Demonstrationsmodell für parallelkinematische ebene Manipulatoren mit Lego-Mindstorms

Prof. Dr. Michael Beitelschmidt

TU Dresden, Professur für Dynamik und Mechanismentechnik
michael.beitelschmidt@tu-dresden.de

Kurzfassung

Parallelkinematische Manipulatoren vereinigen Fragestellungen der Robotik mit denen der klassischen Getriebelehre. Ein auch technisch einfach zu realisierender parallelkinematischer Manipulator ist das ebene Fünfgelenk. Dieser kann aus Lego-Elementen gebaut werden. Zum Antrieb sowie der intelligenten Steuerung wird das Lego-Mindstorms System verwendet. Bringt man im Tool-Center-Point einen Stift an, kann der Manipulator als Plotter benutzt werden. Eine in der Sprache NXT-G speziell programmierte Software auf dem Mindstorms-NXT ist in der Lage, Steuerbefehle, die in einer Textdatei codiert sind, auszuführen, und damit den Manipulator zu bewegen. Damit kann der Manipulator in der Lehre als Demonstrationsgerät eingesetzt werden. Studierende können anhand von Zeichenaufgaben die inverse Kinematik eines parallelkinematischen Manipulators erlernen.

Abstract

Parallel kinematic manipulators unite issues of robotics with the classical theory of mechanisms. A parallel kinematic manipulator also technically easy to realize is the five joint mechanism. This can be built out of LEGO elements. The LEGO MINDSTORMS system can be used for the drive and intelligent control. Placing a pen in the tool center point, the manipulator can be used as a plotter. A specially programmed software in the

DOI: 10.14459/2015md1276148

language of NXT-G on the MINDSTORMS NXT is able to perform commands, which are coded in a text file, and therewith move the manipulator. Thus, the manipulator can be used in teaching as a demonstration device. Students can learn the inverse kinematics of parallel kinematic manipulator on drawing tasks.

1 Einführung

Eine technische Gerätefamilie, in der klassische Getriebelehre bzw. Mechanismentechnik und Robotik zusammen kommen, sind parallelkinematische Manipulatoren. Sie besitzen, im Gegensatz zu Robotern mit serieller Kinematik, kinematische Schleifen, passive Gelenke sowie deutlich mehr Körper und Gelenke als Bewegungsfreiheitsgrade. Die Berechnung der Kinematik ist somit sehr verwandt zu Aufgaben in der Mechanismentechnik.

Bei Manipulatoren mit serieller Kinematik ist es in der Regel einfach und auch eindeutig, aus gegebenen Gelenkkoordinaten $\boldsymbol{q}$, darin können Verdrehungen und Verschiebungen enthalten sein, die Pose $\boldsymbol{x}$ des Tool Center Points (TCP) zu berechnen. Die Pose $\boldsymbol{x}$ enthält dabei die kartesischen Koordinaten im Arbeitsraum sowie eine geeignete Drehbeschreibung. Diese Berechnung kann bei parallelkinematischen Systemen sehr komplex und häufig gar nicht mehr analytisch möglich sein. Das liegt daran, dass in einem parallelkinematischen System eine einzelne Gelenkauslenkung q_i unter Umständen den gesamten Mechanismus in Bewegung setzten kann, während in einem seriellen System immer nur die in der kinematischen Kette folgenden Glieder bewegt werden. Dafür kann bei einem parallelkinematischen System unter Umständen die inverse Kinematik, d.h. die Berechnung der Gelenkkoordinaten $\boldsymbol{q}$ aus einer gegebenen Pose $\boldsymbol{x}$ einfach zu berechnen sein.

Ein auch technisch einfach zu realisierender parallelkinematischer Manipulator ist das ebene Fünfgelenk. Dieser kann aus Lego-Elementen gebaut werden. Zum Antrieb sowie der intelligenten Steuerung wird das Lego-Mindstorms System verwendet. Wird im Tool-Center-Point (TCP) ein Stift befestigt, kann der Manipulator als Plotter eingesetzt werden und somit die Zusammenhänge zwischen Motordrehungen und kartesicher Bewegung des TCP „zu Papier gebracht werden“. Damit steht ein Demonstratiosmodell zur Verfügung, das in der Lehre eingesetzt werden

kann um Begriffe wie Vorwärtskinematik und inverse Kinematik eines parallelkinematischen Systems Studierenden vorzuführen. Zudem können Studierende selbst Steuerprogramme für den Manipulator erstellen und damit die programmtechnische Umsetzung der inversen Kinematik erlernen.

2 Das ebene Fünfgelenk

Das ebene Fünfgelenk mit zwei angetriebenen Achsen ist in Abbildung 1 dargestellt. Es besteht aus den zwei Kurbeln gleicher Länge l_K, die in den Punkten B und C drehbar gelagert sind. Dort befinden sich auch die beiden Antriebsmotoren, deren aktuelle Stellung mit den Winkeln φ_B und φ_C beschrieben wird. In den Endpunkten der Kurbeln B‘ und C‘ sind die beiden Hebel befestigt, die ihrerseits im Punkt G miteinander verbunden sind. Die Hebellänge beträgt l_H. Aus konstruktiven Gründen liegt der TCP nicht im Gelenkpunkt der Hebel, sondern in Verlängerung der Hebelachse C‘-G mit dem Abstand l_S vom Punkt C‘. Damit ist der Mechanismus unsymmetrisch, was die Berechnung etwas aufwändiger macht. In der Mitte zwischen den Punkten B und C ist das kartesische Koordinatensystem lokalisiert, von dem aus die x- und y-Koordinate des TCP gemessen wird.

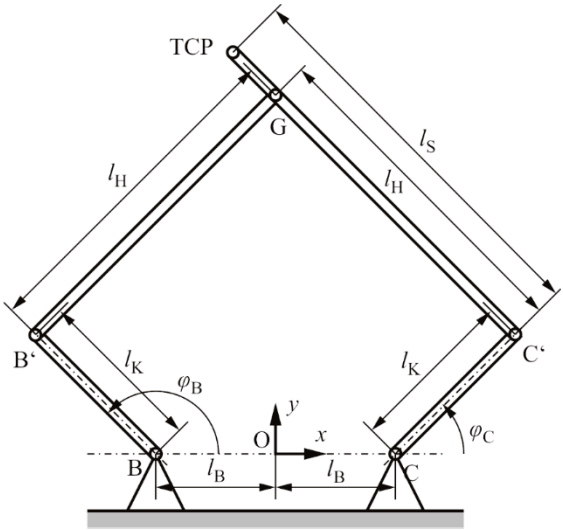

Abbildung 1: Das ebene Fünfgelenk mit Geometriegrößen

Die Anwendung der Grübler Gleichung für 4 Körper und 5 zweiwertige Gelenke in der Ebene lautet $f = 4 \cdot 3 - 5 \cdot 2 = 2$, die zwei verbliebenen Freiheitsgrade können durch die Antriebsmotoren, die in diesem Sinne zwei rheonome Bindungen darstellen, vollständig kontrolliert werden. Da bei einem Plotterstift die Verdrehung keine Rolle spielt, genügt es, die TCP-Bewegung durch seine x- und y-Koordinate zu beschreiben. Somit stehen den zwei Konfigurationsraumkoordinaten φ_B und φ_C zwei Arbeitsraumkoordinaten x und y gegenüber.

Die Abmessungen des Plotters sind so gewählt, dass es eine Grundstellung gibt, bei der die linke Kurbel bei 45° stehend, die rechte Kurbel bei -45° stehend mit den beiden Hebeln in den Punkten B‘ und C‘ jeweils einen rechten Winkel bilden und die Hebel untereinander im Koppelpunkt G ebenfalls einen rechten Winkel bilden. In Tabelle 1 sind die dazu erforderlichen Längen zusammengestellt. Da bei einer Konstruktion in Lego nur Längenschritte im 8 mm Raster möglich sind, lässt sich die ideale Grundstellung nicht erreichen. Für die gewählten Längen beträgt der Schnittwinkel der Hebel im Koppelpunkt bei $\varphi_B = 45°$ und $\varphi_C = -45°$ 88,45° und der Winkel zwischen Kurbeln und Hebeln 90,77°.

Tabelle 1: Geometrische Größen des Manipulators

Name des Elements	FZ	Länge /Lego-BM	Länge /mm
Kurbel	l_K	7	56
Basis (halbe Länge)	l_B	9	72
Hebel	l_H	20	160
Stifthebel	l_S	23	184

Die inverse Kinematik des Mechanismus ist in elementaren Schritten durchführbar, aber nicht in einer kompakten Formel darstellbar. Sie soll hier nicht formelmäßig dargestellt werden, es müssen folgende Schritte abgearbeitet werden:

1. Berechnung des Winkels φ_C aus der Umrechnung des Ortsvektors vom Punkt C zum TCP in das Dreieck C-C‘-TCP. Dabei sind immer zwei Lösungen möglich, die sich durch das Vorzeichen des Knickwinkels am Punkt C‘ unterscheiden. Hier wird immer die Lösung mit dem Knick nach rechts außen verwendet.

2. Berechnung der Lage des Punktes G auf der Verbindungsstrecke C‘-TCP.
3. Berechnung des Winkels φ_B sinngemäß zur Rechnung unter Punkt 1 für das Dreieck B-B‘-G. Dabei wird für den Knickwinkel im Punkt B‘ der Fall mit Knick nach links außen verwendet.

Die inverse Kinematik liefert somit theoretisch für eine TCP-Position vier verschiedene Stellungen im Konfigurationsraum. Die Vorwärtskinematik soll hier ebenfalls nicht detailliert ausgeführt werden, sondern nur die erforderlichen Schritte:

1. Berechnung der Ortsvektoren zu den Punkten B‘ und C‘
2. Bestimmung der Verbindungsstrecke B’-C‘
3. Bestimmung des Abstandes von G zum Mittelpunkt der Strecke B‘-C‘. Aufgrund der Symmetrie liegt G auf dem Mittellot der Strecke B‘-C‘.
4. Aufstellen des Ortsvektors zum Punkt G. Hier sind theoretisch zwei Lösungen möglich: Der Punkt G liegt oberhalb der Strecke B‘-C‘ oder unterhalb. Die Lösung unterhalb ist aufgrund der Stifthalterung im TCP nicht erreichbar.
5. Verlängern der Strecke C‘-G zum TCP

Das ebene Viergelenk zeigt somit beide in der Mechanismentechnik möglichen Uneindeutigkeiten: Eine Arbeitsraumposition kann mit vier verschiedenen Gelenkstellungen erreicht werden und eine Gelenkstellung kann zu zwei Arbeitsraumpositionen führen.

Mit der vorgestellten Geometrie sind für die Kurbeln die Winkel $180° \leq \varphi_B \leq 90°$ und $90° \leq \varphi_C \leq 0°$ sinnvoll. In Abbildung 10 sind die Konfigurationsraumbegrenzung und ihre Abbildung in den kartesischen Arbeitsraum dargestellt.

3 Realisation mit Lego Mindstorms

Der Plotter basiert wesentlich auf der Konstruktion von O. Holowenko vom Institut für Werkzeugmaschinen und Steuerungstechnik der TU Dresden. Lego Mindstorms bietet Antriebsmotoren, die Bewegungen mit einer Auflösung von 1° ermöglichen. Zwischen Motorachse und Kurbe-

lachse ist ein Schneckengetriebe mit einer Übersetzung von 1:40 eingebaut. Dies ermöglicht theoretisch eine Bewegung der Kurbeln mit einer Genauigkeit von 1/40°.

Die Drehwinkelmessung der Motoren geschieht inkremental. Bei jedem Programmstart ist somit die Winkelstellung der Motoren unbekannt. In der Mitte des Manipulators, etwa beim Punkt O des Koordinatensystems sind zwei Taster angebracht, die eine Autokalibration ermöglichen. Bei der Autokalibration wird der Motor C solange mit positiver Winkelgeschwindigkeit bewegt, bis er bei $\varphi_C = 180°$, was außerhalb des normalerweise angefahrenen Winkelbereichs liegt, den Taster berührt. Damit kann der Nullpunkt von φ_C richtig festgelegt werden. Sinngemäß wird Motor B bei $\varphi_B = 0°$ bei Tastenanschlag festgelegt. Der dritte Motor dient dazu, den Plotter insgesamt zu kippen, um ein Anheben und Absetzten des Stifts zu ermöglichen. Die Konstruktion ist in Abbildung 2 dargestellt. Um Spielfreiheit der Gelenke und des Schneckengetriebes sicherzustellen, wird der Mechanismus mit zwei Gummibändern vorgespannt.

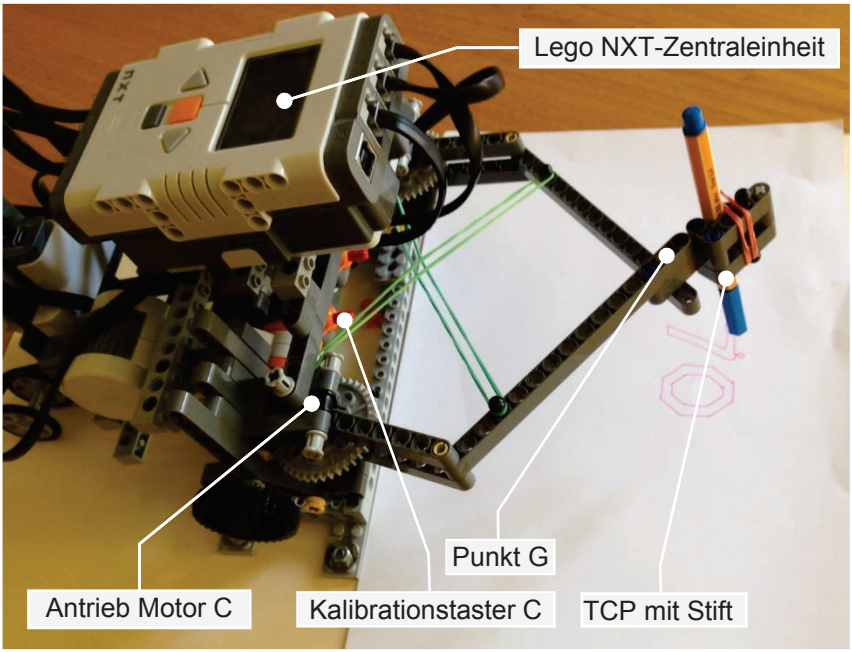

Abbildung 2: Foto des Manipulators

4 Software auf dem NXT

Um den Manipulator als Plotter nutzen zu können, muss auf dem Lego NXT-Zentralbaustein entsprechende Software implementiert werden. Der NXT-Baustein ist technisch gesehen ein Mikrocontroller auf Basis eines Atmel ARM-Prozessors, der bis zu drei Motoren steuern kann und bis zu 7 Sensorsignale, davon drei Motordrehwinkel, aufnehmen kann. Benutzerinteraktion ist zusätzlich durch das monochrome Display, Tonsignale sowie Tasten möglich. Kommunikation zu anderen Geräten, wie z.B. einem PC, wird über ein USB-Kabel oder Bluetooth realisiert.

Für die Programmierung des NXT stehen zahlreiche Sprachen zur Verfügung. Neben speziellen C- und Java-Dialekten gibt es eine Toolbox für Matlab und auch die von National Instruments entwickelte Sprache LabView stellt Sprachelemente für den NXT bereit. Lego selbst bietet die Sprache NXT-G an, die für die Software dieses Projekts verwendet wurde. NXT-G ist eine graphische Programmiersprache, die von National Instruments entwickelt wurde und wesentliche Paradigmen von LabView übernommen hat: Der Programmfluss ist signalbasiert, Blöcke führen Operationen aus, wenn alle Eingangssignale anliegen. Jede Operation des Programms wird in einem Block realisiert. Unterprogramme können gebildet werden, indem mehrere Blöcke zu einem Unterblock zusammengefasst werden, wobei alle über die Blockgrenze laufenden Signale neue Eingangs- und Ausgangssignale des Unterblocks werden. Die Grundausstattung von NXT-G umfasst Blöcke mit folgenden Funktionen:

- Steuern der Motoren mit vielfältigen Bewegungsmöglichkeiten
- Auslesen der Sensoren
- Ausgabe von Bildschirminhalten und Tönen
- Mathematische Grundrechenarten einschließlich der Wurzelfunktion
- Logische Operationen
- Schreiben und Lesen von auf dem NXT gespeicherten Textdateien

Zudem sind die Programmelemente Schleife und Verzweigung vorhanden. Alle Operationsblöcke müssen entlang eines Ablauf-Trägers angeordnet werden. Durch Verzweigung des Trägers lässt sich das Programm in parallel abzuarbeitende Teile aufspalten.

Die benötigten trigonometrischen Operationen Sinus, Kosinus, Arkustangens sowie Arkuskosinus sind nicht im Lieferumfang enthalten. Es existieren jedoch freie Ergänzungen zu NXT-G in der Netzgemeinschaft, unter anderem ein Trigonometriepaket, das Sinus und Kosinus sowie Arkustangens-2 mit zwei Eingabeargumenten realisiert. Aus dem Arkustangens kann mit der Formel

$$\arccos x = 2 \arctan \sqrt{\frac{1-x}{1+x}} \tag{1}$$

der Arkuskosinus berechnet werden. Die Berechnung dieser Formel ist in Abbildung 3 exemplarisch dargestellt, um die Funktionsweise von Berechnungsblöcken in NXT-G zu illustrieren.

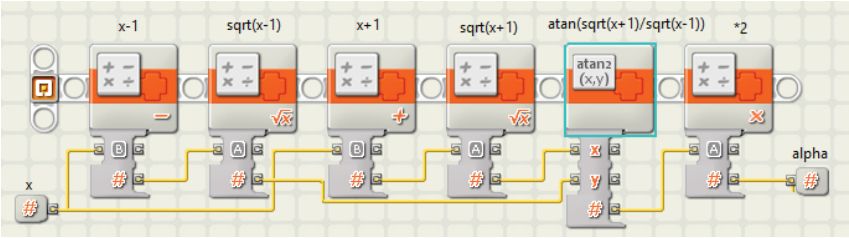

Abbildung 3: Berechnung des Arkuskosinus aus dem Arkustangens in NXT-G gemäß Gleichung (1)

4.1 Steuerungssprache

Um den Manipulator zu steuern, soll dieser in der Lage sein, eine in den Speicher des NXT geladene Textdatei, die einfache Kommandos in einer zu definierenden Steuerungssprache enthält, abzuarbeiten, und die entsprechenden Befehle auszuführen. Somit muss auf dem NXT ein Programm implementiert sein, das nach einer Initialisierungsphase, die Textdatei sukzessive einliest, die darin enthaltenen Befehle decodiert und die Kommandos ausführt. Da die in NXT-G vorhandenen Fähigkeiten zum Umgang mit Textdateien sehr begrenzt sind, ist die Struktur der Kommandos sehr einfach gehalten: Jedes besteht aus einem Großbuchstaben in einer Zeile, der das Kommando spezifiziert, und je nach Kommando bis zu zwei weiteren Zeilen mit Zahlenwerten, den Argumenten des Kommandos.

Es wurden insgesamt 16 Kommandos definiert, die sich in drei Gruppen aufteilen lassen:

- Allgemeine Kommandos, wie Heben und Senken des Plotterstifts, Zeichengeschwindigkeit, Wechsel der Kommandodatei, Programmende usw.
- Bewegungsbefehle im Konfigurationsraum: Bewegen der Kurbeln B und C um definierte Winkel einzeln und synchron. Die Bewegungen sind sowohl relativ als auch absolut möglich.
- Bewegungsbefehle im kartesischen Raum: Bewegen der Kurbeln B und C, um den Stift relativ um Δx und Δy weiter zu Bewegen oder zu einer absoluten Position x und y.

Die letztgenannten Kommandos setzen die Berechnung der inversen Kinematik auf dem NXT voraus. Ein Codeausschnitt mit einigen Kommandos ist in Abbildung 4 dargestellt.

```
%
Absolutbewegung_beide
A
0
500
%
Absolutbewegung_B
P
1000
%
Relativbewegung_B
B
-500
%
Absolutbewegung_C
K
-1000
```

Abbildung 4: Beispielausschnitt einer Steuerdatei.

4.2 Bewegungsbefehle im Konfigurationsraum

Die Bewegungsbefehle, sowohl im Konfigurationsraum als auch im kartesischen Raum, lassen sich immer auf eine relative Winkeländerung der

Motoren B und C zurückführen. Die Motorsteuerungsblöcke der NXT-G Sprache ermöglichen zunächst eine präzise Bewegung einer Achse.

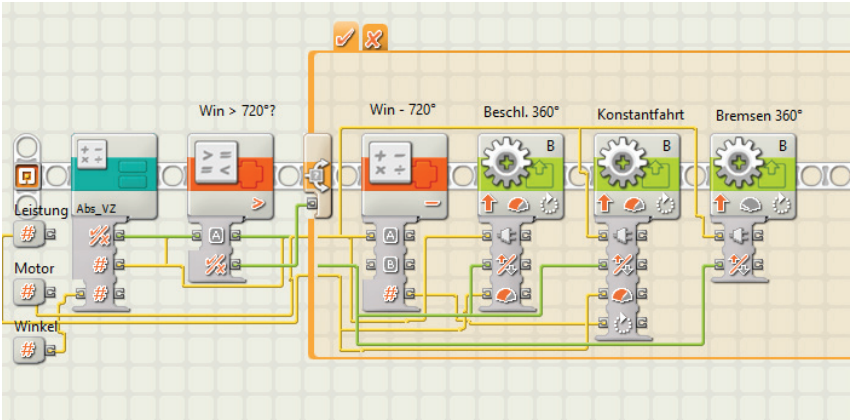

Abbildung 5: Programmausschnitt für die Bewegung eines Einzelmotors

In Abbildung 5 ist die Bewegungssteuerung in einem Ausschnitt aus einem NXT-G Programm zu sehen. Ist der Motor-Drehwinkel φ_{soll} größer als 720°, was einem Kurbelwinkel von 18° entspricht, kann eine voll ausgebildete Geschwindigkeitsrampe (Abbildung 6) gefahren werden: Innerhalb einer Umdrehung von 360° beschleunigt der Motor auf Sollgeschwindigkeit, geht dann in eine Konstantfahrt über, in welcher der Winkel $\varphi_{soll} - 720°$ zurückgelegt wird. Anschließend wird innerhalb von 360° auf Stillstand abgebremst. Der Fall $\varphi_{soll} < 720°$ wird durch das direkte Anstückeln von Beschleunigungs- und Bremsrampe mit verringerter Maximalgeschwindigkeit erreicht. Die Motor-Steuerungsblöcke veranlassen geregelte Bewegungen der Motoren, die Endwinkel werden gradgenau erreicht. Das zugrundeliegende Regelgesetz ist jedoch nicht transparent. Auch die sich genau einstellenden Rampenformen für Beschleunigung und Bremsen sind unbekannt.

Eine absolut synchrone Bewegung von zwei Motoren (Abbildung 6) ist mit den Mitteln von NXT-G nicht ohne weiteres möglich. Es existieren Blöcke, die ein synchrones Ansteuern für zweirädrige Roboter für Kurvenfahrten ermöglichen. Diese erwiesen sich jedoch für die hier geforderte Synchronität der Bewegungen als ungeeignet. Auch die Variante, die Motoren in zwei parallelen Prozessen anzusteuern, führt nicht zum Ziel, da die Bewegungen nicht ausreichend synchron sind. Ergebnis ist

jeweils eine Bewegung die nicht einer Geraden im Konfigurationsraum entspricht.

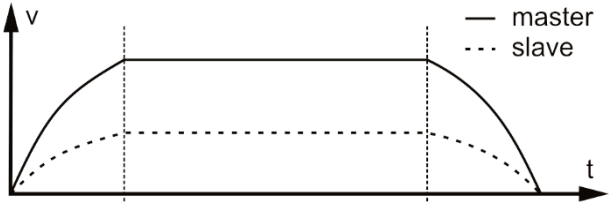

Abbildung 6: Synchrone Bewegung zweier Motoren mit untersch. Drehwinkeln

Für die synchrone Bewegung wurde deshalb ein anderer Ansatz gewählt. Die Achse mit der größeren Bewegung $\Delta\varphi$ wird zur Master-Achse erklärt. Sie fährt ihren Winkelerlauf mit dem oben erläuterten Bewegungsgesetz ab. Die Slave-Achse fährt der Master-Achse mit einer geregelten Bewegung hinterher. Für das Verhältnis der beiden Bewegungen wird das Verhältnis

$$k = \frac{\Delta\varphi_{Slave}}{\Delta\varphi_{Master}}, \quad -1 \leq k \leq 1 \tag{2}$$

gebildet. Damit kann aus dem aktuellen Drehwinkel des Master-Motors ein Sollsignal für den Slave-Motor gebildet werden. Die Folgeregelung (Abbildung 7) ist als reiner P-Regler ausgebildet. Die Verwendung eines I-Anteils zur Beseitigung einer statischen Regelabweichung, die sich als minimale Synchronabweichung bemerkbar macht, wäre wünschenswert, ist aber aufgrund der unbekannten Zykluszeiten der Regelschleife auf dem NXT nicht möglich.

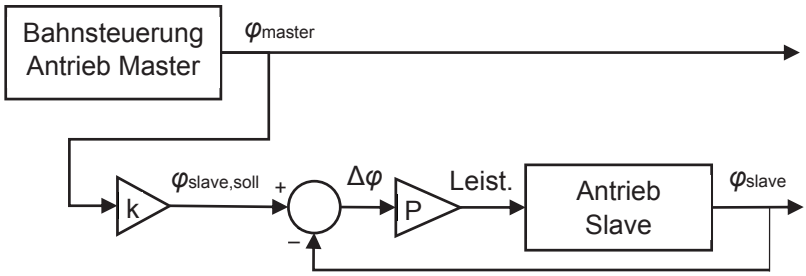

Abbildung 7: Master-Slave Regelungskonzept zur synchronen Bewegung

4.3 Bewegung im Arbeitsraum

Die Bewegung auf Basis der Koordinaten im kartesischen Arbeitsraum setzt auf der winkelgesteuerten Bewegung auf. Es werden die Start- und Zielkoordinaten mit der inversen Kinematik in Start- und Zielwinkel der beiden Kurbeln umgerechnet und eine synchrone Winkelbewegung dazwischen ausgeführt. Das hat zur Folge, dass eine Bewegung um Δx und Δy keine Gerade im Arbeitsraum darstellt. Es wird nur eine Gerade im Konfigurationsraum abgefahren, die sich im Arbeitsraum, je nach genauem Ort dort als leicht verzerrte Kurve abbildet.

Die für das Abfahren echter Arbeitsraumgeraden erforderliche Motorsteuerung, die eine kontinuierliche Berechnung der inversen Kinematik erfordern würde, sprengt die Fähigkeiten der NXT-G Programmiersprache sowie die Rechenleistung des NXT.

4.4 Parallele Dienstprozesse

Der NXT ermöglicht das parallele Bearbeiten mehrerer Tasks. So ist es möglich, neben dem Hauptprozess, der sequenziell die Kommandodatei abarbeitet, noch weitere Dienstprozesse laufen zu lassen:

- Anzeige der aktuellen Motorwinkel auf dem Display
- Anzeige der kartesischen Position im Arbeitsraum auf dem Display
- Überwachungsprozess mit Not-Stopp Funktion

In Abbildung 8 ist der Kontrollfluss der parallel abgearbeiteten Programmteile dargestellt. Der Prozess für die Anzeige der kartesischen Position berechnet in einer Endlosschleife die Vorwärts-Kinematik des Manipulators. Aufgrund der Rechenintensität des Prozesses ist diese Schleife mit einem Delay von 0,3 s getaktet. Dieser Prozess kann erst nach der korrekten Kalibration des Manipulators gestartet werden.

Der Bildschirminhalt während der Operation des Manipulators ist in Abbildung 9 dargestellt. Es werden die Winkel der drei Motorachsen A, B und C, die kartesische Position x und y, sowie der aktuell bearbeitete Steuerbefehl und der Name der Steuerdatei angezeigt.

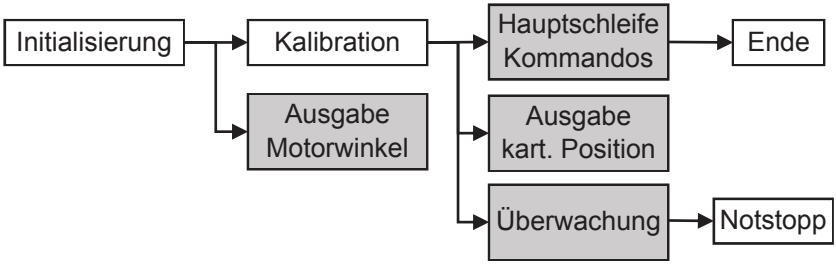

Abbildung 8: Struktur der Prozesse im Manipulator. Grau hinterlegte Blöcke enthalten (Endlos-)Schleifen

Der Überwachungsprozess stoppt den Manipulator, wenn die Positionen der Kurbeln den vorgesehenen Arbeitsbereich von $180° \geq \varphi \geq 0°$ verlassen, wobei der Kalibrationstaster-Anschlag jeweils eine der Grenzen darstellt. Deshalb kann auch dieser Prozess erst nach der Kalibration gestartet werden. Zudem kann jeder Programmteil im Manipulator, der eine gefährliche Situation erkennt, über Setzen eines globalen Error-Flags dem Überwachungsprozess einen Not-Stopp anordnen.

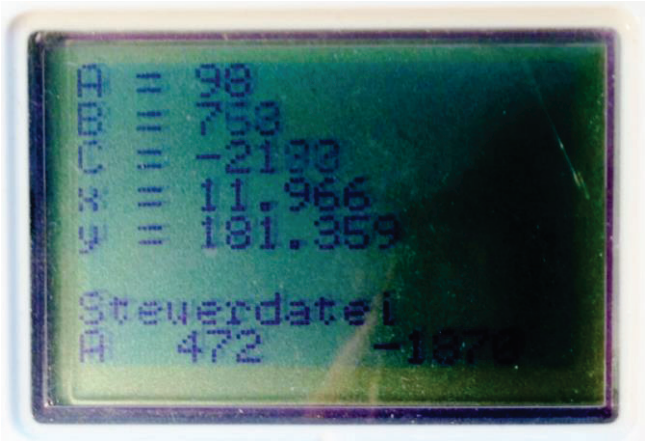

Abbildung 9: Bild des NXT-Displays während der Bewegung des Plotters

Speziell die Berechnung der Vorwärts-Kinematik erkennt eine Überstreckung der Hebel im Punkt G mit einer folgenden Selbstkollision des Stifts am Hebel B‘-G und löst in diesem Fall einen Not-Stopp aus.

5 Ergebnisse

Im ersten Beispiel lautet die Aufgabe, die Zahl 70 in einer Blockschrift, die ausschließlich aus geraden Strecken besteht, zu zeichnen. In Abbildung 10 ist eine Simulation dieser Aufgabe dargestellt. Auf der rechten Hälfte der Abbildung ist das gewünschte Ergebnis im kartesischen Arbeitsraum dargestellt. Die dicken Striche symbolisieren Bewegungen mit abgesenktem Stift, die dünnen Linien Bewegungen des Manipulators ohne Zeichnen.

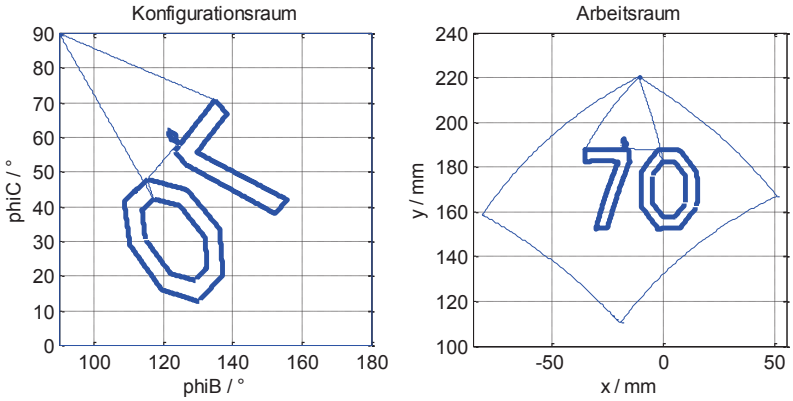

Abbildung 10: Simulation einer Zeichnung im Konfigurations- und Arbeitsraum

Auf der linken Hälfte der Abbildung sind die korrespondierenden Bewegungen im Konfigurationsraum dargestellt. Die Vorwärtskinematik und die inverse Kinematik sind die Abbildungsfunktionen von links nach rechts bzw. rechts nach links. Um das gewünschte Ergebnis im Arbeitsraum zu erzielen, muss im Konfigurationsraum eine verzerrte, gespiegelte und verdrehte Linienfolge abgefahren werden.

Auf Abbildung 11 ist der Scan eines tatsächlichen Ergebnisses der Zeichnung des Plotters in Originalgröße dargestellt. Es lassen sich noch leichte Abweichungen zwischen Wunschbild und tatsächlichem Bild erkennen. Diese lassen sich auf unvermeidliches Spiel und Nachgiebigkeit des realen Manipulators zurückführen. Die kleinen Fehler jeweils in den linken oberen Ecken der Konturen entstehen, wenn der Manipulator den Stift auf das Papier aufsetzt und dadurch ein kleiner Ruck entsteht.

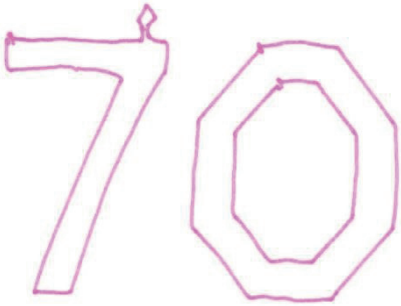

Abbildung 11: Gescanntes Bild der Zeichnung des Plotters

Auf der Abbildung 10 rechts im Arbeitsraumbild ist zusätzlich die entstehende Berandungskurve des maximalen Konfigurationsraums, die der Darstellungsfläche der Abbildung 10 links entspricht, eingezeichnet. Es ist sichtbar, dass die Kinematik des Manipulators zu einem vergleichsweise kleinen und ungünstig geformten Arbeitsraum führt. Allerdings ist dieser Raum deutlich von den Singularitäten, sowohl der Kurbel-Hebel-Gelenke in den Punkten B‘ und C‘ als auch vom Durchschlag im Punkt G entfernt, sodass er weitgehend mit guter Kondition für Zeichenaufgaben genutzt werden kann.

Im Rahmen der Lehrveranstaltung räumliche Kinematik/Roboterkinematik, die für Studierende im 8. Semester des Studiengangs Mechatronik vorgesehen ist, erhalten die Teilnehmer den Auftrag, das Logo der TU Dresden vom Plotter zeichnen zu lassen. Aus dem Befehlsumfang des Manipulators dürfen allerdings nur die Bewegungsbefehle im Konfigurationsraum genutzt werden. Somit müssen die Studierenden das Logo in geeigneter Weise digital aufbereiten, die inverse Kinematik mit Mitteln ihrer Wahl berechnen und eine Steuerdatei für den Plotter erstellen. Ein Matlab-Programm, das die Steuerdateien für den Manipulator testweise verarbeiten kann (damit wurden auch die Abbildung 10 und die Abbildung 12 erzeugt), steht den Studierenden zum Test ihrer Dateien zur Verfügung.

Auf Abbildung 12 ist das Logo der TU Dresden in der Simulation im Konfigurations- und Arbeitsraum dargestellt.

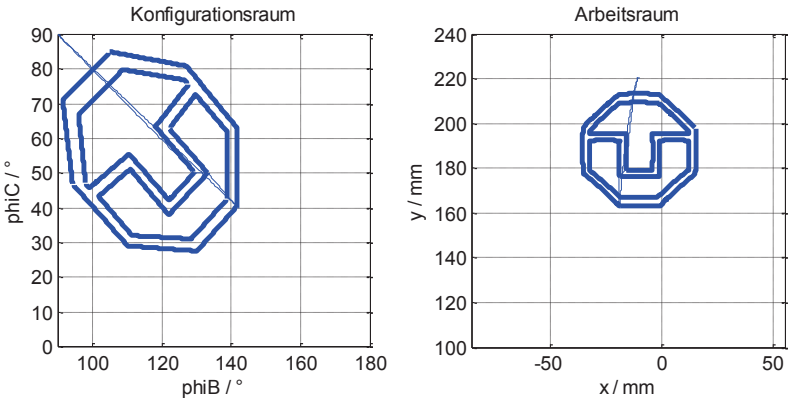

Abbildung 12: Simulation der Zeichnung des Logos der TU Dresden

Auf Abbildung 13 ist der Scan eines tatsächlichen Ergebnisses der Zeichnung des Plotters in Originalgröße dargestellt.

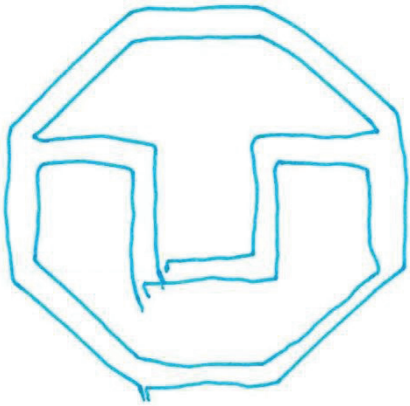

Abbildung 13: Gescanntes Bild der Zeichnung des TU Dresden Logos

Die Abweichungen zwischen Simulation und Realität sind auf Spiel in den Antrieben und Gelenken zurückzuführen. Trotz der Vorspannung des Mechanismus mit Gummibändern kann es in Gelenkstellungen mit $\varphi_C \rightarrow 90°$ und $\varphi_B \rightarrow -90°$ zum Abheben der Zahnflanken im Schneckengetriebe kommen.

6 Zusammenfassung

Lego Mindstorms ist ein geeignetes Werkzeug um einen einfachen parallelkinematischen Manipulator zu bauen. Mit diesem können die Zusammenhänge zwischen Bewegungen im Konfigurations- und Arbeitsraum anschaulich dargestellt werden. Mit einem Stift im TCP wird der Manipulator zum Plotter. Durchgeführte Bewegungen können somit auf Papier dokumentiert werden. Mit der Programmiersprache NXT-G wurde eine Software zur Steuerung des Manipulators entwickelt, die in der Lage ist, eine Textdatei mit Steuerbefehlen abzuarbeiten und die entsprechenden Bewegungsbefehle auszuführen. Das Gerät wurde bereits erfolgreich in der Lehre eingesetzt.

7 Ausblick

Mit Hilfe von Lego Mindstorms können im Prinzip weitere kinematische Konfigurationen und Manipulatoren gebaut werden. Auch eine serielle Kinematik mit zwei hintereinander geschalteten Drehgelenken ist denkbar und in Planung. Es sind auch räumliche Manipulatoren denkbar. Diese haben allerdings den Nachteil, dass die Nutzung als Plotter, bei dem die Bewegung des TCP unmittelbar auf Papier dokumentiert werden kann, entfällt. In nächster Zukunft soll der Plotter mit dem Nachfolger des NXT, dem EV3-Baustein von Lego realisiert werden. Eine deutlich gesteigerte Rechenleistung und ein neues Betriebssystem könnte eine Verbesserung des Programms, z.B. das tatsächliche Abfahren von Geraden im Arbeitsraum oder Bahnplanung mit Splines und Via-Punkten ermöglichen.