

Möglichkeiten der Kraftregelung an Leichtbaurobotern

Christian Thormann, Alexander Winkler

Hochschule Mittweida, Technikumplatz 17, 09648 Mittweida

In diesem Beitrag werden Möglichkeiten der Kraft-/ Momentregelung an Leichtbaurobotern aufgezeigt. Für die Versuche werden zwei Roboter verwendet, einer ist mit einem internen Kraft-/ Momentsensor ausgestattet, bei dem anderen Roboter werden die Motorströme der Antriebe zur Ermittlung der Kontaktkräfte und -momente benutzt. Bei den meisten kommerziellen Systemen ist es schwierig, Kraftregelalgorithmen zu implementieren. Häufig sind die Programmierfunktionalitäten sehr eingeschränkt. Bei den hier verwendeten Systemen sind Algorithmen zur Kraftregelung vorprogrammiert. Diese sollen getestet und bewertet werden. Dazu soll der Roboter zunächst den Kontakt zu einer Oberfläche herstellen. Danach wird die Kontaktkraft sprungförmig geändert. Zudem soll eine komplexe Aufgabe zur Kraftregelung implementiert werden.

1. Einleitung

In Situationen, bei denen der Roboter physischen Kontakt zu seiner Umwelt hat, ist es sinnvoll, die Roboterbewegung durch die Kontaktkräfte zwischen Roboter und Umwelt zu steuern. Diese können mit einem Kraft-/ Momentsensor (KMS) gemessen werden. Mittels dieser Messwerte kann dann die Roboterbewegung beeinflusst werden. Dieser Vorgang wird als Kraft-/ Momentregelung bezeichnet. Der Sensor ist hierbei üblicherweise am Roboterflansch montiert.

Die Kraft-/ Momentregelung wird seit vielen Jahren bzw. Jahrzehnten erforscht. Eine schöne Übersicht über wichtige Konzepte findet man in [1]. Seit einigen Jahren geht der Trend hin zu sogenannten Leichtbaurobotern. Bei denen ist das Verhältnis zwischen Nutzlast und Eigengewicht näher an 1:1 ist als 1:10, was bei konventionellen Industrierobotern der Fall ist. Leichtbauroboter wurden ursprünglich für die Raumfahrt entwickelt. Die geringe Eigenmasse führte zu neuen Einsatzgebieten wie z.B. der Mensch-Roboter Interaktion, welche einen wachsenden Trend darstellt [2]. Daher findet man heutzutage zahlreiche Anbieter von Leichtbaurobotern.

Einer dieser Anbieter ist Universal Robots, deren UR-Serie sehr verbreitet ist. Für die Teach-In Programmierung kann der Roboter handgeführt werden. Dazu werden die Motorströme analysiert, ein Kraftsensor ist nicht nötig. Das Handführen entspricht einer sensorlosen Kraftregelung. Damit könnten auch kraftgeregelte Aufgaben, wie z.B. Oberflächenbearbeitung, ohne einen KMS durchgeführt werden. Zudem bietet UR eine erweiterte Baureihe an (URe), welche mit einem KMS im Flansch ausgestattet ist. Dieser ist in der Robotersteuerung integriert.

In diesem Beitrag sollen die Möglichkeiten der Kraft-/ Momentregelungen der Systeme UR und URe miteinander verglichen werden. Die Qualität der Kraftregelung wird anhand der Kraftverläufe bewertet, die beim Kontaktieren der Umgebung durch den Endeffektor auftreten. Des Weiteren werden Sprungantworten bei bestehendem Kontakt ausgewertet. Zur Demonstration der

Kraftregelung wird außerdem eine komplexe und anschauliche Aufgabe implementiert. Der somit entstandene Demonstrator kann z.B. verwendet werden, um Schülerinnen und Schülern die Robotik näher zu bringen.

Dieser Beitrag ist wie folgt gegliedert: Im nächsten Abschnitt wird der Versuchsaufbau beschrieben. Abschnitt 3 erläutert die Implementierung von Kraftregelalgorithmen auf der UR-Robotersteuerung. Dazu werden die notwendigen Funktionen der UR-Script Programmierung [3] vorgestellt. Anschließend werden im Abschnitt 4 die Versuche beschrieben und ausgewertet, um die Qualität der Kraftregelung der beiden Roboter zu vergleichen. Im Abschnitt 5 wird die Implementierung einer komplexen Aufgabe zur Kraft-/ Momentregelung präsentiert. Der Roboter soll hierbei kraftgeregelt ein Bild zeichnen, welches als Datei vorgegeben ist. Im letzten Abschnitt erfolgt eine Zusammenfassung.

2. Versuchsaufbau

Für die Versuche wurden zwei Roboter, UR3 und UR3e, verwendet. Beide sind 6-achsige Gelenkarmroboter mit einer Nutzlast von 3 kg. Sie sind auch im Rahmen der Mensch-Roboter Interaktionen einsetzbar. Dazu verfügen die Roboter über Leistungs- und Kraftbegrenzungen, wodurch sie auch ohne Schutzzaun betrieben werden können. Wie bereits erwähnt, ist die e-Serie mit einem integrierten KMS ausgestattet. Dessen Messbereich beträgt ± 30 N für Kräfte und ± 10 Nm für Momente.

Im ersten Versuch sollen die Roboter daraufhin verglichen werden, wie sie im Modus der Kraftregelung reagieren, wenn sie aus einer Bewegung heraus auf die Kontaktumgebung treffen. Dazu werden die Kontaktkräfte durch einen am Flansch montierter externer KMS aufgezeichnet, da die Messwerte der internen Sensoren von der Robotersteuerung nicht in Echtzeit bereitgestellt werden. Der Sensor KMS40 der Firma Weiss Robotics besitzt einen Messbereich von ± 120 N und ± 3 Nm für die Kräfte bzw. die Momente. Die Datenübertragung zur Robotersteuerung oder an einen PC erfolgt via Ethernet.

In der Abbildung 1 sind zwei Szenarien zum Herstellen des Kontaktes zwischen Roboter und Umgebung dargestellt. Das erste mit einer steifen, das zweite mit einer nachgiebigen Oberfläche. Der externe KMS ist am Flansch des UR3e zu erkennen. Der Aufbau für den UR3 war identisch.

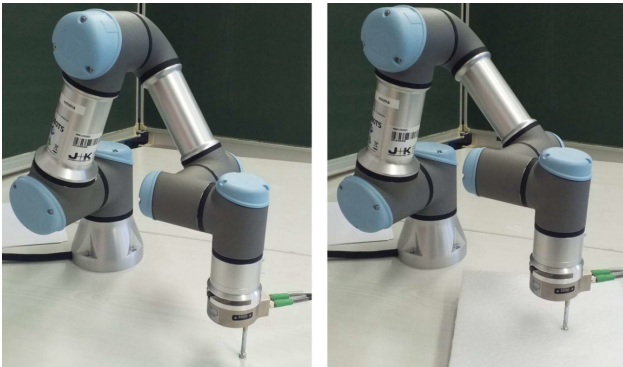


Abbildung 1: Steife und nachgiebige Kontaktumgebung

3. Implementierung der Kraftregelung in die UR-Leichtbauroboter

Die Robotersteuerungen unterstützen Multi-Threading. Die Kraftregelung muss parallel zum Hauptprogramm laufen, welches die Bewegungsbefehle beinhaltet. Die Roboterbewegung kann Achsinterpoliert *movej*, Linearinterpoliert *move* oder Kreisinterpoliert *movec* sein. Ein neuer Thread wird mit dem Befehl *thread* deklariert. Es ist notwendig, die *sync*-Anweisung in jedem Thread aufzurufen, damit andere Threads auch ausgeführt werden können. Für die Implementierung der Kraft-/ Momentregelung muss der Befehl *force_mode* (*Feature, Compliance, Force, Type, Limits*) in UR-Script verwendet werden [3]. Die Parameter haben folgende Bedeutung:

- *Feature*: Spezifiziert das Koordinatensystem, in welchem die Regelung arbeiten soll. Hier kann z.B. das Werkzeugkoordinatensystem ausgewählt werden.
- *Compliance*: Aktiviert die kartesischen Freiheitsgrade für die Kraft-/ Moment-Regelung.
- *Force*: Dieser Vektor beinhaltet die Sollwerte der Kräfte und Momente in N bzw. Nm.
- *Type*: Definiert die Transformation. Der Wert 2 bedeutet, dass keine Transformation stattfindet.
- *Limits*: Hier werden die maximale Geschwindigkeiten eingestellt, die der Roboter in der entsprechenden Freiheitsgraden verfahren darf. Für Positionsgeregelte Freiheitsgrade stellen diese Werte maximale Weg-/ Winkelabweichungen dar.

Nach der Deklaration des Threads zur Kraftregelung muss der Sensor tariert werden. Hierfür wird die Funktion *zero_ftsensor()* verwendet. Danach wird eine Wartezeit von mindestens 20 ms empfohlen. Nachdem der Roboter an den Startpunkt bewegt wurde, kann die Kraftregelung mittels der Funktion *run* aktiviert werden. An-

schließend folgt eine Wartezeit von 500 ms. Nach der Aktivierung des Threads werden alle Bewegungsbefehle mit Kraft-/ Momentregelung durchgeführt. Die Regelung arbeitet nur wenn Bewegungsbefehle aufgerufen werden. Ist keine Bewegung gewünscht müssen Bewegungsbefehle mit sehr kleinen Distanzen programmiert werden. Die Befehle *kill* und *end_force_mode* beenden den Regelungs-Thread und stoppen den Sensor. Die Abbildung 2 zeigt beispielhaft ein Programm in UR-Script, in welchem die verschiedenen Funktionen verwendet werden.

```
thread MyForceCtrlThread() :
    while(True) :
        force_mode(get_actual_tcp_pose(),
                    [0,0,1,0,0,0],
                    [0,0,1.5,0,0,0],2,
                    [0.1,0.1,0.1,0.35,0.35,0.35])

    sync()
end
end

zero_ftsensor()
sleep(0.2)
move ...

global MyForceCtrl=run MyForceCtrlThread()
move ...
...
kill MyForceCtrl
end_force_mode()
move ...
```

Abbildung 2: Beispielprogramm zur Kraft-/ Momentregelung

4. Versuche und Resultate

In diesem Abschnitt werden die Potentiale der beiden Roboter hinsichtlich der Kraft-/ Momentregelung untersucht und ausgewertet. Dazu werden Experimente zur Kontaktierung aus dem freien Raum heraus und Sollwertänderungen bei bestehenden Kontakt zwischen Endeffektor und Umgebung durchgeführt.

4.1. Kraftgeregeltes Kontaktieren

Im ersten Versuch soll das Verhalten beim Kontaktieren verglichen werden. Hierfür wurde das Werkzeug des Roboters in einem geringen Abstand zur Oberfläche platziert. Die Kraftregelung wurde so aktiviert, dass in z-Richtung des Werkzeugs ein Kräfte Sollwert F_d von 5 N vorgegeben wurde. Die Kontaktkräfte wurden mit dem externen KMS gemessen und aufgezeichnet. Die Abbildung 3 zeigt den Zeitverlauf der Kräfte beim Kontaktieren der harten und der weichen Oberfläche.

Zunächst kann beim Kontaktieren ein großes Überspringen beobachtet werden. Im Vergleich der Robotersysteme fällt auf, dass das Überspringen am UR3e größer ist. Der Grund dafür liegt wahrscheinlich daran,

dass der Kraftregelkreis um den Positionsregelkreis herumgeschlossen wurde. Diese Struktur wird auch als positions-basierte Kraftregelung bezeichnet. Diese Regelstruktur besitzt eine schlechte Dynamik und es können große Kraftspitzen auftreten.

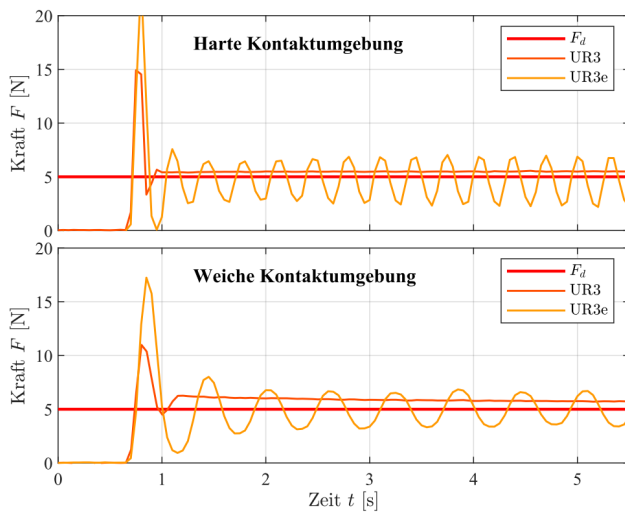


Abbildung 3: Kräfte während des Kontaktierens

Beim UR3 wurde der Kraftregelkreis über die Gelenkmomente bzw. über die Motorenströme realisiert. Daher ist die Kraftregelung am UR3 dynamischer als die am UR3e. Nachdem der UR3e-Roboter die Oberfläche berührt hat, beginnt die Kraft zu schwingen. Die Amplitude beträgt etwa 2 N. Der Grund für die Schwingung liegt wahrscheinlich an der Zeitverzögerung in der positions-basierten Kraftregelung. Je nach Oberfläche und Nachgiebigkeit können unterschiedliche Kraftspitzen während des Kontaktierens beobachtet werden. Eine harte Oberfläche erzeugt eine höhere Kraftspitze. Zudem beeinflusst die Nachgiebigkeit die Periodendauer der Schwingung. Die Schwingungsfrequenz beträgt etwa 3,0 Hz für die harte und 1,6 Hz für die weiche Oberfläche. Der UR3 schwingt nicht, wenn er die Oberfläche berührt. Aufgrund des fehlenden internen Kraft-/ Momentsensors wurde jedoch eine statische Abweichung der Kontaktkraft gemessen.

4.2. Sollwertänderung in der Kraftregelung

Im zweiten Experiment wurden die Sprungantworten der Kraftregelkreise für beide Roboter aufgezeichnet. Die Ergebnisse sind in der Abbildung 4 dargestellt. Zum Beginn des Experimentes berührt das am Roboterflansch befestigte Werkzeug die Oberfläche. Die Sollkraft F_d ist auf 5 N eingestellt. Nach einigen Sekunden wird F_d auf 10 N gesetzt, nach einer weiteren Wartezeit dann auf 20 N. Es wurde festgestellt, dass die Kraftregelung am UR3 recht schlecht funktioniert. Während der Änderung von F_d von 5 N auf 10 N konnte keine Reaktion festgestellt werden. Nach der zweiten Sollwertänderung wird die Kraft am UR3 vergrößert. Eine Abweichung vom Sollwert um ca. 40% kann beobachtet werden. Die Kraftregelung am UR3 kann daher nicht für Aufgaben verwen-

det werden, bei denen eine exakte Kontaktkraft notwendig ist. Die Kraftregelung am UR3e ist im Gegensatz zum UR3 genauer und weist eine bessere Qualität auf. Die Kontaktkraft folgt dem Sollwert. Die im vorherigen Experiment beobachteten Schwingungen sind immer noch vorhanden. Im Fall von $F_d = 20$ N ist die Amplitude bei der steiferen Umgebung etwas geringer.

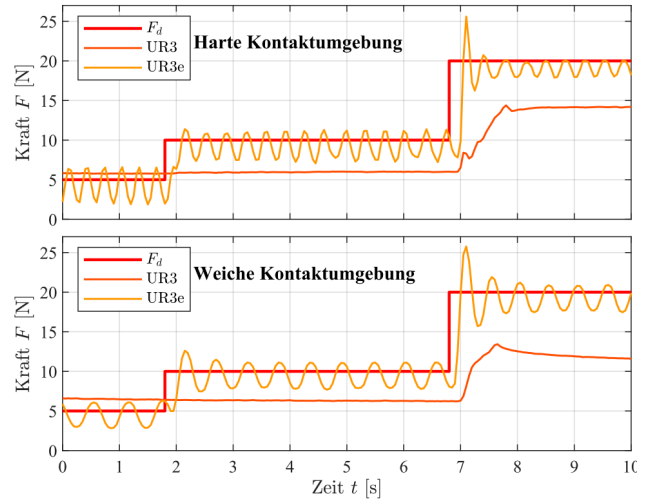


Abbildung 4: Sprungantworten der geschlossenen Regelkreise

5. Verwendung der Kraftregelung am Beispiel einer komplexen Aufgabe

Die Kraftregelung wurde mittels einfacher Experimente bewertet. Nun soll eine komplexere Aufgabe mit dem Leichtbauroboter durchgeführt werden. Beispiele für Aufgaben, welche kraftgeregelt bearbeitet werden können, sind z.B. Montage, Oberflächenbearbeitung oder Mensch-Roboter Interaktion. Ein Beispiel der Oberflächenbearbeitung ist das Zeichnen. Diese Aufgabe kann auch in einem Robotik-Labor implementiert werden.

In unserem Szenario soll der Roboter ein Bild auf ein Blatt Papier, welches auf dem Tisch liegt, zeichnen. Hierfür wurde der UR3e gewählt, da dieser über einen integrierten KMS verfügt, was günstiger für die Kraftregelung scheint. Zum Zeichnen wird ein Bleistift verwendet, welcher am Flansch befestigt wurde. Die Bildquelle kann entweder eine Kamera oder eine Bilddatei sein. Zudem wird der Stift gelegentlich kraftgeregelt angespitzt. Die Implementierung besteht aus zwei Teilen.

Zunächst läuft eine Applikation auf einem PC. In dieser werden die Bildverarbeitung und die Bahnberechnung durchgeführt. Die Bewegungsdaten werden an die Robotersteuerung gesendet, welche die Bewegungssteuerung und die Kraftregelung durchführt. Die PC-Applikation wurde in MATLAB erstellt und beinhaltet folgende Funktionalitäten, die mittels einer entsprechenden Bedienoberfläche gesteuert werden können:

- Aufnahme eines Kamerabildes oder laden einer Bilddatei
- Manuelles zuschneiden des Bildes

- Automatische Reduzierung der Auflösung
- Umwandlung in ein Binärbild
- Berechnung der Roboterbahn

Der zweite Programmteil, der auf der Robotersteuerung abgearbeitet wird, empfängt Positionsdaten vom PC und aktiviert die Roboterbewegung in Form einer Linearinterpolation. Parallel hierzu wird die Kraftregelung für die z-Achse des Werkzeugkoordinatensystems aktiviert. Als Sollkraft wurde 1 N gewählt. Diese Einstellung kann auch durch die Vektor $\mathbf{F}_d$ ausgedrückt werden, der die Sollwerte der Kräfte bzw. Momente enthält.

$$\mathbf{F}_d = [- \quad - \quad 1,0 \text{ N} \quad - \quad - \quad -]^T$$

Die Einstellungen können in der *force_mode*-Anweisung von UR-Script geändert werden, die im Abschnitt 3 bereits beschrieben wurde. Die Kraftregelung wird parallel zur Bewegungssteuerung abgearbeitet, welche aus dem Bild, das gezeichnet werden soll, generiert wird. In Abbildung 5 (links) ist zu erkennen, dass das kraftgeregelter Zeichner gut funktioniert. Das Bild wurde durch den Roboter gezeichnet ohne den Stift zu beschädigen.

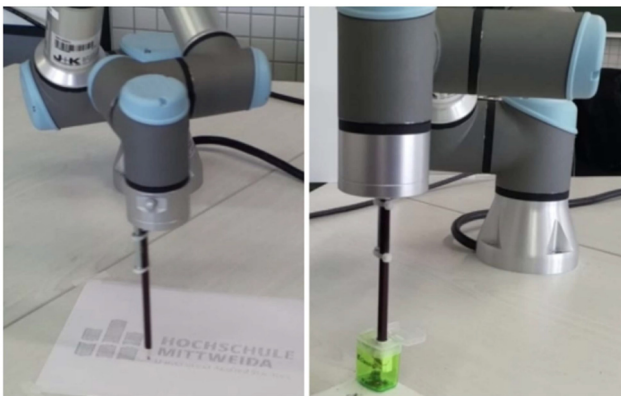


Abbildung 5: Versuchsaufbau für kraftgeregelter Zeichner und Anspitzen des Stiftes

Das rechte Bild in Abbildung 5 zeigt das Anspitzen des Stiftes. Dieses Programm besteht aus zwei Teilen. Zuerst wird der Stift in den Spitzer eingeführt. Hierfür wurde Kraftregelung mittels $\mathbf{F}_{d1}$ wie folgt parametrisiert:

$$\mathbf{F}_{d1} = [0 \quad 0 \quad 1,0 \text{ N} \quad 0 \quad 0 \quad -]^T$$

Im Gegensatz zum Programm des Zeichners wird hier keine Bewegungssteuerung benötigt. Die Kraftregelung wird allerdings nur parallel zu Bewegungsbefehlen abgearbeitet. Daher wurde eine kleine Bewegung von 1 mm programmiert. Typischerweise würde eine solche Bewegung sehr schnell gehen. Die *move*-Anweisung in UR-Script jedoch kann so parametrisiert werden, dass die Dauer der Bewegung eingestellt werden kann. In diesem Fall wurde eine Dauer von 5 s eingestellt, innerhalb der Roboter den Stift in den Spitzer drückt.

Im zweiten Schritt wird das Spitzen durchgeführt. Während der Kraft-/ Momentregelung werden die Freiheitsgrade F_z , M_x und M_y aktiviert, dargestellt in $\mathbf{F}_{d2}$:

$$\mathbf{F}_{d2} = [- \quad - \quad 5,0 \text{ N} \quad 0 \quad 0 \quad -]^T$$

Parallel zur Kraftregelung wird der Stift um die z-Achse gedreht. Die Drehung wurde mittels Linearinterpolation realisiert. Gedreht wurde 10-mal um 90°. Es wurde festgestellt, dass das kraftgeregelter Anspitzen funktionierte.

6. Zusammenfassung

In diesem Beitrag wurden zwei weit verbreitete Leichtbauroboter der Typen UR3 und der UR3e von Universal Robots bezüglich der Kraftregelung untersucht. Der UR3e besitzt einen integrierten KMS. Die Programmierung mittels UR-Script war für beide Roboter identisch.

Es wurden einige Versuche durchgeführt, in welchem ein Freiheitsgrad, die z-Achse des Werkzeugkoordinatensystems, kraftgeregelter arbeitete. Als erstes kontaktierte der Roboter die Tischfläche. Danach wurde die Kontaktkraft geändert. Das Experiment wurde auf einer weichen und einer harten Kontaktumgebung durchgeführt. Die Kontaktkräfte wurden durch einen externen KMS gemessen. Der UR3e war in der Lage die Kontaktkraft zufriedenstellend zu regeln. Eine genaue Kraftregelung war beim UR3 nicht möglich, da die Regelkreise über die Gelenkmomente bzw. die Motorströme geschlossen wurden. Beim UR3e war ein starkes Schwingen der Kontaktkraft zu beobachten, was nachteilig ist. Beim UR3 konnte keine Schwingung beobachtet werden. Aus diesem Grund kann die Kraftregelung mit dem UR3 für Anwendungen verwendet werden, bei denen ein stabiler Kontakt zwischen dem Endeffektor und der Umgebung erforderlich ist und der tatsächliche Wert der Kontaktkraft nicht so wichtig ist. Aufgrund der verschiedenen Steuerungsstrukturen kann der Schluss gezogen werden, dass der UR3 nachgiebiger ist als der UR3e. Am Ende wurde das Zeichnen eine Figur durch einen Roboter ermöglicht, was eine anspruchsvolle Aufgabe ist. Trotz des relativ ungünstigen Verhaltens der Kraftregelung konnte gezeigt werden, dass es möglich ist die Zeichenaufgabe erfolgreich umzusetzen. Zudem wurde das Anspitzen des Bleistiftes durch den Roboter demonstriert.

Literaturverzeichnis

- [1] G. Zeng and A. Hemami, *An overview of robot force control*, Robotica, vol. 15, no. 5, pp. 473–482, 1997.
- [2] B. Finkemeyer, *Towards safe human-robot collaboration*, in Proc. of the IEEE Int. Conf. on Methods and Models in Automation and Robotics (MMAR), 2017, pp. 883–888.
- [3] Universal Robots A/S, *The URScript Programming Language*, 2018.

Danksagung

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