

Bau eines autonomen Spielfeldzeichners

Ali Kharnoub , ETIT

Otto-von-Guericke-Universität Magdeburg

Abstract

Noch nie haben die Extreme von Zeitnot und Langeweile das Leben so stark geprägt, wie in unserer beschleunigten Gesellschaft. Wie viel Tempo verträgt der Mensch? Selbständige Roboter können Hilfe leisten und eine sinnvolle Ergänzung sein.

In der modernen Industrie werden Prozesse automatisiert, um sie von Fehlern zu bewahren. Anlagen sollen 24 Stunden laufen können, effizient, effektiv und sicher arbeiten können. Dabei kommt es entscheidend darauf an, vorgegebene Werte einzuhalten, um eine qualitativ hochwertige, umweltgerechte und sichere Produktion zu gewährleisten.

Die Programmierungsmöglichkeiten erlauben in der Folgezeit genau das, was man sich als Ziel setzt. z.B. die komplexe Montage innerhalb der Automobilfertigung zu erfüllen.

Nicht nur in der Industrie werden wir immer mehr Robotertechnik in den nächsten Jahren erleben, sondern auch im privaten Bereich von der Waschmaschine bis zum Staubsaugroboter.

Dienstleistungen, Pflege, Zustelldienste und alle Routinetätigkeiten werden künftig Maschinen erledigen. Eine mögliche Anwendung ist das Zeichnen von Spielfeldern in Stadien oder Sporthallen.

Diese Routinearbeit kostet Zeit, Geld und kann anstrengend sein aber braucht keine menschliche Intelligenz, weil die benötigten Informationen und Messungen schon angegeben sind.

Deswegen können Spielfeldbegrenzungen von Automaten gezeichnet werden. Außerdem ist der Ersatz eines Roboters in diesem Fall effizienter, schneller und günstiger.

Im Zeitraum von 12. Februar bis 23. Februar 2018 wollten wir solche Maschine Bauen.

Das Projekt wurde im Rahmen des LEGO-Mindstorms Seminars bearbeitet und das Ziel war, einen Roboter zu konstruieren bzw. Programmieren.

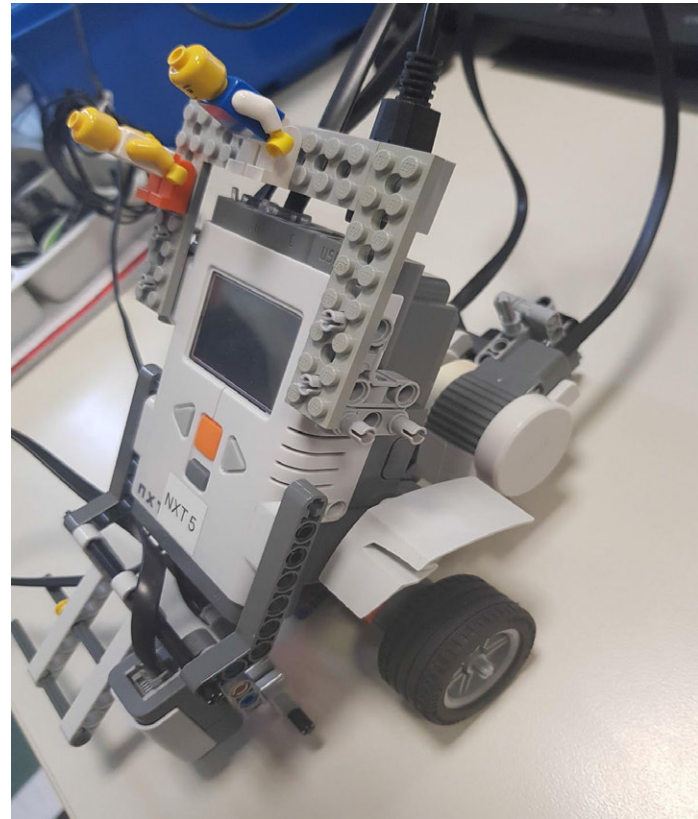
Der Roboter sollte eine schwarze Linie auf weißen Untergrund abfahren und in einer bestimmten Weise sie speichern können. Danach kann er diese Linie reproduzieren.

Außerdem sollte auf den Roboter ein Stift fixiert werden, um die Linie nachzeichnen zu können.

Der Aufbau erfolgt mit Lego unter Verwendung von NXT-Baustein. Mithilfe des von der RWTH Aachen

entwickelten Toolkits wird der Roboter in Matlab programmiert.

Die Anwendung dieses Toolkits unterstützt den direkten Zugriff auf die Sensoren und Motoren unter Matlab. Sempel aber fein, so war der Aufbau.



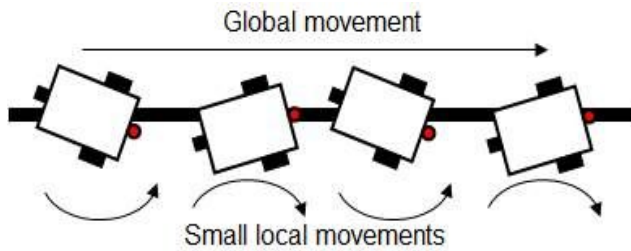
Die Unterschicht besteht aus zwei Motoren, die nebeneinander angebracht wurden und auf jeden Motor ist ein Rad angewiesen.

Zur Unterstützung von Stabilität und Flexibilität wurde ein drittes freies Rad verbunden, das sich um 180 Grad drehen kann. Auf der Unterschicht ist der NXT-Baustein festgelegt und da vorne wird der Farbsensor fixiert.

Die Stifthaltung war ein umstrittenes Thema.

Der Stift sollte entweder neben dem Farbsensor oder in der Mitte festgestellt werden. Zwar wäre die Fixierung in der Mitte besser, weil dieser Punkt im Zentrum liegt. dies hätte die Stabilität unterstützt und hätte zur genaueren Nachzeichnung der Linie aber das war sehr aufwendig und könnte Störungen wegen der zusätzlichen Reibung bewirken.

Das Grundkonzept eines Linienfolgers stellt sich wie folgt dar: Der Roboter folgt nicht der Linie selbst, sondern ihrer Grenze und fährt in Form von Zickzack. d.h. statt einer globalen Bewegung wird er in kleine lokale Bewegungen gesetzt.



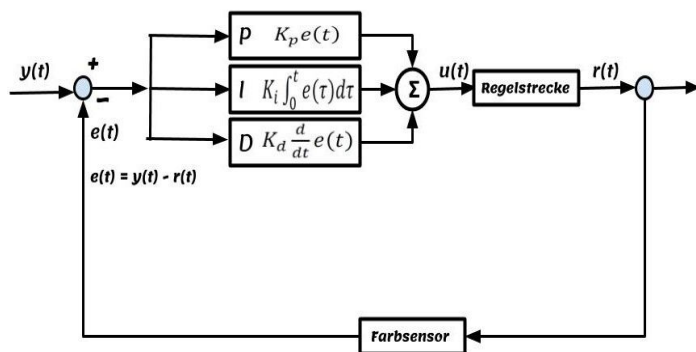
Zuerst erkennt der Sensor den Farbwert und je nach dem eingelesenen Farbwert wird die Spannung der Motoren eingesetzt.

Zwar kann die Bewegung durch diesen Prozess theoretisch korrigiert werden aber es reicht nicht aus, die Bewegung realisieren zu können, da solches Verfahren wie eine einfache boolesche Variable wirkt und Ungenauigkeit bzw. Fehlern beim folgen der Linie, besonders bei geraden und rechteckigen Strecken verursacht. Zur Vermeidung der Fehlern in diesem Prozess, ist es sinnvoller, die Motorsteuerung durch effizienteren Regler zu bestimmen und aus diesem Grund besteht der Bedarf an der Regelungstechnik.

Die Regelungstechnik automatisiert verschiedene Vorgänge so, dass eine gewünschte Betriebsart gestoppt wird oder erhalten bleibt.

Die Regelungstechnik versteckt sich in vielen technischen und natürlichen Systemen und beschäftigt sich mit dem Problem, dass bei den meisten Systemen der gewünschte Einfluss des Eingriffs auf die Stellgröße nicht direkt erzielt werden kann, sondern einen Zeitverzug zwischen dem Eingriff und seinen bemerkbaren Einflüssen auftritt, dabei dient ein Regelkreis dazu, die Regelgröße auf die gewünschte Führungsgröße zu bringen und auf dieser Ebene zu halten.

Bei der Verwendung des Regelkreises im Linienfolger ist das Ziel, die Regeldifferenz zwischen der vorgegebenen Fahrspur (Sollwert) und dem eingelesenen Farbwert (Istwert) auf Null zu setzen.



PID Regelkreis

Die Regelstrecke ist die Störgröße, die vom Regler ausgeregelt werden soll.

Es existieren bereits viele unterschiedliche Regler mit

verschiedenen Art und Funktionsweise. Einer davon ist der in unserem Projekt verwendeten PID-Regler, der aus drei Glieder besteht, und zwar P,I und D. Diese Buchstaben stehen für Proportional, Integral und Differenzial. Zugleich gibt es auch die drei zugehörigen Parameter K_p , K_i und K_d .

Parameter Tabelle

$e(t)$	Regeldifferenz
$y(t)$	Führungsgröße (Sollwert)
$u(t)$	Stellgröße
$r(t)$	eingelesener Farbwert (Istwert)
K_p, K_i, K_d, dT	PID-Regler Beiwerte
L_Power	Kraft des linken Rads
R_Power	Kraft des rechten Rads

Das P-Glied ist die einfachste Teil. aber gleichzeitig ist er der Hauptanteil, der die Gegenwart betrachtet und für eine schnelle Reaktion sorgt. Er misst die Regeldifferenz, multipliziert sie mit dem vorgegebenen Proportionalbeiwert und ergibt den ersten Anteil der Stellgröße.

$$e(t) = y(t) - r(t)$$

$$u(t) = K_p e(t)$$

Das I-Glied berücksichtigt die alten Messwerte und sorgt dafür, dass der sollwert erreicht werden kann. d.h. die bei proportionalen Regelungen bleibenden Störungen können durch die Verwendung eines I-Reglers behoben werden.

$$u(t) = K_i \int_0^t e(\tau) d\tau$$

Das D-Glied sorgt für die künftigen Messwerte und reagiert auf die Änderungsgeschwindigkeit der Regeldifferenz. Alleine lässt sich das D-Glied als Regler nicht einsetzen.

Außerdem ermöglicht es eine schnelle Nachregelung bei plötzlich eingreifenden Störeinflüssen.

$$u(t) = K_d \frac{d}{dt} e(t)$$

Die Ausgangsgröße eines idealen PID-Reglers sieht folgenderweise aus (Die Allgemeine Formel) :

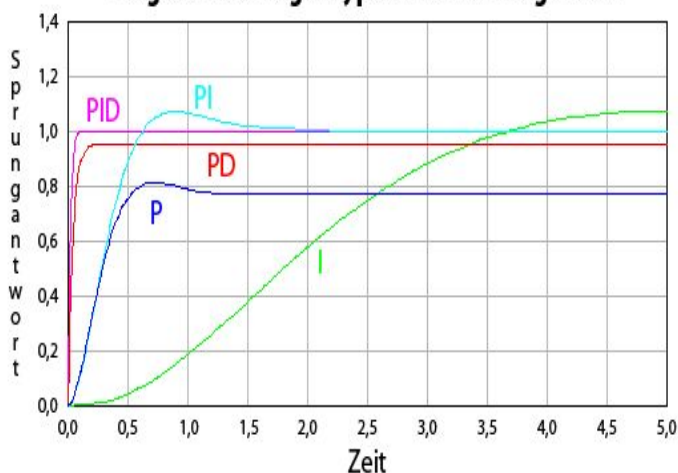
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Je nach dem Bedarf lassen sich die Anteile des PID-Reglers auch als einzige Regler verwenden, wie z.B. P-, I-, PI- und PD-Regler.

Zwar ist es schwierig, ein PID-Regler zu bilden und es reicht manchmal aus zweipunktigen Regler zu verwenden. Aber er bietet ein paar Vorteile an.

PID-Regler verknüpft die guten Eigenschaften der anderen Reglern, außerdem verfügt über gute Genauigkeit und reagiert sehr schnell. Zuletzt kann man mit einem guten Pid-Regler fast jedes System in den Griff bekommen.

Vergleich der Reglertypen in einem Regelkreis



PID-Regler betrachtet man als stetige lineare Regler. Es gibt auch zwei andere Arten von Reglern:

- * Nichtlineare Reglern wie Fuzzy-Regler, adaptive Regler und Extremwertregler.
- * Unstetige Regler wie Zweipunktregler, Dreipunktregler und Mehrpunktregler.

In unserem Roboter wurde ein GUI (Graphical User Interface) benutzt. Die GUI verfügt über drei Knöpfe : Abfahren, Stop, Nachzeichnen, jeder Knopf für eine Funktion.

Das Hauptprogramm verbirgt sich in Abfahren-Funktion und besteht aus einer Schleife.

Sowohl die idealen Beiwerte des PID-Reglers als auch die passenden Spannungen der Motoren haben wir schon durch Experimente bestimmt und in Hauptprogramm als Konstante hinzugefügt.

Die Anfangswerte wie Integral,Differenzial und Regeldifferenz wurden zum Beginn natürlich gleich Null eingesetzt.

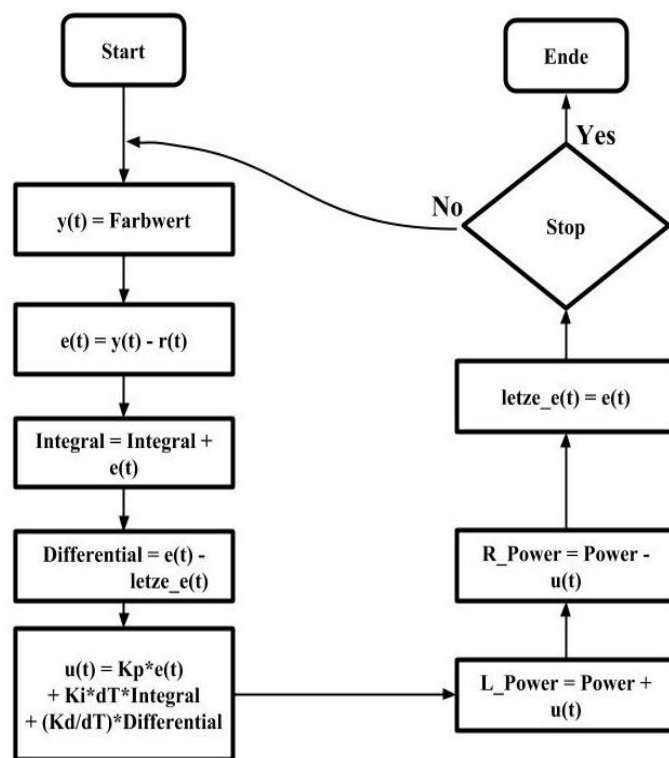
Normalerweise benutzt man im Linienfolger einen Lichtsensor aber wir haben einen Farbsensor verwendet, weil er während der Versuche bessere Genauigkeit zeigte.

Am Anfang wird auch der Roboter so gelegt, dass der

Farbsensor senkrecht über der Linie hängt, damit er zuerst das Schwarz erkennt. Der erkannte Farbwert wird im Programm als Führungsgröße (Sollwert) deklariert, dann fährt der Roboter an und der Farbsensor erkennt neue Werte (Istwert).

Im Programm wird der Regeldifferenz zwischen dem Sollwert und dem zeitlichen Istwert berechnet und zur Integral addiert, dann wird der neue Integralwert gespeichert .

Im nächsten Phase wird die von der vorherigen Schleife entstehenden Regeldifferenz von der neuen Regeldifferenz subtrahiert und als die neuen Differential eingesetzt. Nun kommt der PID-Regler ins Spiel. Er berechnet die Stellgröße und steuert den Motoren so, dass sie das Fahrzeug drehen, um die Regeldifferenz so weit wie möglich um Null zu nähern.



Die Funktion von Linienfolger

Man kann sagen, dass die Stellgröße entspricht der Drehung. Anschließend wird die neue Regeldifferenz für die nächste Runde angemerkt und wiederholt sich der Prozess.

Die während der Funktion eingelesenen Farbwerte werden in einem globalen Array gespeichert.

Die Stop-Funktion hat nur die Aufgabe, die Schleife zu beenden..

In der Nachzeichnen-Funktion sieht es ähnlich wie in der Abfahren-Funktion aus. der Unterschied besteht nur darin, dass anstatt neue Farbwerte abzulesen, werden die im Array schon gespeicherten Farbwerte in die Schleife durchgeführt.

Beim Abfahren der Linie bewies der Roboter die Effektivität des PID-Reglers und folgte der Linie nahtlos aber leider tritt ein Problem beim Nachzeichnen auf.

Eigentlich konnte der Roboter die abgefahrene Strecke ganz genau reproduzieren aber nach der Fixierung des Stifts wurden Störungen beobachtet.

Aufgrund der von Stift entstehenden Reibung und der Ungenauigkeit der Motoren wich der Roboter ab und fuhr in falsche Richtungen.

Wir versuchten mit verschiedenen Stiften die Fixierungsweise zu ändern aber es gelingt uns nicht.

Mangels der Zeit konnte das Problem nicht behoben werden aber eine mögliche Lösung wäre, einen Regler zu gestalten, um die Störungen zu eliminieren und den Prozess in Ordnung zu bringen.

Das Ergebnis ist ziemlich erfreulich trotz der begrenzten baulichen Möglichkeiten und der Ungenauigkeit von Motoren und Sensoren.

Außerdem ist der Roboter aufgrund der GUI simpel zu steuern und lässt sich mit ein paar Erweiterungen in vielen Branchen wie Lieferservice, Fahrdienst oder Reinigungsdienst einsetzen. Eine mögliche Erweiterung wäre z.B. ein Planierschild mit einem Drucksensor und einem Ultraschallsensor. Der Roboter soll auf solche Weise programmiert werden, dass er Hindernisse erkennt und sie entweder zur Seite verschiebt oder umfährt, wie ein autonomer Bulldozer oder Schneepflug.

Eine weitere Anwendungsmöglichkeit wäre beispielsweise beim Begleiten von Sehbehinderten oder auch von Senioren in Altenwohnheimen.

Fazit

Zusammenfassend kann man feststellen, dass den selbstständigen Robotern in unserer modernen Gesellschaft immer mehr Bedeutung eingeräumt wird. Niemand weiß wie unser Leben morgen aussieht aber viele Entwicklungen zeichnen sich schon heute ab. Vielleicht kann unserer Roboter nun nur Feldbegrenzungen zeichnen, aber wer weiß, was die Zukunft für ihn bringt.

Die Erfindungen von heute sind die Basis für die Anwendungen von Morgen und Die Zukunft bleibt unbekannt, doch sicher ist, dass die Innovationen der Robotik, die Bildverarbeitung und Montagetechnik unseres Lebens nachhaltig beeinflussen, Schon heute aber noch viel stärker morgen.

Literatur

- [1] W. Schumacher , M. Maurer : “ Grundlagen der Regelungstechnik” , Institut für Regelungstechnik IFR, https://www.ifr.ing.tu-bs.de/static/files/lehre/vorlesungen/gdr/Skript_GdR.pdf ,Stand: 18.09.2014
- [2] Mondada Francesco, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Klaptocz Adam, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, Alcherio Martinoli, "The E-puck, a Robot Designed for Education in Engineering", Proceedings of the 9th conference on Autonomous Robot Systems and Competitions, Vol. 1, pp: 59-65, 2009.
- [3] Zimu Wang , “A MODEL OF LINE FOLLOWING ROBOT USING PID CONTROLLER” , Stand : 2015, <http://hig.diva-portal.org/smash/record.jsf?pid=diva2%3A921842&dswid=8312>
- [4] Alexander Behrens, Linus Atorf, Robert Schwann, Bernd Neumann, Rainer Schnitzler, Johannes Ballé, Thomas Herold, Aulis Telle, Tobias G. Noll, Kay Hameyer, and Til Aach, “MATLAB Meets LEGO Mindstorms—A Freshman Introduction Course Into Practical Engineering”, IEEE Transactions on Education, vol. 2, no. 2, May 2010.
- [5] Ziegler J.G and Nichols N. B., Optimum Settings for Automatic Controllers, pp: 759–768, trans. ASME, 1942.
- [6] Karl Heinz Fasol, Klaus Diekmann (Hrsg.), “ Simulation in der Regelungstechnik “, ISBN 978-3-642-84261-0
- [7] International Federation of Robotics, World Robotics 2015 Industrial Robots <http://www.ifr.org/industrial-robots/statistics/>
- [8] Mike Williams, History of Robotics. A class assignment, Ball University. ITDPT 303 Manufacturing Systems. <http://www.bsu.edu/web/mawilliams/history.html>
- [9] RWTH - Mindstorms NXT Toolbox for MATLAB. Available: <http://www.mindstorms.rwth-aachen.de/trac/wiki/Download>