

# Regelung eines Schlangenroboters in einem Rohr

Jens Balze, Nursura Demircan, Elias Jargon, Seyed Davood Mousavi

**Zusammenfassung**—Bereits heute werden mehrgliedrige, schlangenähnliche Roboter auf Bohrinseln und bei Ölpipelines für Inspektions-, Instandhaltungs- und Reparaturarbeiten eingesetzt. Diese Arbeit setzt sich mit der Regelung der Bewegung eines solchen Roboters, der zukünftig zur Inspektion innerhalb eines Teilchenbeschleunigers eingesetzt werden soll, auseinander. Dabei soll die Fahrt des Roboters so geregelt werden, dass der Roboter sich an unterster Stelle im Rohr bewegt und zusätzlich der Gierwinkel, der den Winkel zwischen der z-Achse (Rohrachse) und der Bewegungsrichtung des Roboters darstellt, zu Null geregelt wird. Zuerst wird die Herleitung des mathematischen Modells, das die Roboterbewegung im zylinderförmigen Rohr beschreibt, vorgestellt. Anschließend wird auf die Reglerauslegung eingegangen. Ein einfacher Versuchsaufbau zum Testen der Reglerauslegung ist ebenfalls Bestandteil dieser Arbeit. Zuletzt werden die entworfenen Regler anhand der Testergebnisse des realen Systems und der Simulation bewertet und miteinander verglichen. Abschließend wird ein Ausblick auf Erweiterungen dieser Arbeit gegeben.

**Abstract**—Already today, multi-limbed, snake-like robots are used on drilling rigs and oil pipelines for inspection, maintenance and repair. This paper deals with the regulation of the movement of such a robot, which should be used for the inspection within a particle accelerator in the future. The aim is to control the motion of the robot so that the robot moves at the lowest point in the pipe and in addition the yaw angle, which represents the angle between the z-axis (pipe axis) and the direction of movement of the robot, is regulated to zero. First, the derivation of the mathematical model describing the robot motion in the cylindrical pipe is presented. Subsequently, the controller design will be presented. A simple test setup for testing the controller design is also part of this work. Lastly, evaluation and comparison of the controller designs against each other, based on the test results of the real system and the simulation, are discussed. Finally, an outlook on extensions of this work is given.

## I. EINFÜHRUNG

Zahlreiche Roboter finden sich bereits heute etwa in der Elektrotechnik und Elektronik, im Maschinenbau, in der Luft- und Raumfahrt und in der Lebensmittelverpackung. Roboter sind ein guter Ersatz für die Humanressource, da sie vielfach günstiger als ?Kollege? Mensch arbeiten, flexibel sind und stets weiterentwickelt werden, sodass sich eine Investition lohnt.

Roboter gehören zur großen Gruppe von Assistenzsystemen, die unterschiedliche Entwicklungsstufen und Reifegrade besitzen. Dazu gehören beispielsweise Smartphones, die informelle und kognitive Funktionen erfüllen können (wie auch Tablets oder Smartwatches). Assistenzsysteme können aber auch Exoskelette sein, die Personen bei Bewegungen ergonomisch unterstützen, Roboter, die mit ihnen kooperieren und

bei der Aufgabenbewältigung unterstützen oder humanoide Roboter, die komplex mit Personen zusammenarbeiten und auf unterschiedlichen Sinnesebenen (zum Beispiel visuell, auditiv, haptisch) Informationen mit den Personen austauschen.

Ein Serviceroboter ist eine frei programmierbare Bewegungseinrichtung, die teil- oder vollautomatisch Dienstleistungen verrichtet (z. B. Inspektionsroboter in Rohrleitungen, Serviceroboter im Pflegebereich, Rasenmäherroboter, Industrieroboter).

Am Fachgebiet Regelungsmethoden und Robotik der TU Darmstadt wird derzeit ein schlangenähnlicher Roboter entwickelt, der das Innere eines Teilchenbeschleunigers visuell überwachen soll. Der Roboter soll das Rohr des Teilchenbeschleunigers durchfahren und Hindernisse wie Verunreinigungen oder unerwünschte Gegenstände, die die einwandfreie Funktion des Teilchenbeschleunigers behindern, identifizieren. Solche hätten möglicherweise eine Beeinträchtigung der Qualität des Strahls während des Betriebs des Teilchenbeschleunigers zur Folge.

Der Roboter wurde von der vorherigen Projektgruppe so programmiert, dass er sich autonom oder ferngesteuert auf einer Ebene bewegen und zusätzlich Kreisfahrten durchführen kann. Die Aufgabe dieser Arbeit ist die Regelung der Roboterbewegung im Rohr, sodass der Roboter Rollbewegungen und Kurven erkennt und diese ausregelt, sodass er an die unterste Stelle des Rohres zurückkehrt und an dieser Stelle mittig geradeaus weiterfährt. Zur Verfügung steht ein Sensor, der Winkelbeschleunigungen in allen Koordinatenrichtungen erfasst. Außerdem ist eine Simulation für die Roboterbewegung vorgegeben.

## II. AUFBAU DES ROBOTERS

In diesem Kapitel wird der Aufbau des Roboters beschrieben, für den die modellierten Regler implementiert und getestet werden. Der schlangenartige Roboter besteht aus fünf Modulen, welche mit drei aktiven und einem passiven Gelenk miteinander verbunden sind. Im Rahmen dieser Arbeit wurde eine Änderung am Roboter vorgenommen. Das vordere Modul wurde gesplittet und mit einem passiven Gelenk versehen, damit das Rollverhalten und die gemessene Regelabweichung besser ausgewertet werden kann. Durch den Aufbau aus Kunststoffteilen konnte der Prototyp schnell mittels 3D-Drucker angepasst werden.

Der Roboter besitzt acht einzeln ansteuerbare Servomotoren für den Antrieb. Am Ende dieser Motoren sitzen Räder, welche ebenfalls im 3D-Druckverfahren hergestellt wurden. Ein Nachteil dieser Konstruktion ist die geringe Haftreibung, welche in unseren Versuchen zu beobachten ist. Die drei für die Lenkung eingesetzten Servomotoren sind mit denen für den Antrieb identisch. Das ganze System wird mit einem Akku auf Lithium-Polymer Basis betrieben.

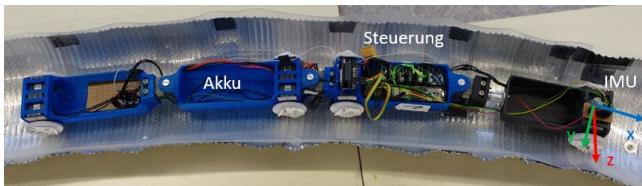


Abb. 1. Foto des Roboters. Von links nach rechts befindet sich später in den Modulen: ein Gewicht (repräsentativ für verbaute Elektronik), ein LiPo-Akku, der Mikrocontroller mit Funkmodul, ein weiteres Gewicht und die inertiale Messeinheit.

Wie in Abbildung 1 beschrieben, besitzt der Roboter eine inertiale Messeinheit (IMU) als Sensorik. Die Datenverarbeitung dieser, sowie die Ansteuerung der Servomotoren und die Kommunikation mit dem Computer, übernimmt ein Mikrocontroller.

#### A. Mikrocontroller

Die Größe des zu durchfahrenden Strahlrohrs eines Ringbeschleunigers bedingt eine schmale Bauform des Roboters. Somit ist auch die Größe des einzusetzenden Mikrocontrollers eingeschränkt. In diesem Projekt wird von der Firma PJRC das Teensy 3.6 Board eingesetzt, welches über die geforderte Baubreite und Rechenleistung verfügt. Mit einer Abmessung von nur 6,1cm x 1,8cm passt es sehr gut in eines der Robotermodule hinein. Ein Arm Cortex M4 Prozessor mit 180 MHz und 32-bit Technologie ist verbaut. Der ausgewählte Mikrocontroller besitzt eine Betriebsspannung von 3,3V. Das bedeutet für die Ansteuerung der Servomotoren ist ein PEGELUMSETZER (3,3V auf 5V) eingesetzt. Für die vollständigen Stromversorgung aller auf dem Roboter verbauten Geräte mit dem LiPo-Akku sind zusätzliche Spannungswandler für 3,3V, 5V und 7,4V nötig. [4]

#### B. Servomotoren

Die eingesetzten Servomotoren, Dynamixel XL-320, stammen von der Firma ROBOTIS. Hieron werden je acht Stück für den Antrieb und je drei Stück für die Lenkung eingesetzt. Für den Antrieb werden die Motoren im Wheel-Mode und für die Lenkung im Joint-Mode betrieben. Ihre Betriebsspannung liegt bei 7,4V. Für die Ansteuerung der Motoren wird eine serielle Schnittstelle des Mikrocontrollers verwendet [4].

#### C. Inertiale Messeinheit

Zur Messung der Regelgröße wird eine inertiale Messeinheit verwendet. Diese misst die Beschleunigung in alle Raumrichtungen und die Winkelbeschleunigungen um alle Koordinatenachsen. Durch Integration der Winkelbeschleunigungen können die Drehrate und die Lage bestimmt werden. Zum Einsatz in diesem Projekt kommt eine Messeinheit von iNEMO. Sie wird unter der Typenbezeichnung LSM6DS33 gelistet. Dieser Chip arbeitet im Performance-Mode mit bis zu 1,66kHz und einer Auflösung von +/-2G, beziehungsweise +/-250 DPS [5]. Da der Roboter mit einer niedrigen Geschwindigkeit durch einen Ringbeschleuniger fahren wird, kann davon ausgegangen werden, dass man nicht mit großen linearen Beschleunigungen

und großen Winkelbeschleunigungen beaufschlagt wird. Somit ist die gewählte Auflösung ausreichend gewählt und entspricht der Standardeinstellung der Messeinheit.

Zur Verbesserung der Genauigkeit des Beschleunigungssensors wird eine Biaskorrektur auf dem Mikrocontroller durchgeführt [6]. Die Messeinheit kommuniziert mittels I2C mit dem Board.

#### D. Ansteuerung und Kommunikation

Für eine kabellose Kommunikation mit dem Roboter wird das Funkmodul XBee der Firma Digi International verwendet. Die Verbindung wird im 2,4GHz Bereich hergestellt. Das auf dem Roboter verbaute Modul kommuniziert über eine serielle Schnittstelle mit dem Mikrocontroller. Für die Verbindung mit einem Computer wird ebenfalls ein XBee Funkmodul verwendet, welches per USB angeschlossen wird. Somit können die Berechnungen des Reglers auf dem Computer ausgeführt und die Lenkbefehle wieder zurück an den Roboter gesendet werden. Hierfür wird das Robot Operating System<sup>1</sup> (ROS) verwendet. Die ROS Bibliotheken und Werkzeuge sind für die Verwendung mit linuxbasierten Betriebssystemen entwickelt. Somit lässt es sich nicht direkt auf dem Mikrocontroller installieren [4].

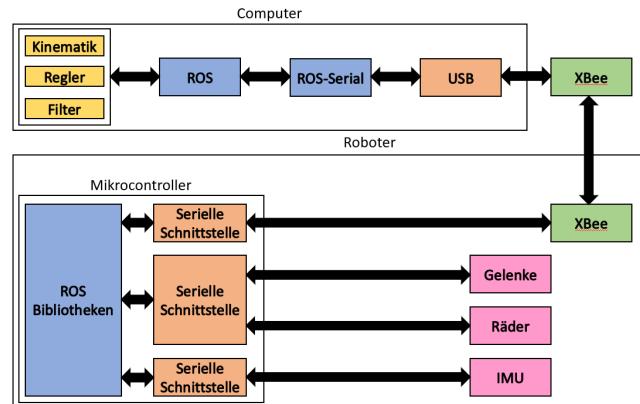


Abb. 2. Kommunikationsstruktur des Roboters. Vom Computer inklusive Kinematik- und Reglermodell, über ROS und XBee Kommunikation und der Ansteuerung von Sensorik und Aktorik auf dem Mikrocontroller [4]

In Abb. 2 wird die Kommunikationsstruktur verdeutlicht. Hier ist zu erkennen, dass das Modell für die Regelung auf dem Computer ausgeführt wird und dessen Ergebnisse unter Verwendung von ROS, ROS-Serial und XBee auf den Mikrocontroller übertragen werden. Dort werden anschließend die passenden Steuerungsbefehle an die Servomotoren ausgegeben [4].

#### E. Simulation und Teststrecke

Zum Testen der implementierten Reglermodelle stehen zwei Testumgebungen bereit. Einerseits eine Simulationsumgebung mit Gazebo<sup>2</sup>, andererseits eine im Rahmen dieser Arbeit aufgebauten realen Teststrecke. Diese beiden Versionen werden in den folgenden Abschnitten kurz vorgestellt.

<sup>1</sup>Robot Operating System: <http://wiki.ros.org/de>

<sup>2</sup>Roboter Simulation mit Gazebo: <http://gazebosim.org/>

*1) Simulation:* Zu Beginn der Projektarbeit wurde eine Simulationsumgebung bereitgestellt, welche den vollständigen Roboter und ein gebogenes Rohr beinhaltet. Diese Simulation ist sehr realitätsnah und kann für schnelles Testen der Regelimplementierungen genutzt werden. Die Simulation beinhaltet, wie das original, elf Motoren (acht für den Antrieb und drei für die Gelenke), und ein passives Gelenk. Ebenfalls wird eine inertiale Messeinheit simuliert. Somit können mit kleinen Anpassungen an die Schnittstelle, dieselben Regleralgorithmen in der Realität, wie auch in der Simulation, getestet werden und dementsprechend Daten aufzeichnen. Dies gewährleistet immer gleichbleibende Bedingungen und eine längere Fahrt als in der Realität.

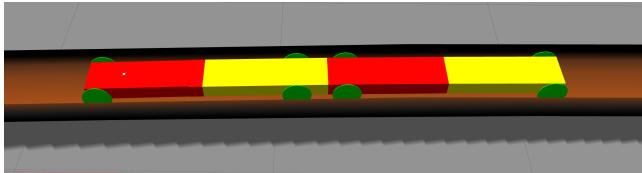


Abb. 3. Roboter fährt geregelt durch ein Rohr in der Simulation.

*2) Teststrecke:* Zum Testen der Regelung steht kein Teil des Originalrohrs aus dem Teilchenbeschleuniger zur Verfügung. Hierbei ist auch zu beachten, dass der Roboter universell in verschiedenen Rohren eingesetzt werden soll und sich somit auch der Rohrradius variiert. Daher ist es nicht wichtig das Rohr exakt nachzubilden. Als Rohrsatz dient ein flexibles Lüftungsrohr, welches längs der Rohrrachse halbiert wurde. Dadurch wird beobachtet, wie sich der Roboter beim Fahren verhält. Zusätzlich ermöglicht das Aufschneiden ein Aufbiegen des Rohres und somit eine Veränderung des Rohrradius. Dieses Setup wurde bereits zu Beginn der Arbeit aufgebaut. Die Teststrecke besteht aus einem geraden Abschnitt am Anfang des Rohres, um zu testen, wie sich der Roboter beim Einschalten verhält, besonders dann, wenn er nicht parallel zur Rohrrachse eingesetzt wird. Danach folgt zuerst eine Rechtskurve, gefolgt von einer Linkskurve, um die Kurvenfahrt zu untersuchen. Dabei wird der Kurvenradius wesentlich enger gewählt als im original um den Roboter unter verschärften Bedingungen testen zu können. Diese Kurven sind zwar mit der Sensorik besser zu detektieren, durch die Größere Auslenkung, sind aber somit auch größere Störgrößen und führen zu größeren Auslenkungen, was den Vergleich zwischen Theorie und Praxis erschwert.

Bedingt durch die Starken Unebenheiten des Lüftungsrohres und dem niedrigen Reibungskoeffizienten, zwischen Rohr und Räder, wurde eine Anti-Rutsch Folie in das Rohr geklebt. Diese verhindert ein Rutschen des Roboters im Rohr.

### III. MODELLBILDUNG

Als Grundlage zur Reglerauslegung dient ein mathematisches Modell, welches die Bewegung des Roboters im Rohr beschreibt. Dieses Modell wird unter der Annahme einiger Vereinfachungen aus der Kinematik des Systems mit anschließender Linearisierung hergeleitet. Im Folgenden wird die Herleitung beschrieben.

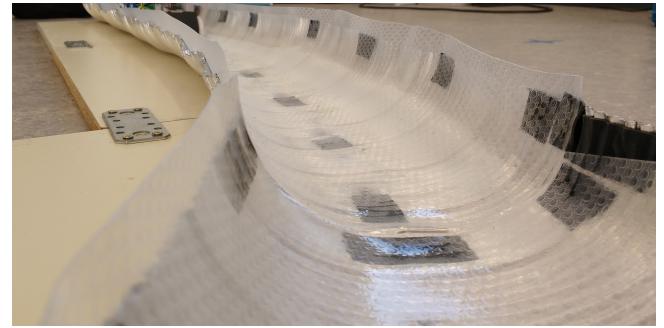


Abb. 4. Teststrecke mit geradem Anteil und erster Kurve. Gut zu sehen ist die eingeklebte Anti-Rutsch-Folie zur Verbesserung der Hafreibung. **WAS FEHLT HIER???**

Zunächst wird das Rohr als ein kreisförmiger Zylinder angenommen. Das Inertialsystem liegt am Beginn des Rohres im Mittelpunkt der kreisförmigen Grundfläche. Die z-Achse zeigt entlang des Rohres (Abbildung 5). Im Teilchenbeschleuniger ist der Rohrquerschnitt ellipsenförmig. Da sich der Roboter aber ausschließlich im unteren Bereich des Rohres aufhält, wird für das Modell eine kreisförmige Querschnittsfläche verwendet. Zudem werden Biegungen entlang der z-Achse vernachlässigt und als Störung angenommen.

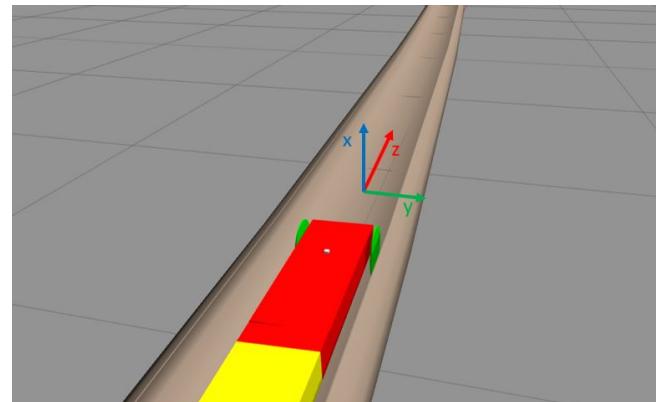


Abb. 5. Rohrkoordinatensystem.

Um die Position des Roboters im Rohr zu beschreiben, gibt es verschiedene Möglichkeiten. Da die Position in z-Richtung für die Regelung nicht relevant ist und sich der Roboter immer auf der Rohroberfläche befindet, genügt eine Variable als Minimalkoordinate um die Position des Roboters zu beschreiben. Abbildung 6 zeigt drei Mögliche Variablen. Die möglichen Variablen sind die Auslenkung in y-Richtung  $y_{Roboter}$ , Die Kreisbogenlänge  $b$  zwischen dem untersten Punkt im Rohr und bis zum Roboter und der Winkel  $\alpha$ .  $\alpha$  ist der Winkel zwischen der x-Achse und einer Verbindungslinie vom Mittelpunkt des Rohrquerschnittes zum Mittelpunkt der vorderen Radachse. Die y-Auslenkung wird nach

$$y_{Roboter} = r_{Rohr} \sin(\alpha) \quad (1)$$

berechnet. Um die dadurch entstehenden Nichtlinearitäten zu vermeiden wird sich gegen diese Variable entschieden. Der Winkel  $\alpha$  entspricht näherungsweise dem Rollwinkel  $\varphi$ , der von der IMU gemessen werden kann (Vergleiche Sektion

IV). Deshalb wird sich im mathematischen Modell für diesen Winkel als Regelgröße entschieden. Ziel der Regelung ist es,  $\alpha$  zu null zu bringen, damit sich der Roboter an unterster Stelle im Rohr befindet. Der Zustand des Roboters wird zusätzlich durch den Gierwinkel beschrieben. Dieser stellt den Winkel zwischen der z-Achse (Rohrachse) und der Bewegungsrichtung des Roboters an.

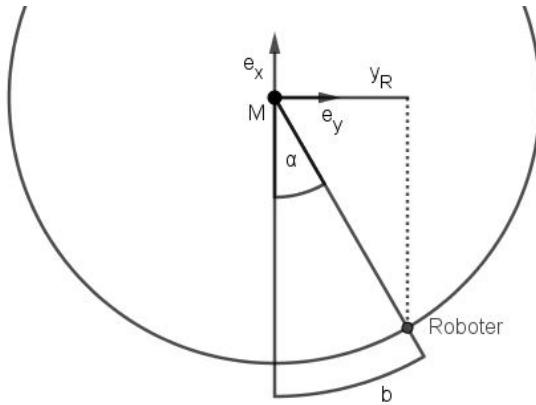


Abb. 6. Mögliche Minimalkoordinaten.

Als Stellgröße für das System kann der Lenkwinkel  $\delta_1$  des Roboters vorgegeben werden. Aus diesem ergibt sich laut [4, S. 6] aus dem Abstand der ersten Radachse zur Spitze des Roboters  $L_{P_1 J_1}$  und der Geschwindigkeit  $v_{Roboter}$  für die Gierrate

$$\dot{\psi} = \frac{\tan(\delta_1)}{L_{P_1 J_1}} v_{Roboter} \quad (2)$$

Durch Integration ergibt sich die Zustandsvariable  $\psi$ . Die Geschwindigkeit des Roboters  $v_{Roboter}$  liegt Konstant bei 0,05 m/s. Im nächsten Schritt wird die Geschwindigkeit in einer Umfangs- und eine Axialkomponente bezogen auf den Rohrradius aufgeteilt (Abbildung ??). Diese berechnen sich nach:

$$v_u = v_{Roboter} \cdot \sin(\psi) \quad (3)$$

$$v_z = v_{Roboter} \cdot \cos(\psi) \quad (4)$$

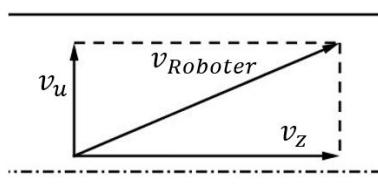


Abb. 7. Aufteilung der Geschwindigkeit

Aus der Umfangskomponente  $v_u$  und dem Rohrradius  $r_{Rohr}$  ergibt sich die Änderungsrate des Winkels  $\alpha$  nach der Formel:

$$\dot{\alpha} = \frac{v_u}{r_{Rohr}} \quad (5)$$

Und somit durch Integration die zweite Zustandsvariable  $\alpha$ , welcher der Regelgröße entspricht.

Zuletzt wird noch die Störung berechnet, die in einer Kurve auf das System wirkt. Dazu wird die Geschwindigkeitskomponente in Axialrichtung  $v_z$  verwendet. Wird angenommen der Roboter fährt geradeaus, so biegt sich die Rohrachse mit

$$\dot{\psi}_{str} = \frac{v_z}{r_{Kurve}} \quad (6)$$

weg von der Bewegungsrichtung des Roboters und wirkt so als Störung auf den Eingang des Systems. Da der Gierwinkel im Bereich von -??? bis +??? liegt, kann der Kosinus in Formel 4 zu eins angenommen werden und somit entspricht  $v_z = v_{Roboter}$ . Der Kurvenradius im Teilchenbeschleuniger  $r_{Kurve}$  beträgt 52632 mm, sodass sich in der Kurve eine sehr geringe Störung von ca  $9,5 \cdot 10^{-4}$  rad/s.

Letztendlich wird auch Formel 3 durch die Kleinwinkeländerung linearisiert. Die Nichtlinearität, die durch den Eingang  $\delta_1$  entsteht, wird vermieden, indem  $\dot{\psi}$  als Eingang verwendet wird und daraus der benötigte Lenkwinkel berechnet wird. Die Formel für  $\delta_1$  ergibt sich durch umstellen von 2 zu

$$\delta_1 = \arctan\left(\dot{\psi} \frac{L_{P_1 J_1}}{v_{Roboter}}\right) \quad (7)$$

Somit ergibt sich das Gesamtmodell in Abbildung 8. Die Übertragungsfunktion lautet:

$$G_{Strecke}(s) = \frac{v_{Roboter}}{r_{Rohr}} \cdot \frac{1}{s^2} \quad (8)$$

Die Halbachsen der Ellipse im Teilchenbeschleuniger betragen 30 mm und 60 mm, Deshalb wird der Rohrradius zu 100 mm angenommen, um den unteren Bereich des elliptischen Rohres anzunähern.

Eine Digitalisierung der Strecke mittels z-Transformation führt zu

$$G_{Strecke}(z) = \frac{1}{T^2} \cdot \frac{r_{Rohr}}{v_{Roboter}} \cdot \frac{z-1}{z+0,5} \quad (9)$$

unter Berücksichtigung des Abtast-Halte-Gliedes.

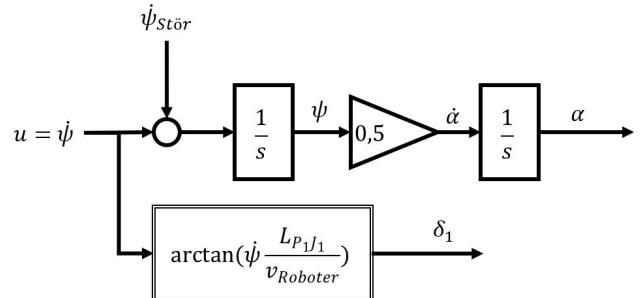


Abb. 8. Blockschaltbild der Regelstrecke

#### IV. MESSUNG

Wie schon in Sektion II beschrieben, verwendet der Roboter eine IMU. Das Koordinatensystem dieser weicht aber von dem Koordinatensystem der Modellbildung ab (Vergleiche Abbildung 1 und Abbildung 5). Ein Komplementärfilter nutzt neben der Integration auch die Anteile der Erdbeschleunigung an

den einzelnen Koordinatenachsen und vergleicht die Werte der Integration der Winkelbeschleunigungen. So wird das Messrauschen verringert und ein Drift der Lagem welcher durch aufsummieren von Messfehlern entsteht, wird verhindert.

Nimmt man an, dass das Gelenk zwischen dem ersten und dem zweiten Modul des Roboters nicht eingeknickt ist, so ergibt sich der Winkel  $\alpha$  aus dem Rollwinkel  $\varphi$  und dem Nickwinkel  $\theta$ . Durch den großen Radabstand zwischen der ersten Achse und der zweiten Achse des Roboters, bleibt der Nickwinkel beim Auffahren im Rohr so gering, dass er nicht messbar ist und nahezu keinen Einfluss auf  $\alpha$  hat. Sodass  $\alpha = \varphi$  gesetzt werden kann.

Da die Werte für die Lage durch den Komplementärfilter an Auflösung verlieren, wird der Rollwinkel  $\varphi$  direkt aus den Anteilen der Erdbeschleunigung auf den einzelnen Achsen berechnet. Hier kann sowohl der Anteil in z-Richtung verwendet werden

$$\varphi = \arccos\left(\frac{a_z}{-g}\right) \quad (10)$$

als auch der Anteil in y-Richtung

$$\varphi = \arcsin\left(\frac{a_y}{-g}\right) \quad (11)$$

Es wird sich für eine Berechnung nach 11 entschieden, da  $a_y$  im Bereich von  $\varphi = 0$  stärker variiert, denn der Sinus ist in diesem Bereich wesentlich steiler, als der Kosinus. Damit wird ein größeres Verhältnis zwischen Messwert und Rauschen geschaffen.

Aus dem Nickwinkel und dem Rollwinkel könnte auch die Zustandsgröße  $\psi$  (Gierwinkel) berechnet werden, solange die IMU nicht parallel zur yz-Ebene ausgerichtet ist. In der yz-Ebene kommt es zu einer Singularität, die die Berechnung verhindert. Allerdings kann diese Berechnung durch den zu kleinen Nickwinkel nicht angewendet werden. Auch die Bestimmung des Gierwinkels durch Integration ist möglich, da als Bezugspunkt die Rohrachsenrichtung dient (Vergleiche Sektion III). Dieser Bezugspunkt ändert sich mit dem unbekannten Verlauf des Rohres. Bei der Integration der Winkelbeschleunigung hingegen bleibt der Bezugspunkt immer gleich, bzw. ist mit einem Drift behaftet.

Die Position des Sensors ist oberhalb der ersten Radachse. Da der Roboter sehr steig ist für Torsion um seine Achse in Bewegungsrichtung (rote Achse in Abbildung 9), bleibt der Rollwinkel des vorderen Moduls näherungsweise gleich zu den anderen Modulen. Dies hat zur Folge, dass das erste Modul sich nicht verdreht und kein Rollwinkel gemessen werden kann. Fährt der Roboter beispielsweise an der rechten Rohrwand auf, so verliert das linke Rad den Bodenkontakt und der Rollwinkel bleibt näherungsweise gleich zu dem aktuellen Rollwinkel der restlichen Module, die sich noch in unteren Teil des Rohres befinden. Die Regelabweichung ist also nicht messbar. Daher wird ein passives Gelenk in das erste Modul eingefügt, dass eine Rollbewegung unabhängig von den anderen Modulen zulässt.

## V. REGLER

Zur Regelung des Roboters werden verschiedene Regelungskonzepte und Auslegungsmethoden verwendet. Zum

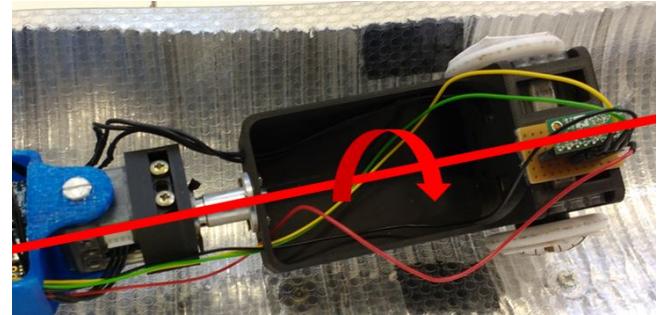


Abb. 9. Passives Gelenk zur Bestimmung des Rollwinkels

einen wird eine Ausgangsregelung in Form eines proportional-differential Reglers (PD-Regler) umgesetzt, der mittels dem Wurzelortskurvenverfahren ausgelegt wird. Zum anderen wird ein Zustandsregler verwendet, dessen Parameter nach zwei unterschiedlichen Verfahren bestimmt werden. Bei den Verfahren handelt es sich um Polplatzierung und die linear-quadratische Regelung (LQR). Da die Zustände nicht gemessen werden können, werden zusätzlich noch Beobachter verwendet. Dabei handelt es sich zum einen um einen Luenberger-Beobachter in Kombination mit einem Filter und zum anderen um ein Kalman-Filter. In den folgenden Unterkapiteln werden diese Konzepte erläutert.

Für den PD-Regler ist auch eine Filterung der Messgröße erforderlich. Hierzu wird derselbe Filter wie für den Luenberger-Beobachter verwendet. Es handelt sich um einen Moving-Average aus den letzten 20 gemessenen Werten.

Die Stellgrößenbeschränkung ergibt sich durch die mangelnde Bewegungsfreiheit für den Roboter im Rohr und durch den maximalen Winkel, der an den Gelenken eingestellt werden darf. Wobei erstere wesentlich geringere Stellgrößen zulässt. Zur Festlegung dieses Wertes wird der Roboter per Hand durch die Teststrecke gesteuert und immer wieder dem maximalen Lenkwinkel ausgesetzt. Dieser wird per Durchlauf immer weiter erhöht. Wird der maximale Lenkwinkel zu groß, so kann es passieren, dass einzelne Module gegen die Rohrwand drücken, da die nötige Bewegungsfreiheit nicht gegeben ist. Daher wird der maximale Lenkwinkel auf  $4^\circ$  gesetzt. So wird verhindert, dass der eben beschriebene Fall eintritt und dadurch hohe Belastungen auf die Gelenke wirken. Damit sollen durch diese Stellgrößenbeschränkung auch Schäden am Roboter vermieden werden. Für den Regler ergibt sich so nach Formel 2 eine maximale Stellgröße von  $0,233 \text{ rad/s}$ .

### A. PD-Regler

Für eine Ausgangsregelung muss nur die Regelgröße eines Systems gemessen werden. Da der Gierwinkel, der eine Zustandsvariable des Systems darstellt, nicht gemessen werden kann (vgl. Sektion III), bietet sich diese Regelmethode für das vorliegende System an. Durch die Linearisierung kann auf eine Großzahl an linearen Auslegemethoden zurückgegriffen werden. Letztendlich wird sich hier für das Wurzelortskurvenverfahren, da so die Dynamik des Systems besonders gut eingestellt werden kann. [7] Die Regelstrecke weist einen doppelten Pol in null auf und ist somit instabil. Deshalb

wird ein differentieller Anteil (D-Anteil) im Regler benötigt, der das System stabilisiert. Die Störgröße, die durch die Kurvenfahrt entsteht, greift noch vor dem ersten Integrator der Regelstrecke an (Abbildung 8). Daher ist ein integraler Anteil (I-Anteil) im Regler nötig um diese Störung stationär genau auszuregeln. Allerdings ist die Störung sehr klein gegenüber anderer Systemgrößen (vgl. Sektion IV), sodass der I-Anteil nicht benötigt wird.

Bei der Auslegung mittels Wurzelortskurve (WOK) ist zu beachten, dass der Regler so eingestellt wird, dass die Stellgrößenbeschränkung eingehalten wird. Zudem darf der D-Anteil nicht zu groß gewählt werden, da dieser das Messrauschen verstärkt. Dadurch schwankt die Stellgröße zwischen ihren maximalen- und minimalen Wert und es ist keine Reaktion auf die Regelabweichung möglich. Je größer der Filter gewählt wird, desto größer kann auch der D-Anteil gewählt werden. Allerdings führt der Filter auch zu einer Zeitverzögerung, die mit der Filtergröße ansteigt. Daher darf auch der Filter nicht zu groß gewählt werden. Bei einer Filtergröße von 20 Werten, über die der Mittelwert berechnet wird, ist die Zeitverzögerung so gering, dass sie gegenüber den Zeitkonstanten des Systems vernachlässigt werden kann. Der Verstärkungsfaktor für den D-Anteil kann in diesem Fall bis ca. 4 erhöht werden, ohne dass diese Stellgröße große Sprünge aufweist.

Letztendlich ergibt sich der Regler zu

$$G_r(z) = \frac{279,4z - 276,6z}{z - 0,3116} \quad (12)$$

### B. Zustandsregler mittels Polplatzierung

Bei der Auslegung eines Zustandsregler mittels Polplatzierung sind die Pole des geschlossenen Regelkreises frei wählbar, was dazu führt, dass die Dynamik des Regelkreises frei bestimbar ist. Dazu werden allerdings auch mehr Informationen über den aktuellen Systemzustand benötigt. So werden wie in Abbildung 10 zu sehen alle Zustandsgrößen zurückgeführt und mit dem Regelvektor  $\underline{k}^T$  verstärkt. Die Stellgröße ergibt sich noch zusätzlich nach [7] aus der Vorverstärkung  $\nu$  zu:

$$u = \nu\omega - \underline{k}^T \underline{x} \quad (13)$$

Da die Sollgröße  $\omega$  allerdings null ist, entfällt der erste Teil der Gleichung und  $u$  berechnet sich nur aus  $\underline{k}^T \underline{x}$

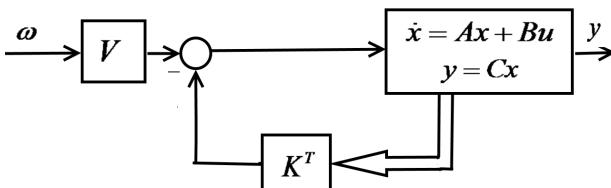


Abb. 10. Blockschaltbild eines Zustandsregler [7]

Um einen Zustandsregler zu benutzen muss noch die Steuerbarkeit nachgewiesen werden. Dafür muss die Steuerbarkeitsmatrix  $M_S$  nach [7] vollen Rang haben, beziehungsweise eine Determinante ungleich null:

$$\det(M_S) = |b \quad A \underline{b}| = \begin{vmatrix} 0 & \frac{v_{Roboter}}{r_{Rohr}} \\ 1 & 0 \end{vmatrix} = -\frac{v_{Roboter}}{r_{Rohr}} \neq 0 \quad (14)$$

Jetzt können die Eigenwerte des Systems unter der Berücksichtigung der Stellgrößenbeschränkung frei gewählt werden und daraus der Regelvektor  $\underline{k}^T$  berechnet werden. Die Wunschkopole werden beide auf -1 gelegt. Damit ergibt sich  $\underline{k}^T$  zu  $[0, 5 \quad 2]$ . (Vgl. [7])

Letztlich müsste für den Zustandsregler noch ein I-Anteil vorgeschaltet werden, um die Störung stationär genau auszuregeln. Hier gilt allerdings dasselbe wie bei dem PD-Regler in Sektion V-A. Die Störung ist so gering, dass der I-Anteil nicht benötigt wird.

### C. Zustandsregler mittels LQR

Die LQR ist eine weitere Auslegungsmethode für Zustandsregler. Dafür wird ein Gütemaß für die Regelung berechnet, welches anschließend durch Bestimmung des Regelvektors  $\underline{k}^T$  optimiert wird. Bei dem Gütemaß werden die Zustände quadriert und mit der Matrix  $Q$  gewichtet. Zudem wird auch die Stellgröße quadriert und mit  $R$  gewichtet. So hat man im Gütemaß eine Größe, mit der die später im Regler verwendete Stellgröße direkt bewerten kann und so auch bei der Auslegung direkt beeinflussen kann. Der weiteren werden durch das Quadrieren große Abweichungen von der Ruhelage besonders schlecht bewertet, was auch eine gute Beeinflussung des Zeitverhaltens mit sich bringt. Das quadratische Gütemaß berechnet sich nach [7] mit der Formel:

$$J = \int_0^\infty \underline{x}^T \underline{Q} \underline{x} + Ru^2 \quad (15)$$

Hierbei ist darauf zu achten, dass die Ruhelage in den Ursprung transformiert wird, sodass das Integral konvergiert. Mit Werten für  $Q$  gleich Einheitsmatrix  $I$  und 200 für  $R$  ergibt sich der Regelvektor  $\underline{k}^T$  zu  $[0, 2752 \quad 0, 0707]$ .

### D. Vollständiger Luenberger-Beobachter

Um die Zustandsregler zu benutzen, ist es nötig, dass der komplette Zustandsvektor zur Verfügung steht. Da wie bereits in Sektion IV erwähnt, die Zustandsgröße  $\psi$  nicht gemessen werden kann, muss der Zustandsvektor mittels eines Beobachters bestimmt werden. Der Beobachter besteht aus dem in Sektion III beschriebenen Modell der realen Regelstrecke. Dieses Modell wird mit denselben Stellgrößen wie die reale Strecke beaufschlagt und berechnet so den Zustandsvektor. Um Fehler durch Modellgenauigkeiten, Störungen und Anfangsbedingungen zu vermeiden, werden die Ausgangsgrößen der realen Strecke und des Modells miteinander verglichen und der Fehler über den Gewichtungsfaktor  $\underline{l}$  zurückgeführt. Abbildung 11 veranschaulicht den Beobachter, wobei die mit Tilde gekennzeichneten Werte, die vom Beobachter berechnet werden.

Über den Vektor  $\underline{l}$  können die Eigenwerte des Beobachters festgelegt werden. Diese sollen laut Faustformel [7] das 5-10 fache weiter links in der komplexen Ebene liegen als die dominanten Eigenwerte des geschlossenen Regelkreise, damit mögliche Fehler vom Beobachter schnell gegen null konvergieren. Allerdings dürfen die Pole auch nicht zu weit links liegen, da der Beobachter sonst differenzierendes Verhalten aufweist. Dies ist besonders bei verrauschten Messsignalen

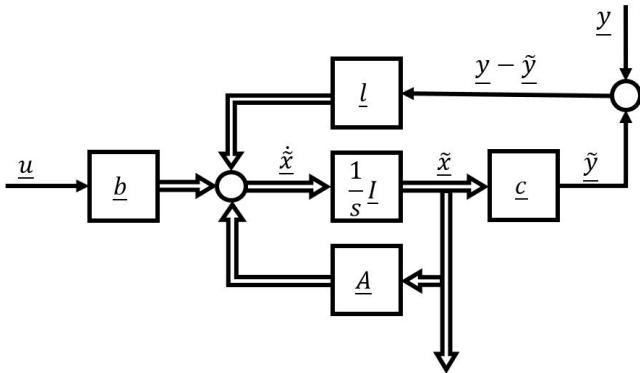


Abb. 11. Luenberger-Beobachter [7]

problematisch, da so das Rauschen verstrt wird. Bei dem in Sektion V-A beschriebenen Filter knnen die Beobachterpole nicht weiter links liegen als bei -5 in der komplexen Ebene. Damit sind sie 5 mal schneller als der Regelkreis, der mittels Polplatzierung ausgelegt wird. Der Vektor  $\underline{l}$  wird ber das Verfahren der Polplatzierung berechnet und ergibt sich zu  $[10 \quad 50]$ .

#### E. Reduzierter Beobachter

Der reduzierte Beobachter berechnet nicht den kompletten Zustandsvektor. Er berechnet nur die Zustandsgren, die nicht gemessen werden knnen. Dazu benutzt er neben der Stellgre auch die gemessenen Zustandsgren, um zum einen den Schtzfehler zu minimieren und zum anderen mglichst viel Information ber den Systemzustand zu nutzen. In diesem Fall wird die Zustandsgre  $\psi$  geschztzt, wrend die Auslenkung  $\alpha$  gemessen werden kann. Allerdings macht das Rauschen hier Probleme, sodass die Ergebnisse, die mit dem vollstndigen Luenberger-Beobachter erzielt werden, besser sind.

#### F. Diskretes Kalman-Filter

Fr den Entwurf von Beobachtern wird davon ausgegangen, dass Zustands- und Ausgangsvariablen keinem Rauschen unterlagen. Da das Rauschen klein ist, wurde es bisher vernachlssigt. Wenn jedoch Systemrauschen und Messrauschen vorliegen, kann man mittels eines Kalman-Filters aus den verrauschten Ausgen und den Eingangsgren den Zustandsvektor rekonstruieren. Das Kalman-Filter besitzt also die gleiche Struktur wie ein Luenberger-Beobachter. Auch hier wird der Schtzfehler  $e = x - \hat{x}$  mittels der Beobachtermatrix  $K$  so beeinflusst, dass er asymptotisch gegen Null strebt. Die Matrix  $K$  soll so bestimmt werden, dass die Rauscheinflse  $\mu n_{x,k}$  und  $n_{y,k}$  auf den geschtzten Zustandsvektor  $\hat{x}$  minimiert werden.

Da die Bewegung des Fahrzeugs ebenfalls gestrt (im Sinne von berlagertem, normalverteiltem Rauschen) ist, wird dies ber die Prozess-Rausch-Kovarianzmatrix  $Q$  eingebracht. Diese Matrix teilt dem Filter mit, wie der Systemzustand von einem zum nchsten Schritt springen kann. Die Matrix  $Q$  ist Kovarianzsystemrauschen und kann nicht praktisch ermittelt werden. Wie blich wird  $Q$  hier als Einheitsmatrix gewählt.

$R$  ist die Messkovarianzmatrix und gibt an, wie sehr man den Messwerten der Sensoren vertrauen kann. Da wir nur  $\alpha$  messen, ist dies ein Skalar. Ist der Sensor sehr genau, so sollten hier kleine Werte eingesetzt werden. Ist der Sensor relativ ungenau, so sollten hier groe Werte eingesetzt werden.

Es wird das zeitdiskrete System

$$\begin{aligned} x_{k+1} &= A_k + B_k u_k + n_{x,k} \\ \hat{y}_k &= C_k x_k + D_k u_k + n_{y,k} \end{aligned}$$

betrachtet.

Das Systemrauschen  $n_{x,k}$  und das Messrauschen  $n_{y,k}$  sind jeweils mittelwertfrei, in sich unkorreliert und besitzen die Kovarianzmatrizen.

Das Systemrauschen beschreibt tatschliche nderungen des Zustandsvektors. Idealerweise wrde der Schtzwert diesem Rauschen daher folgen. Das Messrauschen beschreibt Strungen, die der Schtzwert idealerweise ignorieren wrde.

$$\begin{aligned} Q_k &= E(n_{x,k} n_{x,k}^T) & E(n_{x,k}) = 0 & E(n_{x,k} n_{x,l}^T) = 0 & \forall k \neq l \\ R_k &= E(n_{y,k} n_{y,k}^T) & E(n_{y,k}) = 0 & E(n_{y,k} n_{y,l}^T) = 0 & \forall k \neq l \end{aligned}$$

Initialschtzwert:

$$\begin{aligned} E(\hat{x}_0) &= E(x_0) \\ Cov(x_0 - \hat{x}_0, x_0 - \hat{x}_0) &= p_0 \end{aligned}$$

Praediktionsschritt: Dieser Teil des Kalman-Filter wagt nun eine Vorhersage des Systemzustands in die Zukunft.

$$\begin{aligned} \hat{x}_{k+1,-} &= A_k \hat{x}_{k,+} + B_k u_k \\ p_{k+1,-} &= A_k p_{k,+} A_k^T + Q_k \end{aligned}$$

Korrekturschritt: Liegen neue Messdaten vor, so besteht die Aufgabe darin, ausgehend von dem geschtzten Systemzustand  $\hat{x}_{k,-}$ , mit der bekannten Kovarianz  $p_{\Delta,k,-}$ , des Schtzfehlers und den Messwerten  $\hat{y}_k$  mit der  $p_{y,k}$  einen neuen Schtzwert  $\hat{x}_{k,+}$  zu bestimmen, der biasfrei ist und dabei die minimale Varianz besitzt. ( vgl. Abbildung 12)

$$\begin{aligned} k_k &= p_{k,-} c_k^T \cdot (c_k p_{k,-} c_k^T + R_k)^{-1} \\ \hat{x}_{k,+} &= \hat{x}_{k,-} + k_k \cdot (\hat{y}_k - (c_k \hat{x}_{k,-} + D_k u_k)) \\ p_{k,+} &= (I - k_k c_k) \cdot p_{k,-} \end{aligned}$$

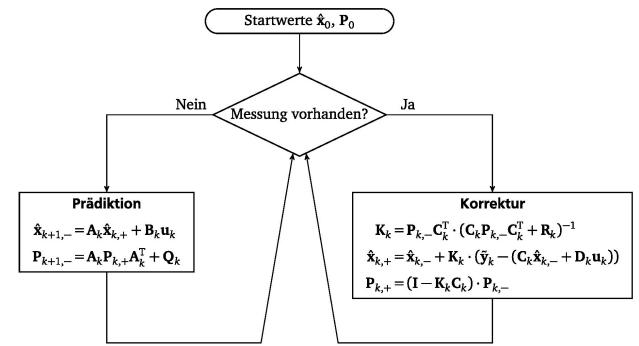


Abb. 12. Kalman Filter

TABELLE I  
REALE TESTFAHRDEN

Regler	Maximalwert	Bleibende Regelabweichung
PD KP=4 KD=0,5	15,9°	-0,9°
Zustandsregler	16,9°	0,5°
Kalman-Filter	17,3°	-4,8°

## VI. BEWERTUNG

In dieser Sektion werden die Messergebnisse mit den entworfenen Reglern bewertet und miteinander verglichen. Dabei ist bei den realen Messwerten zu beachten, dass die reale Teststrecke wesentlich engere Kurven hat als die originale Strecke im Teilchenbeschleuniger. Daher fallen auch die Testergebnisse wesentlich schlechter aus. Dies liegt vor allem daran, dass die engeren Kurven eine größere Störung mit sich bringen. Dennoch kann die Teststrecke mit jedem Regler vollständig durchfahren werden, ohne dass der Roboter umkippt oder stecken bleibt.

Zur Bewertung werden die Sensormesswerte für den Rollwinkel ( $\varphi$ ), die mit dem Average-Filter geglättet werden, verwendet. Die Glättung erfolgt durch das Entfernen höherer Frequenzanteile, was zur Reduzierung des Rauscheinflusses dient. Im Ergebnis wird eine neue Datenpunktmenge erstellt, die aus den Mittelwerten gleich großer Untermengen der ursprünglichen Datenpunktmenge besteht. In der Signaltheorie wird der Moving-Average (gleitender Mittelwert) als Tiefpassfilter mit endlicher Impulsantwort (FIR-Tiefpass) beschrieben. Die Filterlänge des Average Filters wird bei den Tests der realen Strecke 10 und in der Simulation 20 gewählt. Da in der Testaufbau in der Realität zwei wesentlich engere Kurven enthält als die Simulation, wirkt sich die zeitliche Verzögerung durch den Filter in der Simulation nicht so stark aus. Dadurch wird das Rauchen in der Simulation besser unterdrückt und in der Realität lenkt der Roboter früher ein.

In Matlab werden die resultierenden Werte für  $\alpha$  (vgl. Sektion III) ebenfalls gefiltert um den Einfluss des Rauschens zu reduzieren, da sonst der Maximalwert zu stark vom Rauschen abhängt. Je größer die Filterlänge ist, desto besser wird das Rauschen unterdrückt. Allerdings werden auch schnelle Änderungen des Messwertes geglättet. Deshalb muss eine geeignete Filterlänge iterativ gewählt werden. Als Filtergröße in Matlab wird, unter Beachtung der Stellgrößenbeschränkung, 2800 gewählt.

Überschwingweite und Ausregelzeit lassen sich nicht bestimmen, da der Roboter bei allen Reglern, außer beim Kalman-Filter, bei  $\alpha = 0$  startet und nicht viel ausregeln muss. Aus diesem Grund wird das Störverhalten in der Kurve untersucht. Dabei werden die maximale Abweichung vom Sollwert Null sowie der Durchschnittswert der bleibenden Regelabweichung in der Kurve nach dem Einschwingen bestimmt.

Für die bleibende Regelabweichung wurde der Mittelwert von den Werten nach dem Einschwingen in der Kurve gebildet. Dabei werden die Messwerte mit dem Average-Filter, jedoch ohne Matlab Filter, verwendet, da der Mittelwert direkt aus diesen Werten bestimmt werden kann. Bei der Aufnahme

der Sensormesswerte in ROS wird zwar der Messprozess zu Beginn der Regelung gestartet, aber nach Durchfahren des Rohres manuell mittels Tastatureingabe gestoppt. Dadurch haben die Messungen eine variable Länge. Aus diesem Grund werden bei der Berechnung der bleibenden Regelabweichung 50 Werte nach dem Einschwingen nach der Störung, der Krüve, verwendet.

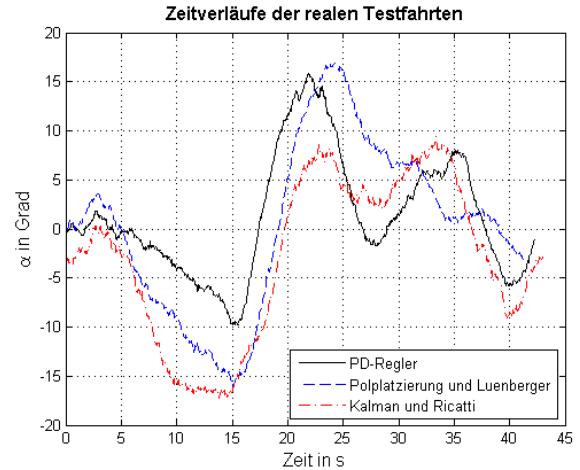


Abb. 13. Kalman-Filter

Die kumulierte Häufigkeitsverteilung in Abbildung 14 fasst alle aufgenommenen Werte aus den Testverfahren zusammen. Sie gibt an, wie häufig, bezogen auf die gesamte Simulationsdauer, die auf der x-Achse aufgetragenen Abweichungen vom Sollwert unterschritten werden. Je höher der Graph also verläuft, desto besser ist der entsprechende Regler zu bewerten. Es ist zu sehen, dass der PD-Regler hier am besten abschneidet. Des Weiteren ist zu erkennen, dass gerade sehr große Abweichungen nur selten überschritten werden. So wird ein  $\alpha$  von  $14^\circ$  nur in 4% der Fahrzeit überschritten. Bei den anderen beiden Reglern sind es circa 15%.

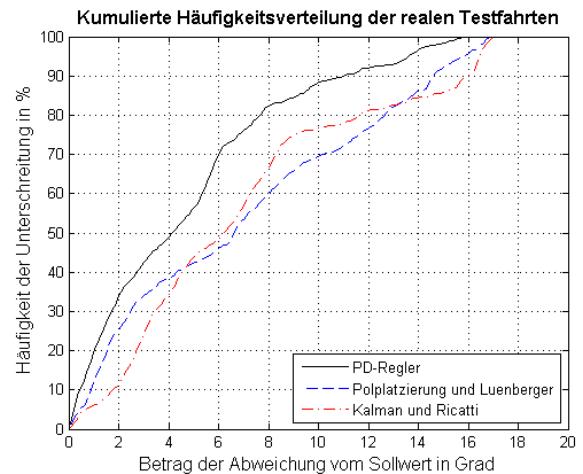


Abb. 14. Kumulierte Häufigkeit für Kalman-Filter, PD-Regler und Zustandsregler

Da die Verläufe der Simulation nicht angemessen gefiltert und geplottet werden können, wird hier darauf verzichtet.

TABELLE II  
SIMULATION

Regler	Maximalwert	Bleibende Regelabweichung
PD KP=4 KD=0,5	1,9°	-0,1°
Zustandsregler	1,3°	-1,9°
Kalman-Filter	0,8°	-1,3°

Die Abweichungen vom Wert Null sind sehr klein und die Störungen werden schnell ausgeregelt. Dies kann auf den größeren Kurvenradius in der Simulation zurückgeführt werden. Bezuglich der maximalen Abweichung liefert das Kalman-Filter das beste Ergebnis. Dann kommt der Zustandsregler gefolgt vom PD-Regler. Alle Abweichungen sind in einem der Stellgrößenbeschränkung. Bezuglich der Regelabweichung liefert der PD-Regler das beste Ergebnis, gefolgt vom Kalman-Filter und dem Zustandsregler. Diese überschreiten die Stellgrößenbeschränkung ebenfalls nicht. Die Simulationsergebnisse sind deutlich besser, als die Ergebnisse der realen Strecke. Es kann dafür mehrere Gründe geben. Zum einen ist in der Simulation eine Kurve mit sehr großem Radius, also einer ganz leichten Krümmung, hinterlegt. Die reale Strecke besteht jedoch aus zwei hintereinander folgenden Kurven kleinerer Radien, wodurch das Ausregeln der Störung erschwert wird. Zum anderen ist der Reibfaktor zwischen den Rädern des Roboters und der Rohroberfläche in der Simulation nicht realitätsgerecht gesetzt, beziehungsweise verhält es sich aktuell nicht so.

Alles in allem sind die erzielten Ergebnisse mit unterschiedlichen Reglern als gut zu bewerten.

## VII. ZUSAMMENFASSUNG UND AUSBLICK

Abschließend lässt sich sagen, dass der Roboter ohne zu kippen oder stecken zu bleiben das Rohr auf der realen Teststrecke und in der Simulation durchfahren kann. Damit wird das Ziel der Arbeit erreicht. Bevor die Regler ausgelegt werden konnten, wurde noch ein großer Fokus auf die Entwicklung eines geeigneten Modells des Systems Schlangenroboter im Rohr gelegt. Zudem wurde sich mit der Wahl und Bestimmung einer geeigneten Regelgröße befasst. So könnte zusätzlich ein linearisiertes Modell erarbeitet werden, was die Reglerauslegung stark unterstützt hat.

Weitere Verbesserungen können erzielt werden, indem weitere Informationen über das Rohr erfasst werden. So kann zum Beispiel eine Erkennung der Kurve, die durch sie entstehende Störung kompensiert werden. Weiterhin ist eine Bestimmung des Gierwinkels  $\psi$  des Roboters sinnvoll, da so alle Zustandsgrößen bekannt sind und ein Zustandsregler ohne Beobachter entworfen werden kann.

Auf Hardwareseite wäre eine Verbesserung der Kommunikationsschnittstelle zwischen Roboter und Computer wünschenswert. Bei einer Ausführungsrate von 100 Hz stößt die Datenübertragung für die Sensormesswerte und Ansteuerung der Aktorik schon an ihre Grenzen. Eine sichere Datenübertragung ist nur sicher zu stellen, indem man auf unnötigen Overhead achtet und diesen minimiert. Ebenso würde eine neue und hinreichende Kalibrierung der IMU nochmal eine

Verringerung des Messrauschens bringen. Die bisher eingesetzte Kalibrierung bringt nicht den gewünschten Erfolg und wurde daher im Rahmen dieses Projektes nicht weiter berücksichtigt.

## LITERATURVERZEICHNIS

- [1] A. Burkert, "Roboter übernehmen die Welt der Produktion," 2017. [Online]. Available: <https://www.springerprofessional.de/industrieroboter/industrie-4-0/roboter-uebernehmen-die-welt-der-produktion/15236792>
- [2] M. Kuther, "Diese Roboter prägen unsere Zukunft," 2018. [Online]. Available: <https://www.elektrotechnik.vogel.de/diese-roboter-praelegen-unsere-zukunft-a-733751/>
- [3] ROBOTIS, "XL-320 e-Manual." [Online]. Available: <http://emanual.robotis.com/docs/en/dxl/x/xl320/>
- [4] Z. Pang, S. Raab, and J. Tamm-morschel, "Validierung eines kinematischen n -Trailer Modells für die Steuerung eines Schlangenroboters," pp. 1–10, 2018.
- [5] Life.augmented, "iNEMO LSM6DS33 Datasheet," pp. 1–77. [Online]. Available: <https://www.st.com/resource/en/datasheet/lsm6ds33.pdf>
- [6] S. Imer, M. Kempkes, S. R. Klein, and M. Schröder, "Kalibrierung einer Inertial Measurement Unit (IMU) für ein UAV," pp. 1–11, 2016.
- [7] J. Adamy, *Systemdynamik und Regelungstechnik II*, 5th ed. Aachen: Shaker Verlag, 2016.



**Jens Balze** hat seinen Bachelor Abschluss in Informatik an der TU Darmstadt gemacht und befindet sich derzeit im Master Studium mit der Vertiefung Autonome Systeme. Diese Fachrichtung hat er gewählt wegen seiner Vorliebe für Robotik und Kraftfahrzeugtechnik. Deshalb war er auch ein Jahr Mitglied beim TU Darmstadt Racing Team e.V. und hat am Formula Student Driverless Fahrzeug mitgearbeitet. Des Weiteren ist er als Werksstudent im Bereich der Messtechnik und Datenanalyse tätig.



**Nursura Demircan** hat ihren Bachelor Abschluss in Mechatronik an der TU Darmstadt gemacht und befindet sich derzeit im Master Studium mit der Vertiefung Allgemeine Mechatronik. Sie hat ein Praktikum in der Türkei in den Bereichen Entwicklung, Produktion und Qualitätssicherung gemacht. Im Weiteren hat sie das Praktikum Regelungstechnik I betreut. Zur Zeit ist sie als Werkstudentin in den Bereichen der Hard- und Softwareentwicklung tätig.



**Elias Jargon** studiert Mechatronik an der TU Darmstadt. Er hat seine Bachelorarbeit 2018 am Fachgebiet Fahrzeugtechnik im Bereich Fahrsimulation geschrieben. Zurzeit befindet er sich im Masterstudium in der Vertiefungsrichtung der Allgemeinen Mechatronik, in der er die Schwerpunkte auf Regelungstechnik und Robotik legt.



**Seyed Davood Mousavi** hat seinen Bachelor Abschluss in Electrical Engineering im Fachgebiet Electronic an der Teheran Polytechnic University gemacht. Das Interesse für Mathematik und Robotik brachte ihn an die TU Darmstadt, wo er seinen Master in Automatisierungstechnik macht.