

HAUPTSEMINAR AUTOMATISIERUNGS-, MESS- UND REGELUNGSTECHNIK: ANLEITUNG UND AUFGABENSTELLUNG MODUL „CONTROL“

VERSION 4.3.0 (2019-10-15)

INHALT

1	Allgemeines.....	2
2	Aufgaben des Control-Moduls –Überblick	2
2.1	Line-Control (Line follower)	2
2.2	Velocity-Control	2
2.3	Park-Control (Path follower).....	3
2.4	Inactive	3
3	Aufgabenstellung	3
3.1	Linienerkennung und -verfolgung	3
3.2	Kinematik.....	4
3.2.1	Steuerung auf Grundlage kinematischer Berechnungen	4
3.2.2	Unterlagerte Drehzahlregelung.....	4
3.3	Regelung der Geradeausfahrt.....	4
3.3.1	Theoretische Vorbetrachtungen am linearisierten Modell	4
3.3.2	Implementierung und Test.....	5
3.4	Ein- und Ausparken mit Bahnplanung	5
3.4.1	Theoretische Vorbetrachtungen	5
3.4.2	Implementierung und Test.....	6
4	Dokumentation	7
5	Hinweise zur Bearbeitung und Bewertung der Aufgaben	8
5.1	Erste Verteidigung	8
5.2	Zweite Verteidigung	9
5.3	Allgemeine Hinweise zur Bewertung.....	9
5.4	Konsultationen mit dem Betreuer	9
	Anhang A – Kinematikberechnung.....	11
	Pseudocode für die Implementierung der Kinematikberechnung und Antriebssteuerung	Fehler! Textmarke nicht definiert.
	Anhang B – Roboterkinematik	12
	Robotermodell	12
	Linearisierung des Robotermodells.....	13

1 ALLGEMEINES

Das Modul Control beinhaltet alle Funktionen zur Steuerung und Regelung des Lego NXT. Dazu sollen die durch die Module Navigation und Perception gewonnenen Informationen genutzt und die Aufgaben durch geeignete Algorithmen realisiert werden.

2 AUFGABEN DES CONTROL-MODULS –ÜBERBLICK

Das Modul Control implementiert das bereitgestellte Interface IControl, welches die benötigten Schnittstellen enthält. Dadurch können andere Module unabhängig vom Fortschritt des Control-Moduls die Schnittstellen verwenden. Änderungen am Code beeinflussen dadurch nicht die anderen Module. Das Modul soll die folgenden Aufgaben umsetzen:

2.1 LINE-CONTROL (LINE FOLLOWER)

Der NXT soll befähigt werden, autonom eine schwarze Linie zu erkennen und ihr zu folgen. Dafür stehen zwei Lichtsensoren zur Verfügung, deren Helligkeitswerte vom Perception-Modul bereitgestellt werden.

2.2 VELOCITY-CONTROL

Hiermit soll realisiert werden, dass das Fahrzeug sich mit einer konstanten translatorischen Geschwindigkeit v (in m/s) und einer konstanten Winkelgeschwindigkeit ω (in rad/s), jeweils bezogen auf den Achsmittelpunkt bewegt. Die Größen v bzw. ω werden dabei als Eingangssignale des Roboters aufgefasst). Diese Methode dient als Grundlage für Park-Control. Mit diesem Ansatz kann über eine „Drehen-Fahren-Drehen“-Sequenz (siehe Abb. 1) jede beliebige Pose (Kombination aus Position und Orientierung) erreicht werden.

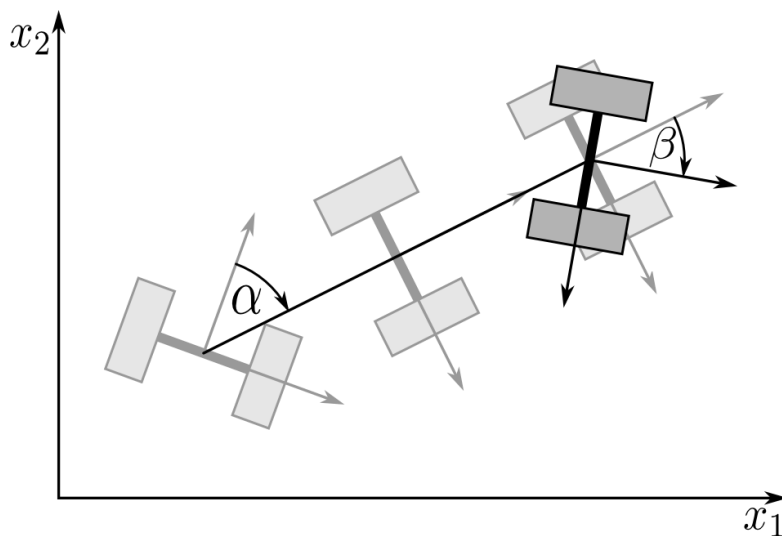


ABBILDUNG 1 – DREHEN-FAHREN-DREHEN-SEQUENZ

2.3 PARK-CONTROL (PATH FOLLOWER)

Mit Hilfe einer Bahnfolgeregelung soll das Fahrzeug entlang eines geplanten Pfades an seinen Bestimmungsort gebracht werden. Diese Methode soll zum Ein- und Ausparken verwendet werden.

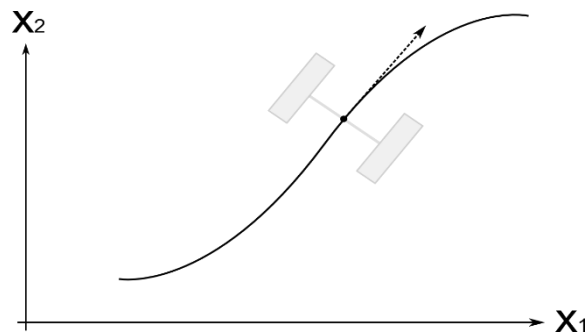


ABBILDUNG 2 – EIN-UND AUSPARKEN ENTLANG EINES PFADES

2.4 INACTIVE

Das Fahrzeug wird in Ruhe versetzt.

3 AUFGABENSTELLUNG

3.1 LINIENERKENNUNG UND -VERFOLGUNG

Es soll ein Algorithmus implementiert werden, der mit Hilfe der am NXT vorhandenen Lichtsensoren eine schwarze Linie erkennen kann und dieser folgt. Hierzu ist bereits ein einfaches Beispielprogramm vorgegeben, welches zu verbessern ist.

Das Modul Perception stellt hierfür die benötigten Messerwerte in zwei Varianten bereit. Die erste Variante ordnet den drei Fällen weiß, grau und schwarz jeweils eine Zahl zu, die abgefragt werden kann. Die zweite Variante erlaubt eine elegantere Implementierung, da sie je nach Lichtintensität einen Wert von 0 bis 100 ausgibt.

1. Verbessern Sie den Algorithmus zur Linienverfolgung aus dem bereits implementierten Beispielprogramm, welcher die drei diskreten Farbwerte (Variante 1) nutzt.
2. Dokumentieren Sie die vorgenommenen Veränderungen und ihre Auswirkungen.

Es soll nun ein Regler implementiert werden, der mit Hilfe der Lichtintensität (Variante 2) einen PID-Regelalgorithmus nutzt.

3. Definieren Sie unter Berücksichtigung der vorhandenen Sensorik und Ihrer konstruktiven Gegebenheiten eine geeignete skalare Regelgröße und den zugehörigen Regelfehler (Sollwert minus Istwert).
4. Implementieren Sie auf Grundlage dieses Fehlers einen PID-Regler. Überlegen Sie, welcher der drei Anteile P-I-D ggf. verzichtbar ist.
5. Dokumentieren und Bewerten Sie Ihr Ergebnis.

Anmerkung: Die Implementierung erfolgt nicht modellbasiert, d.h. Reglerparameter müssen empirisch und nicht auf Grundlage von theoretischen Vorüberlegungen gewonnen werden. Gegebenenfalls kann es sinnvoll sein, zusätzlich zum Regel-Algorithmus Mechanismen zu implementieren, die eine Kurvenfahrt erleichtern.

3.2 KINEMATIK

3.2.1 STEUERUNG AUF GRUNDLAGE KINEMATISCHER BERECHNUNGEN

Es soll eine Steuerung implementiert werden, durch die das Fahrzeug mit der Geschwindigkeit v und der Winkelgeschwindigkeit ω fährt. Diese Methode stellt somit eine Schnittstelle für die Eingangssignale des Systems zur Verfügung. Dafür soll die Klasse `NXTMotor` benutzt werden (diese Klasse verwendet keine untergelagerten PID-Regler für die Motoren).

1. Bestimmen Sie experimentell für jedes angetriebene Rad den Zusammenhang zwischen dem PWM-Tastverhältnis („Power-Werte“) und der erreichten Drehgeschwindigkeit.
2. Implementieren Sie eine Steuerung auf Grundlage der dieser Ergebnisse und Kinematikberechnungen im Anhang A. Machen Sie sich dafür den Unterschied zwischen Steuerung und Regelung klar.
3. Testen Sie die Steuerung mit einfachen Werten, zum Beispiel für eine Geradeausfahrt, einen Kreis bzw. eine Ellipse. Beschreiben Sie ihre Beobachtungen. Welche Schlussfolgerungen ziehen Sie aus ihren Beobachtungen?

3.2.2 UNTERLAGERTE DREHZAHLEGEUNG

Beim Testen der vorherigen Aufgabe werden Sie feststellen, dass das Fahrzeug von seiner vorgegebenen Route abweicht. Trotz der Vorgabe gleicher Power-Werte drehen sich die Motoren unterschiedlich schnell. Dies liegt zum einen daran, dass die Motoren nicht identisch sind und zum anderen ist die tatsächlich realisierte Antriebskraft stark vom Ladezustand des Akkus abhängig. Um diesen unerwünschten Effekten zu begegnen, ist für jeden Motor eine unterlagerte Drehzahlregelung erforderlich.

1. Implementieren Sie für jedes angetriebene Rad eine Geschwindigkeitsmessung unter Verwendung der vom Interface `IPerception` bereitgestellten Methoden und Klassen. Definieren Sie eine geeignete Regelgröße und den zugehörigen Regelfehler.
2. Nutzen Sie die Geschwindigkeitsmessung um eine separate Drehzahlregelung für jedes Rad zu realisieren (PID-Regler). Orientieren Sie sich an den PID-Einstellregeln in der Praktikumsanleitung „V0“ vom Institut für Regelungs- und Steuerungstheorie.

Hinweis: Auch an dieser Stelle erfolgt die Auslegung des Reglers empirisch und nicht modellbasiert!

3. Vergleichen Sie die Ergebnisse mit denen der vorherigen Teilaufgabe sowohl qualitativ (Beurteilung, Einschätzung) als auch quantitativ (Angabe von Zahlenwerten bezüglich Genauigkeit).

3.3 REGELUNG DER GERADEAUSFAHRT

Ungeachtet der unterlagerten Drehzahlregelung für die Motoren erfolgt die Vorgabe von v und ω bisher als Steuerung. Im Folgenden soll eine Positionsregelung umgesetzt werden.

3.3.1 THEORETISCHE VORBETRACHTUNGEN AM LINEARISIERTEN MODELL

Als Ausgangspunkt dient das kinematische Modell in Anhang B.

1. Linearisieren Sie das nichtlineare Modell für den Fall einer Geradeausfahrt in x_2 -Richtung.

Hinweis: Nutzen Sie zur Linearisierung die Hinweise aus der Regelungstechnik-Vorlesung, der 2. Übung und dem Anhang. Überlegen Sie sich, wie ein sinnvoller „Arbeitspunkt“ für die Geradeausfahrt definiert ist.

2. Beschreiben Sie die Querabweichung e mit Hilfe einer geeigneten **linearen** Gleichung in der Form: $e = f(x_1, x_2, x_3, u_1, u_2)$!

Hinweis: Es müssen nicht notwendiger Weise alle Elemente von x_i und u_i enthalten sein.

- Erstellen Sie zur Regelung der Geradeausfahrt ein lineares dynamisches Modell, d.h. eine DGL, der Form $e^{(n)} = g(u_1, u_2)$, welches die Vorwärtsbewegung in der Nähe der **konstanten** Sollgeschwindigkeit $v_0 > 0$ beschreibt. Verwenden Sie dafür die unter 2. aufgestellte Gleichung.

Hinweis: Ziel ist die Ermittlung einer Differentialgleichung in e , zu deren Berechnung nur noch die beiden Eingänge u_1 und/oder u_2 benötigt werden. Dieses Modell dient der Beschreibung des Einflusses der Stellgrößen u_1 und u_2 auf den Fehler e .

- Untersuchen Sie das Systemverhalten mit verschiedenen Reglertypen (P-, PI-, PD- und PID) und Beurteilen Sie jeweils die Stabilität des geschlossenen Regelkreises mit Hilfe des Wurzelortskurvenverfahrens. Wählen Sie einen Reglertyp aus und bestimmen Sie eine geeignete Parametrierung des Reglers.

Hinweis: Zur Veranschaulichung des Systemverhaltens können die im OPAL bereitgestellten Skripte genutzt werden. Fragen Sie ggf. den Betreuer.

3.3.2 IMPLEMENTIERUNG UND TEST

Das Lego NXT-Fahrzeug soll auch ohne Fahrbahnmarkierung relativ weite Strecken mit möglichst geringen Positionsabweichungen überbrücken können. Dazu dient die geregelte Geradeausfahrt, bei der unterwegs auftretende Störungen kompensiert werden. Das Modul Navigation stellt jeweils die aktuelle Pose bereit (LeJOS-Datenstruktur, welche die Koordinaten für x , y und die Ausrichtung enthält). Aus dieser kann zusammen mit einer mathematischen Beschreibung der zu verfolgenden Gerade die aktuelle Querabweichung bestimmt werden.

- Implementieren Sie eine Methode zur geregelten Geradeausfahrt, welche den oben bestimmten Regler benutzt.

Hinweise:

- Ggf. ist eine geeignete Koordinatentransformation notwendig.
- Diese Methode kann zu einer Drehen-Fahren-Drehen-Sequenz erweitert werden und stellt dann eine einfache Testmöglichkeit für das Ein- und Ausparken dar, ohne die relativ aufwendige Bahnplanung des Guidance-Moduls.
- Ggf. kann der für die Geradeausfahrt entworfene Regler auch für die Verfolgung der Fahrbahnmarkierung (Line-Control) eingesetzt werden.

3.4 EIN- UND AUSPARKEN MIT BAHNPLANUNG

Es soll eine Methode implementiert werden, die den NXT eine definierte glatte Bahn abfahren lässt. Als Stellgrößen für die Bewegung werden hier wieder die Größen v und ω aufgefasst und unterlagert in die Radgeschwindigkeiten umgerechnet.

Hinweis: Die Bearbeitung des Abschnitts 3.4 muss in enger Abstimmung mit dem/der Verantwortlichen für das Guidance-Modul erfolgen.

3.4.1 THEORETISCHE VORBETRACHTUNGEN

Mit Hilfe von Polynomen soll eine Bahnkurve für den Einparkvorgang hergeleitet werden, die den Verlauf der Bahn in der x_1 - x_2 -Ebene beschreibt.

- Welche Vorteile bietet eine explizite Bahnplanung für den Einparkvorgang im Vergleich zu einer Drehen-Fahren-Drehen-Sequenz?
- Bestimmen Sie unter Verwendung des Robotermodells die Gleichungen zur Berechnung der Stellgrößen v und ω aus den Verläufen von x_1 und x_2 . Auf welches Problem stoßen Sie dabei und wie können Sie dieses Problem lösen?

3. Welche geometrischen Randbedingungen sind für die Pfad zu wählen? Welchen Grad benötigen die Polynome?
4. Stellen Sie die Polynome auf, um den Anfangspunkt mit dem Endpunkt zu verbinden und geben Sie die Koeffizienten der Polynome an.

3.4.2 IMPLEMENTIERUNG UND TEST

1. Stellen Sie sicher, dass das Guidance-Modul eine funktionsfähige Methode zur Bahnplanung zwischen zwei Posen aufweist (Parklücke und Fahrbahnmitte).
2. Implementieren Sie im Control-Modul eine Methode zur Berechnung und Umsetzung der Stellgrößen, sodass die geplante Bahn möglichst genau abgefahren wird.
3. Testen Sie das Ein- und Ausparken entlang der geplanten Bahn. Bewerten Sie das Ergebnis.

Hinweis: Zur Veranschaulichung des Systemverhaltens können die bereitgestellten Skripte genutzt werden. Ggf. beim Betreuer Nachfragen.

4 DOKUMENTATION

In der Dokumentation sollen die Lösungen und Lösungswege zu den gestellten Aufgaben nachvollziehbar und ansprechend dargestellt werden. Die Abarbeitung der jeweiligen Teilaufgaben ist in einzelne Schritte unterteilt. In der Dokumentation wird erwartet, dass Sie zu jedem der Schritte eine kurze Aussage treffen. Notieren Sie, wann immer möglich, lieber eine Formel anstatt umfangreich formulierter Absätze. Oftmals ist eine Gleichung eindeutiger als die sprachliche Formulierung.

Der Umfang der Dokumentation soll **15 echte Seiten pro Student (ohne evtl. Anhang, ohne Titel und Inhaltsverz.)** nicht überschreiten. **Hochwertige kürzere** Dokumentationen sind willkommen. Weitere Hinweise zur Bewertung und Abgabe der Dokumentation sind in der allgemeinen Anleitung zu finden.

Es ist sehr empfehlenswert, während der Durchführung des Hauptseminars bereits begleitend an der Dokumentation zu schreiben.

Der Inhalt muss sich an nachfolgend gegebenem Inhaltsverzeichnis orientieren.

1. Analyse der Aufgabenstellung
 - 1.1. Motivation, Aufgabenstellung / Ziel des Seminars
 - 1.2. kurze Zusammenfassung der einzelnen Teilaufgaben
 - 1.3. Definition / Erläuterung der Übergabeparameter, welche das Control-Modul benötigt / liefert
2. Linienverfolgung
 - 2.1. Verbesserung des Beispielprogramms
 - 2.2. Linienverfolgung und Kurvenfahrt mit PID-Regler
 - 2.3. ...
3. v/ω -Control
 - 3.1. Steuerung des Roboters
 - 3.1.1. Berechnung der Radgeschwindigkeiten anhand des kinematischen Modells
 - 3.1.2. Beobachtung, Ergebnisbewertung, Folgerungen
 - 3.1.3. ...
 - 3.2. Geschwindigkeitsregelung der Räder
 - 3.2.1. Auslegung des Reglers
 - 3.2.2. Vergleich der Regelung mit der Steuerung
 - 3.2.3. ...
 - 3.3. ...
4. Regelung der Geradeausfahrt
 - 4.1. Theoretische Betrachtungen am linearisierten Modell
 - 4.1.1. ...
 - 4.1.2. Implementierung der Regelung
 - 4.1.3. ...
5. Ein- und Ausparken
 - 5.1. Theoretische Vorbetrachtungen der Bahnplanung
 - 5.2. Implementierung
 - 5.3. ...
6. ... (selbst hinzugefügte Abschnitte)
7. Anmerkungen und Verbesserungsmöglichkeiten

5 HINWEISE ZUR BEARBEITUNG UND BEWERTUNG DER AUFGABEN

Es finden insgesamt zwei Verteidigungen statt. Damit bis zur zweiten Verteidigung noch ausreichend Zeit zum Testen und für die Integration der Module besteht, sollen bestimmte Teilaufgaben bereits zur ersten Verteidigung funktionstüchtig sein. Typischerweise kann die Realisierung der einzelnen Teilaufgaben im Gesamtprojekt nicht vollständig gezeigt werden, da im Gesamtprojekt eine Interaktion mit anderen (unfertigen) Modulen stattfindet. Deshalb sind für jede Verteidigung Demo-Programme vorzubereiten. Diese werden im Anschluss an die Demonstration des Gesamtprojektes aufgespielt und vorgeführt. Die Demo-Programme können die Vorgaben als direkten Programmcode enthalten. Eine Schnittstelle zur Datenübergabe (bspw. über das Tablet) ist nicht erforderlich. Der Quelltext ist zusammen mit der Präsentation vor der Verteidigung einzureichen.

Die zeitlichen Rahmenbedingungen sind in der allgemeinen Anleitung aufgeführt. Die Verteidigung besteht aus drei Teilen: Vorführung, Vortrag und Diskussion. Zu Beginn erfolgt die Vorführung. Ein vorbereiteter Roboter wird vorausgesetzt. Im Anschluss folgt der Vortrag. Aus diesem muss für jede Teilaufgabe klar ersichtlich werden, ob sie gelöst wurde, wie sie gelöst wurde und welche Probleme es ggf. noch gibt. Die Lösungsfindung muss **nachvollziehbar** dargelegt werden. Es soll erläutert werden, welche Überlegungen zur Lösung des jeweiligen Problems angestellt wurden und es soll deren Umsetzung bzw. das Ergebnis beurteilt / bewertet werden. Auf Grundlage des Vortrages und der Vorführung der Demo-Programme folgt dann eine Diskussion mit dem Modulbetreuer. Alle drei Teile werden bewertet.

***Hinweis:** Für die Implementierung der Control-Demo-Programme kann bspw. eine Methode in der Control-Klasse definiert werden, welche zu Beginn aufgerufen wird und in der die erforderliche Sequenz hinterlegt ist. Es wird dann in jedem Funktionsaufruf (z.B. alle 100ms) die Ist-Position geprüft und bei ausreichend gut erreichter Zielvorgabe der nächste Schritt in der Sequenz (siehe jeweilige Anforderungen) ausgeführt. Der Übergang in den nächsten Schritt könnte bspw. mit Hilfe des Summers oder einer Ausgabe auf dem Display angezeigt werden.*

5.1 ERSTE VERTEIDIGUNG

In der ersten Verteidigung ist die Funktionstüchtigkeit folgender Teilaufgaben nachzuweisen:

- Linienverfolgung
- Geschwindigkeitsregelung (v/ω)
- Geregelte Geradeausfahrt

Darüber hinaus sind in der Präsentation die Ergebnisse der theoretischen Vorbetrachtungen aus Teilaufgabe 3.3.1 darzulegen.

Anforderungen an Control-Demo-Programm-1:

- Geradeausfahrt einer Strecke von 120cm mit einer Geschwindigkeit von 10cm/s (v-Control)
- Drehung um 90° in math. pos. Richtung, 15°/s (ω -Control)
- Geradeausfahrt 30cm mit 5cm/s (v-Control)
- Drehung um 90° in math. neg. Richtung, 30°/s (ω -Control)
- Linienverfolgung mit Höchstgeschwindigkeit bis zum Startpunkt (Line-Control)

Hinweise:

- Überlagern Sie diesen Sequenzen keinen Positionsregler oder andere Regelungsmechanismen zur Verbesserung der Positionsgenauigkeit. Nutzen Sie lediglich die angegebenen Funktionen.
- Die vorgegebenen Geschwindigkeiten sollten technisch realisierbar sein. Gelingt dies nicht, so verwenden sie die ihnen höchstmöglichen Geschwindigkeiten. Die Güte der Steuerung bzw. Regelung ist wichtiger.

- Bei Bedarf können Sie die NXT-Tasten benutzen (Unabhängigkeit des Control-Demo-Programms vom HMI-Modul)

5.2 ZWEITE VERTEIDIGUNG

In der zweiten Verteidigung ist die Funktionstüchtigkeit aller Teilaufgaben nachzuweisen:

- Linienverfolgung
- v/ω -Control
- Geregelte Geradeausfahrt
- Park-Control

Hinweis: In der zweiten Verteidigung sollte der Schwerpunkt der Vorführung und Präsentation auf den Aspekten liegen, die noch nicht Bestandteil der 1. Verteidigung waren oder dort noch nicht funktioniert haben.

Anforderungen an Control-Demo-Programm-2:

- Ausgangspunkt: Startpose
- Geradeausfahrt einer Strecke von 120cm mit einer Geschwindigkeit von 10cm/s (v-Control, keine Linienfolge)
- Drehung um 90° in math. pos. Richtung, 15°/s (ω -Control)
- Geradeausfahrt 30cm mit 5cm/s (v-Control)
- Drehung um 90° in math. neg. Richtung, 30°/s (ω -Control)
- Linienverfolgung entgegen der üblichen Fahrtrichtung mit Höchstgeschwindigkeit bis zum Startpunkt (Line-Control)
- Drehung um 180° mit Höchstgeschwindigkeit, Drehrichtung beliebig (ω -Control)
- Einparken in die erste Lücke (es werden keine Hindernisse aufgestellt) (Park-Control)

Anforderungen an Control-Demo-Programm-3:

- Ausgangspunkt: Startposition
- auf direktem Weg in die erste Parklücke einparken (Parklücke einprogrammieren, kein Scout-Modus)
- Ausparken
- Linienverfolgung bis zur ersten Ecke
- Einparken in die Parklücke an der rechten Seite (nach der Kurve)
- Ausparken
- Linienverfolgung

5.3 ALLGEMEINE HINWEISE ZUR BEWERTUNG

Bei der Implementierung der Geschwindigkeitsregelung der Räder soll die Klasse NXTMotor verwendet werden. Es gibt darüber hinaus eine Klasse NXTRegulatedMotor, welche intern die Regelung der Motoren übernimmt. Es ist untersagt, diese Klasse zu verwenden, da es die Aufgabe dieses Moduls ist, unter anderem genau diese Funktionalität zu implementieren. Wird diese Klasse dennoch verwendet, wird dies als Betrugsversuch gewertet und das Seminar gilt damit als nicht bestanden! Eine Verwendung zu Testzwecken ist natürlich gestattet.

In den Verteidigungen und vor allem in der Dokumentation sind die eigenen Aussagen quantitativ und/oder qualitativ zu belegen. Das heißt, dass bspw. empirisch ermittelte Reglerparameter angegeben werden. An den Stellen wo Stabilitätsuntersuchungen durchgeführt werden, sind diese nachvollziehbar darzustellen (Formel, Diagramme, o.Ä.). Die Güte einer Regelung ist durch Messungen zu belegen.

5.4 KONSULTATIONEN MIT DEM BETREUER

Während des Seminars steht ihnen ein Betreuer zur Unterstützung zur Verfügung. Wenden Sie sich bei Fragen an ihren Betreuer! Nutzen Sie dieses Angebot, es wird ihnen nicht als Nachteil ausgelegt, wenn Sie sich mit Fragen melden! Entgegen den Angaben im Stundenplan ist für eine Konsultation keine konkrete Zeit vorgese-

hen. Vereinbaren Sie ihre Termine je nach Bedarf individuell per Mail. Bei der Konsultation wird jedoch erwartet, dass Sie sich entsprechend vorbereitet haben und konkrete Fragen stellen.

ANHANG A – KINEMATIKBERECHNUNG

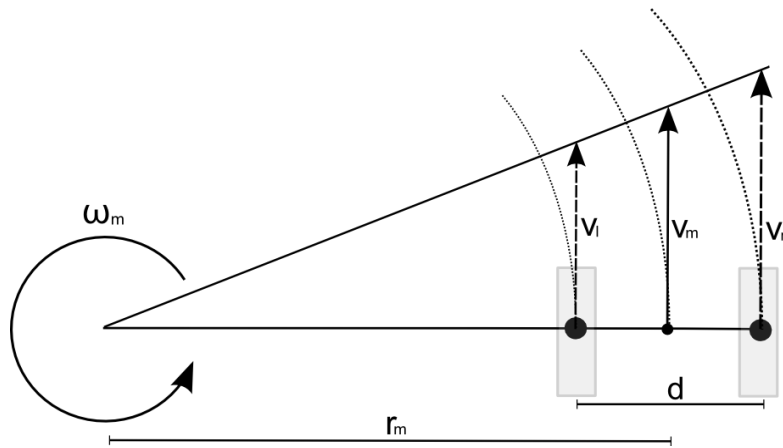


ABBILDUNG 3 – KINEMATIKBERECHNUNG

Der Radius des zu fahrenden Kreisbogens berechnet sich aus den zu übergebenen Werten für die translatorische Geschwindigkeit und die Winkelgeschwindigkeit im kinematischen Zentrum.

$$r_m = \frac{v_m}{\omega_m}$$

Der Zusammenhang zwischen translatorischer Geschwindigkeit des Achsmittelpunktes v_m und Winkelgeschwindigkeit ω einerseits und den Geschwindigkeiten einzelnen Räder andererseits ist durch das folgende Gleichungssystem gegeben:

$$\begin{pmatrix} v_m \\ \omega \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(v_r + v_l) \\ \frac{1}{d}(v_r - v_l) \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{d} & \frac{1}{-d} \end{pmatrix} \begin{pmatrix} v_r \\ v_l \end{pmatrix}.$$

ANHANG B – ROBOTERKINEMATIK

ROBOTERMODELL

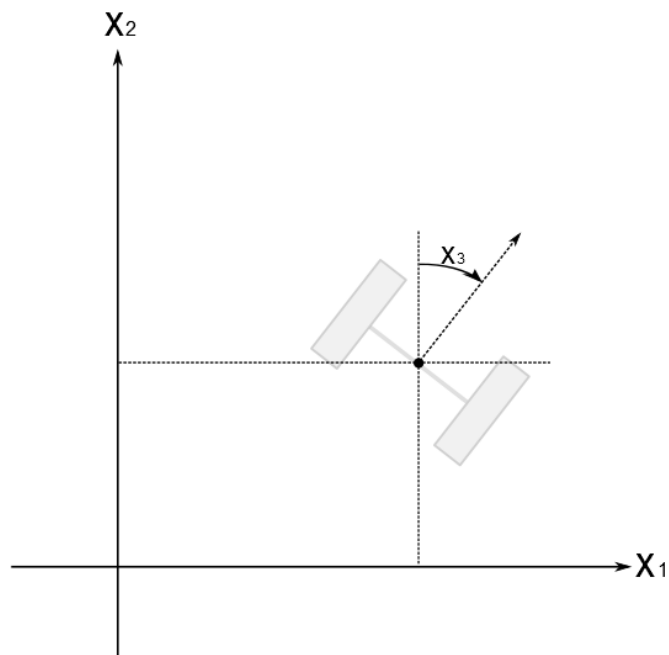


ABBILDUNG 4 - EINACHSIGER ROBOTER

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} \sin(x_3) \\ \cos(x_3) \\ 0 \end{pmatrix} \cdot v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \omega$$

$$\mathbf{x} = (x_1 \ x_2 \ x_3)^T \text{ und } \mathbf{u} = (u_1 \ u_2)^T = (v \ \omega)^T$$

LINEARISIERUNG DES ROBOTERMODELLS

An dieser Stelle sollen ein paar Hinweise bezüglich der Linearisierung gegeben werden. Nutzen Sie des Weiteren auch die Unterlagen aus der Vorlesung und den Übungen.

Das Robotermodell kann auch in der Form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

angegeben werden („Zustandsraumdarstellung“). Dabei ist \mathbf{f} ein Vektor von Funktionen

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} f_1(\mathbf{x}, \mathbf{u}) \\ f_2(\mathbf{x}, \mathbf{u}) \\ f_3(\mathbf{x}, \mathbf{u}) \end{bmatrix} = \begin{bmatrix} \sin(x_3)v \\ \cos(x_3)v \\ \omega \end{bmatrix}$$

mit $\mathbf{x} = [x_1 \ x_2 \ x_3]^T$ und $\mathbf{u} = [v \ \omega]^T$.

Die Linearisierung eines Vektors aus Funktionen mit Hilfe der Taylorreihe um den Arbeitspunkt $(\mathbf{x}_0, \mathbf{u}_0)$ lautet

$$\begin{aligned} \mathbf{f}(\mathbf{x}, \mathbf{u}) &= \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}} (\mathbf{x} - \mathbf{x}_0) + \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}} (\mathbf{u} - \mathbf{u}_0) + T. h. O. \\ &\approx \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \left[\begin{array}{cc} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_3} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_3}{\partial x_1} & \dots & \frac{\partial f_3}{\partial x_3} \end{array} \right]_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}} \begin{bmatrix} x_1 - x_{10} \\ x_2 - x_{20} \\ x_3 - x_{30} \end{bmatrix} + \left[\begin{array}{cc} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{array} \right]_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}} \begin{bmatrix} u_1 - u_{10} \\ u_2 - u_{20} \end{bmatrix}. \end{aligned}$$

Durch die Vernachlässigung der Terme höherer Ordnung erhält man wieder eine Darstellung der Form

$$\dot{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}),$$

wobei nun in $\tilde{\mathbf{f}}$ die Elemente von \mathbf{x} und \mathbf{u} nur noch mit erster Ordnung auftreten. Ggf. ist auch die Einführung von neuen Koordinaten $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$ (analog für \mathbf{u}) hilfreich. Die Terme nullter Ordnung entfallen weil $(\mathbf{x}_0, \mathbf{u}_0)$ ja als Arbeitspunkt (=Ruhelage) gewählt wurde, d. h. es gilt $\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) = \mathbf{0}$.