

Aufbau und Eigenschaften einer offenen, adaptierbaren Robotersteuerung für chirurgische Anwendungen

Stephanie Sahn und Jürgen Wahrburg

Zentrum für Sensorsysteme, Universität Siegen, Deutschland
Kontakt: wahrburg@zess.uni-siegen.de

Einleitung

Um ein Robotersystem sinnvoll und effektiv im Operationssaal einsetzen zu können, müssen sowohl der Roboterarm als auch die Robotersteuerung besondere Anforderungen erfüllen. Grundsätzlich gibt es zwei verschiedene Ansätze: Zum Einen die Verwendung eines Standard-Industrieroboters, zum Zweiten den Entwurf eines speziellen Chirurgieroboters von Grund auf. Die erste Lösung bietet relativ preiswert industriell erprobte Funktionalität, aber ihre Betriebsmodi sind für den OP wenig geeignet, während die zweite Lösung die optimale Anpassung an die OP-Umgebung ermöglicht, aber sehr aufwändig und teuer ist. Dieser Beitrag beschreibt die Kombination von Standard-Roboterarmen mit einer selbst entwickelten flexiblen Steuerung als Vorschlag zur Lösung dieses Zielkonfliktes. Die Steuerung ermöglicht sowohl die Adaption an verschiedene Roboterarme als auch die Implementierung der zur Unterstützung verschiedener chirurgischer Eingriffe benötigten Funktionen. Diese Funktionen können über eine offene Schnittstelle von bestehender Planungs- und Steuerungssoftware aufgerufen werden.

Methoden und Materialien

Architektur der Robotersteuerung

Die Abb. 1 zeigt als Gegenüberstellung die Komponenten eines Industrierobotersystems (links) und den Aufbau der selbst konzipierten Lösung (rechts). Eine Industriesteuerung besteht in aller Regel aus zwei wesentlichen Komponenten: a) der Leistungselektronik mit Funktionen zur Regelung der einzelnen Gelenkbewegungen (Layer 2 in Abb. 1), und b) einer Steuerkomponente zur Generierung der koordinierten Gesamtbewegung basierend auf dem gespeicherten Programm (Layer 3). Unser Ansatz beruht darauf, Standardkomponenten bis einschließlich Layer 2 zu verwenden, und die Funktionen in Layer 3 und Layer 4 (Benutzer-Interface) aufgabenspezifisch selbst zu entwickeln.

Als Hardware wird dazu ein System aus zwei über eine Ethernetverbindung gekoppelten PCs eingesetzt, wobei ein Rechner mit einem Echtzeit-Betriebssystem die echtzeitkritischen Aufgaben ausführt, während auf dem zweiten Rechner unter Windows Verwaltungs- und Koordinierungsfunktionen laufen. Dieser Rechner verfügt über eine von uns spezifizierte, offengelegte Softwareschnittstelle zur Software in Layer 4, dem XML-basierten HNI (Host-Network-Interface). Darüber können Kommandos an das

Robotersystem geschickt und wichtige Parameter und Systeminformationen ausgelesen werden. Wir verwenden diese Schnittstelle selbst, um das Robotersystem mit der von uns zur Planung und Durchführung verschiedener chirurgischer Anwendungen entwickelten Software zu steuern. Sie kann aber ebenso von unabhängig entwickelter Software genutzt werden, die auf einem dritten, über TCP/IP verbundenen Rechner installiert ist. Anwender, die eigene Planungs- und Steuerungsfunktionen für robotergestützte Eingriffe entwickeln, können das Robotersystem auf diese Weise in ihre Anwendung einbinden. Eine ausführlichere Beschreibung des HNI findet sich in [1].

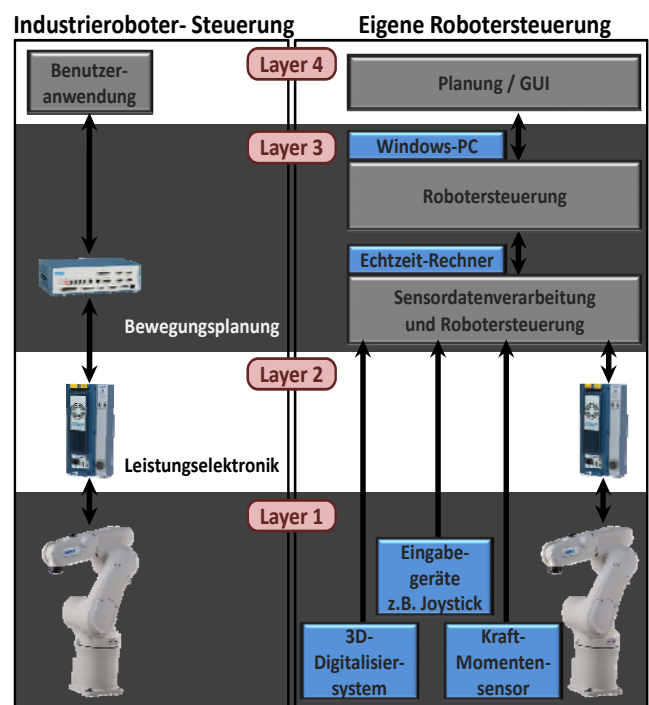


Abb. 1: Hardwarekomponenten der Robotersteuerung

Schnittstelle zu kommerziellen Roboterarmen

Die Steuerung ist zunächst auf die Verwendung von Gelenkarm-Robotern mit 6 Freiheitsgraden ausgelegt, deren Eigenschaften (Aufbau und Kinematik des Arms) der Anwendung entsprechend gewählt werden. Voraussetzung ist, dass die vom Roboterhersteller angebotene Leistungselektronik über offene Schnittstellen verfügt, um mit ihr zu kommunizieren und die Hersteller-Steuerung in Layer 3 zu ersetzen. Erfreulicherweise hat hier die Bereitschaft der Roboterhersteller, entsprechende Lösungen an-

zubieten, im Lauf der letzten Jahre deutlich zugenommen (siehe auch unten beschriebene Beispiele). Dabei können abhängig davon, welche Gelenkregelfunktionen der Leistungselektronik in Layer 2 genutzt werden sollen, Sollwerte für die Gelenkpositionen, die Gelenkgeschwindigkeiten oder auch für das gewünschte Drehmoment vorgegeben werden.

Übersicht der Hardwarekomponenten

Außer dem Roboterarm selbst können auch Sensoren wie die Kamera eines optischen 3D-Digitalisiersystems oder ein Kraft-Momenten-Sensor und Eingabegeräte wie ein Joystick oder Fußtaster an die selbst entwickelte Robotersteuerung angeschlossen werden. Die Sensordaten werden je nach gewähltem Betriebsmodus ausgewertet und zur Generierung der Roboterbewegung herangezogen. Insbesondere wird durch die Sensorintegration das von uns bereits seit längerem verfolgte Konzept eines haptisch fühlbaren, navigierten Assistenzroboters realisiert [2].

Basisfunktionen der Robotersteuerung

Die implementierte Software umfasst die Funktionen zur Generierung der Positions- oder Geschwindigkeits-Sollwerte für die Roboterbewegung in Gelenkkoordinaten. Sie kann in einen roboterunabhängigen und einen roboterabhängigen Teil untergliedert werden. Zu ersterem gehören beispielsweise

- Trajektorienplaner für den Gelenkraum und den kartesischen Raum,
- Regelalgorithmen zur Positionsregelung der Gelenke bei Generierung von Geschwindigkeits-Sollwerten für die Gelenke,

Die roboterabhängigen Funktionen umfassen unter anderem

- die Vorwärts- und Rückwärtskinematik,
- eine singularitätsrobuste Geschwindigkeitskinematik,
- Kommunikation mit der Leistungselektronik.

Ergebnisse und Diskussion

Prototypen der adaptierbaren Steuerung sind bisher für den MHI (Mitsubishi Heavy Industries) Roboter PA10 (Arcnet-Schnittstelle zur Leistungselektronik) und den Adept Roboter Viper S850 (serielle Schnittstelle oder Firewire) aufgebaut worden. Die verwendete Taktfrequenz liegt zwischen 2 und 10 ms.

Die Roboterbewegung erfolgt unter Ausnutzung der oben beschriebenen Funktionen in einem der folgenden Betriebsmodi:

- Abfahren einer geplanten Trajektorie zu vorgegebenen Zielposen im Gelenk- und im kartesischen Raum.
- Bewegung des Roboters durch Menüsteuerung oder Joystick in verschiedenen Koordinatensystemen mit einstellbarer Geschwindigkeit, auch durch Singularitäten.
- Haptische Führung des Roboters über Handgriffe am Flansch des Roboters.

Die singularitätsrobuste inverse Geschwindigkeitskinematik erlaubt dabei auch eine sichere Bewegung durch Singularitäten [3].

Außerdem ist es möglich, die Bewegungen des Roboters durch sog. „virtual fixtures“ zu beschränken. Damit kann das am Roboter befestigte Instrument beispielsweise nur entlang seiner Achse oder in einer Ebene bewegt oder um einen Punkt pivotiert werden [4]. Das Risiko, sensible Patientenstrukturen unbeabsichtigt zu verletzen, wird so erheblich reduziert.

- Online-Nachführung des Roboters bei kleinen Patientenbewegungen.

Durch die Integration der Messdaten des 3D-Digitalisiersystems konnte eine Werkzeugnachführung in Echtzeit realisiert werden. Dies bedeutet, dass der Patient nicht mehr rigid fixiert werden muss [5].

Schlussfolgerungen und Ausblick

Die aufgebauten Prototyp-Systeme haben gezeigt, dass die selbst entwickelte Steuerung bis auf die in einem roboterabhängigen Teil zusammengefassten Funktionen ohne Änderung an verschiedene Roboterarme angepasst werden kann. Ein Schwerpunkt unserer zukünftigen Arbeiten wird darin liegen, die Anwendungsmöglichkeiten flexibel gesteuerter Assistenzroboter in verschiedenen chirurgischen Applikationen zu erproben. Dazu müssen vor allem entsprechende Komponenten für die Planungs- und Steuerungssoftware in Layer 4 entwickelt werden, die dann über die HNI-Schnittstelle den Roboter steuern.

Literatur

- [1] Schlimbach, M.; Wahrburg, J.: Netzwerkschnittstelle zur Steuerung eines navigierten Chirurgieroboters. Beitrag eingereicht zur CURAC-Tagung, Düsseldorf, Nov. 2010
- [2] Knappe, P.; Pieck, S.; Wahrburg, J.: Komponenten und Architektur eines navigierten Assistenzroboters für chirurgische Anwendungen. In: *at – Automatisierungstechnik* 53 (2005), Heft 12, S. 615-626
- [3] Castillo Cruces, R.A.; Wahrburg, J.: Improving robot arm control for safe and robust haptic cooperation in orthopaedic procedures. In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 3 (2007), pp. 316-322
- [4] Castillo Cruces, R.A.; Wahrburg, J.: Virtual Fixtures with Autonomous Error Compensation for Human-Robot Cooperative Tasks. In: *Robotica*, vol. 28, part 2, March 2010, pp. 267-277
- [5] Schneider, H.C.; Wahrburg, J.: Neuer Echtzeitkern zur Verbesserung der Dynamik und Sicherheit eines navigierten Chirurgie-Assistenzroboters. In: *at – Automatisierungstechnik* 56 (2008), Heft 9, S. 483-493

Danksagung

Teile dieser Arbeit wurden im Rahmen des DFG-SPP 1124 und durch das BMBF-Projekt 01EZ0740 gefördert.