Systeme, bietet in verteilten Systemen häufig verwendete Kommunikationsmuster, wie Publish/Subscribe, Server/Client und Remote-Procedure-Call (RPC) an. Im Mittelpunkt der Kommunikation steht ein zwingend erforderlicher Master-Prozess welche den Aufbau des ROS-Netzwerkes steuert. Dabei wird das aktuelle Netzwerk intern als Graph, dem sogenannten ROS-Graph gespeichert. Die Knoten in diesem Graph entsprechen den kommunizierenden Prozessen, die Kanten stellen die jeweiligen Kommunikationsbeziehungen dar. Für administrative Vorgänge, wie das Registrieren als Publisher, Subscriber oder Service-Server bzw. für die Inanspruchnahme eines Services wird XML-RPC verwendet. Darüber hinaus können auch Informationen über den gegenwärtigen Zustand des ROS-Netzwerks (Liste der ROS-Knoten, Kommunikationsbeziehungen, etc.) mithilfe von XMLRPC abgefragt werden. Der eigentliche Datenaustausch zwischen den ROS-Knoten findet dann über ROS-Topics (Publish/Subscribe) und ROS-Services (Server/Client) statt. Bezüglich CIA+ bietet ROS keinerlei Mechanismen zur Sicherstellung dieser Schutzziele. In diesem Zusammenhang wurde die Verwundbarkeit von ROS bereits in [StPo14, MSFMn13] evaluiert. In diesem Abschnitt stellen wir ein Angriffsszenario vor, bei dem einzelne Knoten von der Kommunikation innerhalb des ROS-Netzwerkes isoliert und zusätzlich falsche Informationen eingeschleust werden können, ohne dass dies im vom ROS-Master verwalteten ROS-Graphen sichtbar wird. Bei folgendem Angriff nutzen wir die Tatsache aus, dass es keinerlei Einschränkungen für den Zugriff auf die beim Master gespeicherten Informationen gibt und dass für das Senden und Empfangen von Daten keine Authentifizierung erforderlich ist. Das Se-

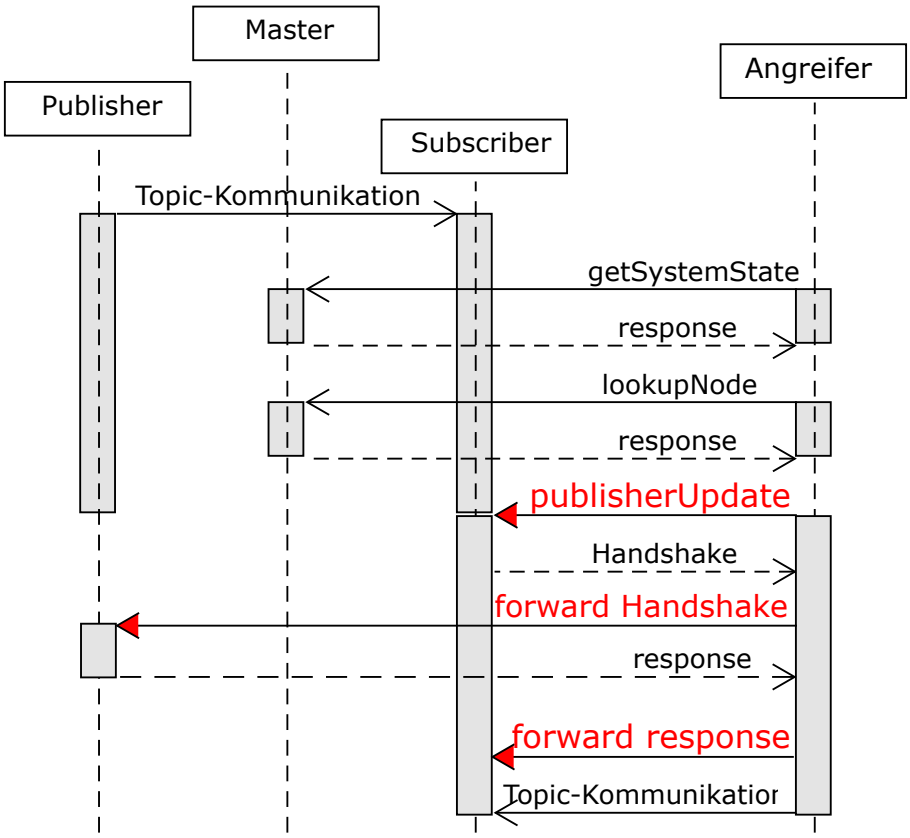

Abb. 1: Fake-Publisher-Angriff

quenzdiagramm aus Abbildung 1 beschreibt den Ablauf dieses Angriffs, welcher von nun an als Fake-Publisher-Angriff bezeichnet wird. Dabei wird ein Subscriber von der Kommunikation auf einem Topic isoliert und ein Publisher gestartet, welcher diesem Subscriber falsche Informationen sendet.

Dazu wird zuerst die Methode `getSystemState` via XMLRPC beim Master aufgerufen um so einen Überblick über das ROS-Netzwerk zu erhalten. Im Anschluss daran erfolgt ein Aufruf der Methode `lookupNode`, was die URI des anzugreifenden Subscribers liefert. Im Anschluss daran schickt der Angreifer einen `publisherUpdate`-Aufruf zum Subscriber. Dieser Aufruf enthält eine Liste mit den aktuellen Publishern zu einem bestimmten Topic als Parameter, wobei diese Liste hier nur die URI des falschen Publishers enthält. Damit wird die Verbindung zu allen anderen Publishern beendet und eine Verbindung zum falschen Publisher aufgebaut. Dieser leitet den vom Subscriber gesendeten Header zum richtigen Publisher weiter um eine korrekte Antwort zu erhalten, die er dann an den Subscriber zurücksenden kann. Nun empfängt der Subscriber nur mehr die Nachrichten des falschen Publishers. Da der ROS-Master nur bei den ersten beiden XMLRPC-Calls involviert ist, werden nachfolgende Änderungen auch nicht im ROS-Graph angezeigt. Der Angreifer erhält hier nicht nur einen Überblick über das ROS-Netzwerk, viel mehr kann er das Verhalten von einzelnen Prozessen im Netzwerk gezielt beeinflussen. Bei Prozessen, welche die physische Sicherheit der Anwendung steuern, kann ein derartiger Angriff zu Verletzungen oder gar zum Tod von Menschen im Umfeld des Roboters führen. Um ein solches Szenario zu verhindern muss das Robotersystem, im konkreten Fall das ROS-Netzwerk, vor derartigen Angriffe geschützt werden.

3.2 ROS-Security

Die eben beschriebenen Angriffe auf ROS basieren zum einen auf der Tatsache, dass die vom ROS-Master verwalteten Informationen über das Gesamtsystem mithilfe von XMLRPC-Anfragen relativ leicht extrahiert werden können. Zum anderen kann die Sicherheit der Kommunikation in einem ROS-Netzwerk im Sinne von CIA+ nicht gewährleistet werden. Eine Möglichkeit die Extraktion von Systemdaten beim ROS-Master zu verhindern wäre, diesen nach der Initialisierung des Systems abzuschalten. Dies hätte allerdings den Nachteil, dass keine neuen Systemkomponenten in die Kommunikation eingebunden werden könnten und der äußerst nützliche Parameter-Server nicht mehr zur Verfügung stehen würde. Können die sich daraus ergebenen Nachteile nicht in Kauf genommen werden, muss der Zugriff auf den ROS-Master reguliert werden. Dazu wird in [HEZM+14] das Runtime Verification Framework (ROSRV) vorgestellt. Dabei wird ein zusätzlicher Knoten, der sogenannte RV-Master, gestartet, welcher die Einhaltung einer zuvor vorgegebenen Security-Policy bezüglich der Kommunikation im gesamten ROS-Netzwerk überprüft. Es wird somit festgelegt, welcher Knoten welche XMLRPC-Anfragen an den Master senden darf. Außerdem wird hier auch der Zugriff auf Topics und Services reguliert. Damit kann weder der ROS-Graph durch das Senden des entsprechenden XMLRPC-Calls ohne weiteres ausgelesen, noch können Topic-Informationen durch das Starten eines gewöhnlichen Subscribers ausgeschleust werden. In [DPRS16] wird die Kommunikation zwischen den Knoten abgesichert, indem sie sich zuvor bei einem zentralen Authentifizierungsserver mittels Zertifikat authentifizieren müssen. Dieser stattet die Knoten mit Sessions-Keys aus, die daraufhin zur Verschlüsselung und zum Signieren der auszutauschenden Nachrichten verwendet werden. Dadurch ist zwar nicht ausgeschlossen, dass sich unautorisierte Knoten über den ROS-Master registrieren, allerdings können empfangene Nachrichten nicht entschlüsselt und gesendete Nachrichten nicht entsprechend signiert werden. Beide Ansätze arbeiten auf Anwendungsebene, wodurch beispielsweise nicht verhindert werden kann, dass ein Angreifer direkt mit einem Knoten auf Transportebene kommuniziert. Um derartigen Szenarien entgegen zu wirken, muss allerdings die Implementierung von ROS selbst hinsichtlich CIA+ geändert werden. Dazu wird in [BrDS17] die Kommunikation zwischen ROS-Knoten via TCP und UDP durch die Verwendung von TLS und DTLS abgesichert. Dies bedeutet, dass der Besitz eines vertrauenswürdigen Zertifikats unbedingt erforderlich ist um Publisher-Subscriber bzw. Server-Client-Verbindung zu einem C++-basierten ROS-Knoten aufzubauen, während jedoch Python-Knoten nach wie vor ungesichert sind und die Kommunikation via XMLRPC nicht berücksichtigt ist, weshalb nur einem Teil der Angriffsvektoren entgegen gewirkt wird. Im Gegensatz dazu werden in SROS[1] die Master-Knoten- und die Knoten-Knoten-Kommunikation abgesichert. Allerdings bezieht sich dies nur auf die Python-Implementierung von ROS, wodurch ein großer Teil der ROS-Applikation von dem Security-Konzept ausgeschlossen wird. Beide Konzepte haben darüber hinaus den Nachteil, dass keine gemischten Netzwerke aus Python- und C++-basierten Knoten aufgebaut werden können. Alle eben vorgestellten Konzepte stellen nur Teillösungen in Hinblick auf IT-Sicherheit dar. Werden die Ansätze allerdings zusammen mit einem adäquaten Key-Management kombiniert, kann man ROS gegen einen erheblichen Teil der möglichen Angriffe absichern. Zusammengefasst kann gesagt werden, dass nur eine gesamtheitliche Lösung die postulierten Anforderungen bezüglich CIA+ erfüllen kann.

[1] `http://wiki.ros.org/sros`

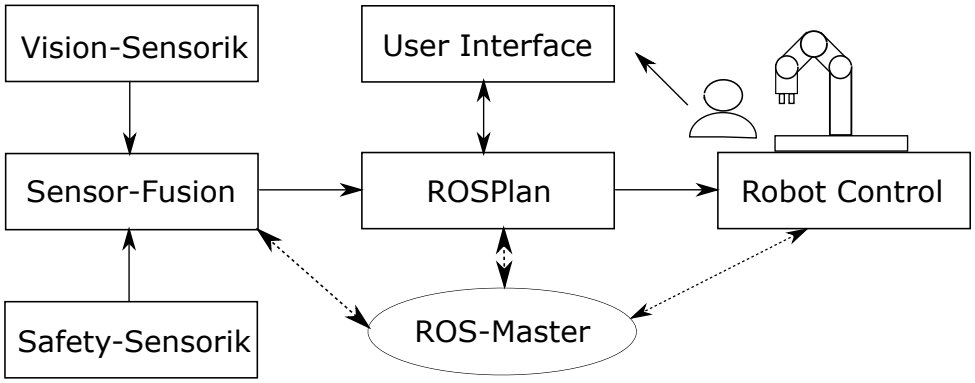

Abb. 2: Demonstrationsaufbau

4 Demonstrationsaufbau

In einem Versuchsaufbau in unserem Labor, wird der in [BYRH17] beschriebene Ansatz initial umsetzt. Dabei gehen wir von einem Szenario aus, in dem ein Mensch und ein zur Kollaboration geeigneter Roboter gemeinsam eine Assembly-Aufgabe lösen. Dabei variiert bei den Teilen die Vorgabe, ob sie vom Roboter oder vom Menschen montiert werden müssen oder ob beides möglich ist. An sensorischen Komponenten verwenden wir u.a. eine Kombination aus Machine-Vision-Algorithmen und ein dazu notwendiges Kameraensemble für 3D-Pose-Estimation der Teile. Für die sicherheitsgerichtete Wahrnehmung sind zwei annähernd nach unten gerichtete ToF-Kameras oberhalb des Arbeitsbereichs angebracht. Die Blickfelder dieser Sensoren werden von zwei waagrechten Ebenen geschnitten, die von jeweils einem Laser-Scanner überwacht werden. In der darauf aufgebauten Datenfusion wird wie zuvor beschrieben unter Berücksichtigung konfigurierter überwachter Zonen ein aktueller Sicherheitsstatus berechnet. Als Planungskomponente ist ROSPlan mit einer dem Szenario entsprechenden Domänenbeschreibung im Einsatz. Vorgeschalten ist eine Wissensgewinnungskomponente, welche laufend die relevanten sensorischen Daten, etwa die Posen der zu manipulierenden Teile, in für das Planen verwertbare Fakten konvertiert. Der festgestellte Sicherheitsstatus fließt nicht nur dort, sondern auch direkt in das Robotersteuerungssystem ein, das die Ausführung der vom Planungssystem erhaltenen Bewegungskommandos in Bezug auf die Geschwindigkeit entsprechend anpassen oder einen Nothalt auslösen kann. Der Mensch erhält Anweisungen für durch ihn zu erledigende Schritte per Sprachausgabe und einer grafischen Benutzeroberfläche. Ein solches System realisiert also Safety-Aspekte in Software und muss daher auch informationstechnisch geschützt werden. Konkret ist die Integration der Komponenten mit ROS realisiert, daher treffen auch alle oben beschriebenen Sicherheitslücken zu.

Die Komponente Sensor-Fusion veröffentlicht den berechneten Safety-Wert auf einem ROS-Topic. Die Planungskomponente und der Roboter-Controller starten jeweils einen Subscriber-Prozess welcher den veröffentlichten Safety-Wert kontinuierlich ausliest und dementsprechend die Geschwindigkeit der auszuführenden Bewegung anpasst. Grundsätzlich würde es für einen Angriff genügen einen ROS-Knoten zu starten, welcher falsche Safety-Werte auf dem Topic mit hoher Frequenz veröffentlicht. Damit würde die Bewegungsplanung bereits empfindlich gestört werden. Allerdings würde man den Angreifer-Prozess dauerhaft im ROS-Graphen sichtbar machen. Mit dem in Abschnitt 3.1 beschriebenen Fake-Publisher-Angriff kann nicht nur die Anwesenheit des Angreifers verschleiert, sondern die Sensor-Fusion als rechtmäßiger Publisher

von der Kommunikation ausgeschlossen werden. Damit erhält der Angreifer volle Kontrolle über die Geschwindigkeit mit welcher der Roboter seine Bewegungen ausführt. Dies kann dazu führen, dass sich der Roboter mit der für diese Anwendung maximalen Geschwindigkeit bewegt, obwohl sich ein Mensch in einem Safety-kritischen Bereich befindet.

5 Schlussfolgerung

Safety und Security sind für sich betrachtet bereits vielfach diskutierte und zum Teil standardisierte Aspekte in der (kollaborativen) Robotik. Allerdings kann weder das eine noch das andere für sich genommen die Unversehrtheit von Menschen im direkten Umfeld von Robotern garantieren, womit diese beiden Themen in diesem Gebiet untrennbar miteinander verknüpft sind. Im Zuge dieser Arbeit stellen wir daher Safety und Security als unbedingt erforderliches ganzheitliches Konzept für kollaborative Robotik vor. Dazu wurden sowohl Aspekte der Safety (Risikominimierung, Perzeption, sichere Planung) als auch der Security (Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität) diskutiert und zueinander in Beziehung gesetzt. Anhand eines Beispiels wurde gezeigt wie Safety in einem Robotersystem dynamisch realisiert werden kann und welche Angriffsmöglichkeiten sich ergeben, wenn keine Maßnahmen bezüglich Security ergriffen werden. Wir erachten es daher für wichtig, potenzielle Angriffe auf die Informationssicherheit in einem Robotersystem bei Verfahren der Risikoanalyse und Risikominimierung zu berücksichtigen. In weiterführenden Schritten werden wir uns auch noch verstärkt mit der für standardkonforme Sicherheit erforderlichen Anwendung von Verifikation und Validierung auf im Robotersystem eingesetzte Softwarekomponenten auseinandersetzen. Dazu antizipieren wir besonders im Bereich der Aufgabenplanung aufgrund der hohen Dynamik eine Häufung von Herausforderungen.

Danksagung

Die in diesem Beitrag veröffentlichte Arbeit wurde vom österreichischen Bundesministerium für Transport, Innovation und Technologie (bmvit) im Zuge des Projektes Collaborative Robotics gefördert.

Literatur

[BCOU16] G. Bernardi, A. Cesta, A. Orlandini, A. Umbrico: Dynamic Task Planning for Safe Human Robot Collaboration. *In: A. Shafti, A. Orlandini (Hrsg.), Proceedings of the 1st Workshop on Planning, Scheduling and Dependability in Safe Human-Robot Interactions*, London, UK (2016), Bd. 1, 26–33.

[BeTW15] M. Beetz, M. Tenorth, J. Winkler: Open-EASE. *In: 2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), 1983–1990.

[BGIA11] BGIA: BG/BGIA risk assessment recommendations according to machinery directive. Tech. Rep., Institute for Occupational Safety and Health of the German Social Accident Insurance (2011).

[BrDS17] B. Breiling, B. Dieber, P. Schartner: Secure communication for the Robot Operating System. *In: Proceedings of the 11th IEEE International Systems Conference*, IEEE (2017), 360–365.

[BYRH17] I. Brijacak, S. Yahyanejad, B. Reiterer, M. Hofbaur: Toward Safe Perception in Human-Robot Interaction. *In: Proceedings of the OAGM & ARW Joint Workshop* (2017), 87–92.

[CFLM+15] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtós, M. Carreras: ROSPlan: Planning in the Robot Operating System. *In: R. I. Brafman, C. Domshlak, P. Haslum, S. Zilberstein (Hrsg.), Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015.*, AAAI Press (2015), 333–341, .

[DPRS16] B. Dieber, M. Pichler, S. Rass, P. Schartner: Sicherheit für ROS-basierte Applikationen auf Anwendungsebene. *In: DACH Security 2016* (2016).

[FoLo03] M. Fox, D. Long: PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. In: *J. Artif. Intell. Res. (JAIR)*, 20 (2003), 61–124, .

[FrZi16] R. G. Freedman, S. Zilberstein: Safety in AI-HRI: Challenges Complementing User Experience Quality. *In: 2016 AAAI Fall Symposium Series* (2016).

[GHLS+09] A. E. Gerevini, P. Haslum, D. Long, A. Saetti, Y. Dimopoulos: Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. In: *Artificial Intelligence*, 173, 5–6 (2009), 619 – 668, , advances in Automated Plan Generation.

[HASHR+11] S. Haddadin, A. Albu-Schaffer, F. Haddadin, J. Rosmann, G. Hirzinger: Study on Soft-Tissue Injury in Robotics. In: *IEEE Robotics Automation Magazine*, 18, 4 (2011), 20–34.

[HEZM+14] J. Huang, C. Erdogan, Y. Zhang, B. Moore, Q. Luo, A. Sundaresan, G. Rosu: ROSRV: Runtime Verification for Robots, Springer International Publishing, Cham (2014), 247–254, .

[IEC] IEC 62061:2005, Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems. Tech. Rep.

[ISOa] ISO 10218-1/2:2011 Robots and robotic devices - Safety requirements for industrial robots. Tech. Rep.

[ISOb] ISO 12100:2010, Safety of machinery - General principles for design - Risk assessment and risk reduction. Tech. Rep.

[ISOc] ISO 13849-1:2015, Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design. Tech. Rep.

[ISOd] ISO/TS 15066:2016, Robots and robotic devices - Collaborative robots. Tech. Rep.

[Köc16] U. Köckemann: Constraint-based Methods for Human-aware Planning. Dissertation, Örebro University, School of Science and Technology, Örebro University, Sweden (2016), .

[KiSB17] D. Kirschner, A. Schlotzhauer, M. Brandstötter: Validation of Relevant Parameters of Sensitive Manipulators for Human-Robot Collaboration. *In: To appear in Proceedings of the 26th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD)* (2017).

[KLYW15] E. Karpas, S. J. Levine, P. Yu, B. C. Williams: Robust Execution of Plans for Human-Robot Teams. *In: R. I. Brafman, C. Domshlak, P. Haslum, S. Zilberstein*

(Hrsg.), Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015., AAAI Press (2015), 342–346, .

[MR-06] Richtlinie 2006/42/EG des Europäischen Parlaments und des Rates. Tech. Rep., Europäische Union (2006).

[MSFMn13] J. McClean, C. Stull, C. Farrar, D. Mascareñas: A preliminary cyber-physical security assessment of the Robot Operating System (ROS). *In: Proc. SPIE* (2013), Bd. 8741, 874110–874110–8, .

[PaPM16] J. S. Park, C. Park, D. Manocha: Safe Motion Planning for Human-Robot-Interaction. *In: A. Finzi, E. Karpas (Hrsg.), Proceedings of the 4th Workshop on Planning and Robotics (PlanRob)*, London, UK (2016), 127–130.

[PHSAS11] J. J. Park, S. Haddadin, J. B. Song, A. Albu-Schäffer: Designing optimally safe robot surface properties for minimizing the stress characteristics of human-robot collisions. *In: IEEE International Conference on Robotics and Automation* (2011), 5413–5420.

[PoZR17] M. P. Polverini, A. M. Zanchettin, P. Rocco: A computationally efficient safety assessment for collaborative robotics applications. In: *Robotics and Computer-Integrated Manufacturing*, 46 (2017), 25 – 37, .

[QCGF+09] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng: ROS: an open-source Robot Operating System. *In: ICRA workshop on open source software* (2009), Bd. 3, 5.

[RAHN+12] P. E. Rybski, P. Anderson-Sprecher, D. Huber, C. Niessl, R. G. Simmons: Sensor fusion for human safety in industrial workcells. *In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012* (2012), 3612–3619, .

[ScWa13] B. Schmidt, L. Wang: Contact-less and Programming-less Human-Robot Collaboration. In: *Procedia {CIRP}*, 7 (2013), 545 – 550, .

[StPo14] L. Stroetmann, H. Pohl: Robot Operating System (ROS): Safe & Insecure. Technical report, SoftScheck GmbH (2014).

[WDCK16] Y. Wang, N. T. Dantam, S. Chaudhuri, L. E. Kavraki: Task and Motion Policy Synthesis as Liveness Games. *In: A. J. Coles, A. Coles, S. Edelkamp, D. Magazzeni, S. Sanner (Hrsg.), Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016.*, AAAI Press (2016), 536, .