

Autonomes Einparken im Modellversuch

Vincent Siermann, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Während des Projektseminars „Elektrotechnik/Informationstechnik“ der Otto-von-Guericke-Universität Magdeburg wurde ein Modellversuch zum Parklenkassistenten durchgeführt. Unter Benutzung von „Lego-Mindstorms“ konnten die technischen Abläufe eines autonom ein- und ausparkenden Fahrzeuges demonstriert und erweitert werden. Über die Vermessung einer Parklücke ist das entwickelte System in der Lage, eine passende Lücke zu erkennen. Die über einen Algorithmus ausgeführte Messung wird im Anschluss mit der Fahrzeuggröße verglichen. Zur Umsetzung der Lenkbefehle des Ein- und Ausparkvorganges dienen definierte Fahrbahnmarkierungen.

Schlagwörter—Assistenzsystem, Ausparken, Autonom, Einparken, Lego-Mindstorms, Parklückenerkennung

I. EINLEITUNG

Ein Fahrzeug besitzt heutzutage immer häufiger Assistenzsysteme. Die *Park Distance Control* (PDC) gehört heute nahezu zur Serienausstattung eines PKWs. Selbst Systeme wie Spurhalteassistent, Spurwechselassistent oder Intelligente Tempomaten gehören zum „Must-have“ bei einem Neufahrzeug. Diese sollen dem Fahrer assistieren und in bestimmten Situationen das Führen des Fahrzeuges erleichtern. Das autonome Fahren ist heutzutage in der Forschung und Entwicklung an oberste Stelle geraten und kann eine enorme Unterstützung für die Fahrzeugführer darstellen. Autonom fahrende Fahrzeuge sollten nicht nur selbstständig fahren, sondern auch in verschiedenen Alltagssituationen autonom Ein- und Ausparken können. Der Parklenkassistent erleichtert schon heute das Einparken, sodass der Fahrer nur die Geschwindigkeit des Fahrzeuges regulieren muss.

Innerhalb des Projektseminars „Elektrotechnik/Informationstechnik“ der Otto-von-Guericke-Universität Magdeburg war es das Ziel, einen autonom ein- und ausparkenden Roboter zu entwickeln. Die Konstruktion sollte einem PKW nachempfunden werden. Für die Konstruktion wurde ein „Lego-Mindstorms“ Bausatz verwendet. Den Kern bildet hier der „NXT-Baustein“. Dort lassen sich bis zu 4 Sensoren und 3 Motoren anschließen. Durch die „RWTH - Mindstorms NXT Toolbox“ der RWTH Aachen [1] ist es möglich, eine Kommunikation zwischen NXT und MATLAB herzustellen. Der Roboter sollte selbstständig erkennen, ob die Größe einer Parklücke für das Fahrzeug geeignet ist. Je nach Größe der Parklücke müssen verschiedene Szenarien berücksichtigt werden. Während des Parkvorgangs sollte der Roboter mit einer konstanten Geschwindigkeit einparken und zusätzlich beim Ausparken selbstständig erkennen, ob das Fahrzeug zuvor quer oder längs eingeparkt wurde.

II. VORBETRACHTUNGEN

A. Stand der Technik

Der Parklenkassistent ist nun seit mehr als 10 Jahren in Kraftfahrzeugen erhältlich. Er basiert auf der *Park Distance Control* (PDC), dieser misst die Distanz zu vorderen oder hinteren Gegenständen. Zusätzlich werden dem Fahrer durch akustische und/oder optische Signale Informationen zur Distanz zum erkannten Hindernis übermittelt. Bei dem PDC-System muss der Fahrer noch selbstständig die Lenkvorgänge durchführen. Der Parklenkassistent hingegen übernimmt die Lenkvorgänge und der Fahrzeugführer muss nur noch die Geschwindigkeit beim Einparken regulieren. Der Parklenkassistent der „Volkswagen AG“ muss vor dem Einparkvorgang aktiviert werden. Sobald das System aktiv ist, wird dem Fahrzeugführer empfohlen, mit einer Geschwindigkeit von unter $30 \frac{\text{km}}{\text{h}}$ an Parklücken vorbeizufahren. Währenddessen sucht das System eine passende Parklücke. Sollte die Geschwindigkeit von $30 \frac{\text{km}}{\text{h}}$ überschritten werden, schaltet sich das System ab. Erkennt das System eine Lücke, wird mit dem Einparkvorgang begonnen. Dazu wird dem Fahrer signalisiert anzuhalten und den Rückwärtsgang einzulegen. Während des Einparkvorganges darf eine Geschwindigkeit von $7 \frac{\text{km}}{\text{h}}$ nicht überschritten werden [2].

B. Geschwindigkeitsmessung

Tabelle I
MESSDATEN ZUR GESCHWINDIGKEIT

Power (in %)	Zeit (in s)	Geschwindigkeit (in $\frac{\text{cm}}{\text{s}}$)	Faktor (in $\frac{\text{cm}}{\text{s}\cdot\%}$)
10	21,190	4,71	0,471
20	10,616	9,41	0,470
30	7,066	14,15	0,471
40	5,271	18,97	0,474
60	3,461	28,89	0,481
80	2,642	37,85	0,473
100	2,591	38,59	0,385

Bei den „Lego-Mindstorms“-Motoren kann die Leistung nur in % der Batteriespannung angegeben werden, was mit „Power“ bezeichnet wird. Um die Geschwindigkeit des Fahrzeuges zu ermitteln, wurde ein Versuch durchgeführt. Dazu erfolgte eine Messung der Zeit, die das Fahrzeug benötigt, um einen Meter (100 cm) Strecke zurückzulegen. Anhand dieser Werte konnte die Geschwindigkeit (in $\frac{\text{cm}}{\text{s}}$) ermittelt werden. Indem die berechneten Geschwindigkeiten durch die eingegebene Power (in %) dividiert werden, ergab sich ein Faktor (in $\frac{\text{cm}}{\text{s}\cdot\%}$) mit dem man jede Powereingabe in eine Geschwindigkeit umrechnen kann. Aus der Tabelle I können die für dieses Fahrzeug ermittelten Faktoren entnommen werden.

C. Lenkgeometrie

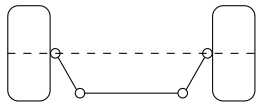


Abbildung 1. Lenktrapez

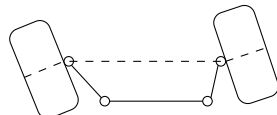


Abbildung 2. Lenktrapez bei nach links eingeschlagener Lenkung

Um ein problemloses Abrollen der Räder bei einer Kurvenfahrt zu ermöglichen, spielt die Lenkgeometrie eine große Rolle. Da der Aufbau in diesem Modellversuch nah an einem realen Fahrzeug liegen soll, wurde die Lenkung nach dem Lenktrapez (Abbildung 1) aufgebaut. Das kurveninnere Rad muss beim Fahren einer Kurve einen kleineren Radius zurücklegen als das kurvenäußere Rad. Hierzu muss die Lenkung einen unterschiedlichen Einschlagwinkel der Räder ermöglichen. Das kurveninnere Rad muss demzufolge mehr einschlagen als das kurvenäußere Rad (Abbildung 2) [3].

III. AUFBAU UND FUNKTIONSABLÄUFE

A. Aufbau des Fahrzeuges



Abbildung 3. Farbsensor

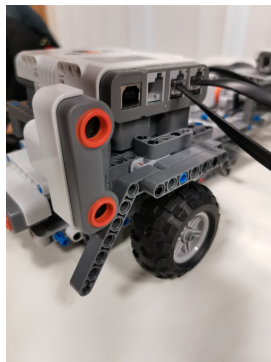


Abbildung 4. Seitlicher Ultraschallsensor

1) *Sensorik:* Durch drei Ultraschallsensoren und einen Farbsensor wird die Abfolge des Ein- und Ausparkens umgesetzt. Die Ultraschallsensoren werden zur Abstandsmessung benutzt, wobei der erste an der Front angebracht ist. An dieser Position dient er der Überwachung des Abstandes zu Objekten, welche sich vor dem Roboter befinden und löst ggf. ein Stoppen des Bewegungsablaufes aus. Der am Heck angebrachte zweite Ultraschallsensor ist dem bereits beschriebenen Sensor in seiner Funktion äquivalent, außer, dass er den Abstand zu Objekten hinter dem Roboter ermittelt. Ein weiterer, dritter Ultraschallsensor misst den Abstand zur rechten Seite, wodurch die Größe der Parklücke ermittelt wird. Um den Zeitpunkt des Umlenkens während des Einparkvorgangs zu finden, wird ein Farbsensor benutzt.

Die genaueren Funktionen der Sensoren werden in den nachfolgenden Abschnitten näher erläutert.

2) *Aktorik:* Um das Fahren und Lenken zu realisieren, wurden zwei Motoren integriert. Motor A treibt über ein Differential die Hinterachse an, um vor- und zurück zu fahren. Motor B ist für die Lenkung während des Ein- und Ausparkens zuständig. Die „Lego-Mindstorms“-Motoren lassen sich so ansteuern, dass sie nur eine bestimmte Anzahl an Umdrehungen durchführen. Dies wird über einen Befehl „TachoLimit“ festgelegt. Hierbei wird ein Winkel in Grad eingegeben um den sich der Motor drehen soll. Ein Winkel von 360° entspricht einer vollen Umdrehung. Bei der hier konstruierten Lenkung ist es möglich, 140° nach links oder rechts einzuschlagen. In diesem Fall und allen folgenden Erläuterungen wird vom Drehwinkel des Motors ausgegangen und nicht vom eigentlichen Lenkeinschlag des Roboters. Außerdem ist sie einem realen Fahrzeug nachempfunden und nach dem Lenktrapez (siehe Abbildung 1) aufgebaut.

B. Parklückenerkennung

Der Roboter sollte während des Vorbeifahrens an parkenden Gegenständen selbstständig eine Lücke erkennen und ggf. einparken. In diesem Abschnitt wird genauer auf den Ablauf zur Erkennung einer Lücke eingegangen.

Realisiert wird die Abstandsmessung über einen seitlich angebrachten Ultraschallsensor (Abbildung 4), welcher kontinuierlich beim Vorbeifahren den Abstand misst. Die Messung wird in Abbildung 5 dargestellt.

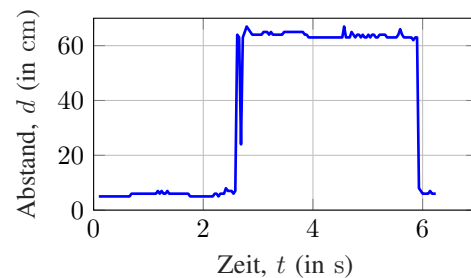


Abbildung 5. Abstandsmessung durch den seitlichen Ultraschallsensor beim Passieren einer Lücke.

Dort ist veranschaulicht wie der Sensor beim Passieren eines Gegenstandes kleine Abstandswerte misst und die Abstandswerte ansteigen, sobald die Lücke beginnt. An dem Punkt wo der Ultraschallsensor einen deutlich größeren Abstandswert misst, wird der erste *clock*-Befehl gesetzt (ermöglicht es, das Datum und die exakte Uhrzeit unter MATLAB auf ms genau abzuspeichern). Die genaue Größe des Abstandes ist für diesen Versuch nebensächlich, wichtig dabei ist ein klarer Unterschied zwischen den Abstandswerten. Sobald also der Anfang einer Lücke erkannt wird, fährt das Fahrzeug weiter und der Ultraschallsensor misst einen großen Abstand (Abbildung 5 Bereich von $2,3s < t \text{ (in s)} < 5,9s$). Wird nun vom Ultraschallsensor wieder einen Abstand der kleiner als 10 cm gemessen, startet ein zweiter *clock*-Befehl. Aus der Differenz der Anfangs- und Endzeit kann nun die Zeit ermittelt werden, die von Anfang bis zum Ende der Lücke verstrichen ist. Das geschieht über den in MATLAB definierten *etime*-Befehl. Aus dieser Zeit und mit der Geschwindigkeit des Fahrzeuges, kann nun die

Größe der Parklücke ermittelt werden, da die Geschwindigkeit konstant gehalten wird. Dies erfolgt über das physikalische Gesetz der gleichförmigen Bewegung $v = \frac{s}{t}$. Grundsätzlich wird die Größe der Parklücke beim Längseinparken mit der Fahrzeuglänge verglichen und beim Quereinparken mit der Fahrzeugbreite. Wie die Geschwindigkeit dieses Roboters mit „Lego-Mindstorm“-Motor ermittelt wurde, zeigt Punkt B der Vorbetrachtung (II-B).

C. Einparkvorgang

Der Einparkvorgang erfolgt direkt nachdem eine Parklücke identifiziert wurde. Dieser beginnt mit einem definierten Rechtseinschlag (hier ein Winkel von 130° zum Quer- und 120° zum Längseinparken). Ist dies abgeschlossen wird mit der Rückwärtsfahrt begonnen. Hier kommt der Farbsensor zum Einsatz (Abbildung 3), welcher am Heck des Fahrzeuges angebracht wurde.

1) *Quereinparken*: Der weitere Ablauf wird am Beispiel des Querparkens näher erläutert. Der Programmablaufplan des Quereinparkens wird in Abbildung 6 gezeigt. Der Farbsensor ermittelt beim Rückwärtsfahren kontinuierlich die Farbe des Untergrundes. Erkennt er eine Markierung (hier: rote Linie), wird die Rückwärtsfahrt beendet und der Motor B schlägt die Lenkung auf vollen Linkseinschlag (entspricht dem nach rechts eingeschlagenen Winkel vom 130° + einen Winkel von 150°). Nun beginnt das Fahrzeug nach vorn zu fahren und der Farbsensor ermittelt erneut die Farbe des Untergrundes. Hat dieser nun eine weitere Markierung (hier: schwarze Linie) erkannt, wird die Bewegung nach vorn gestoppt und die Lenkung gerade gestellt. Der Farbsensor wird ab diesem Zeitpunkt nicht mehr benötigt und geschlossen. Das Fahrzeug fährt nach der geraden Ausrichtung der Lenkung rückwärts, nun wird über den am Heck angebrachten Ultraschallsensor die Distanz nach hinten ermittelt. Sollte die Distanz den Wert von 5 cm erreichen, wird die Rückwärts-Fahrt beendet und der Parkvorgang damit abgeschlossen.

2) *Längseinparken*: Die Abfolge des Parkens bis zur Erkennung der roten Linie entspricht der Abfolge des Quereinparkens. Allerdings wird ein anderer Lenkeinschlag verwendet. Nach dem Erkennen der roten Linie wird die Lenkung voll nach links eingeschlagen (entspricht dem nach rechts eingeschlagenen Winkel von 120° + einen Winkel von 160°) aber anders als beim Quereinparken weiter zurück gefahren. Des Weiteren müssen beim Längseinparken drei Fälle unterschieden werden, welche nach der Größe der Parklücke definiert sind.

Fall 1: Die Parklücke ist kleiner als 50 cm. Diese Lücke ist für den entwickelten Roboter so klein, sodass in der Lücke die Position korrigiert werden muss um gerade zu stehen. Beim zurück fahren mit vollem Linkseinschlag wird der hintere Ultraschallsensor ausgelesen und solange gefahren bis dieser einen Wert von 5 cm misst. Anschließend steht der Roboter noch nicht gerade ausgerichtet in der Parklücke. Die Lenkung wird daher voll nach rechts eingeschlagen (entspricht dem nach links einschlagenen Winkel von 140° + einen Winkel von 140°) und der Roboter fährt nach vorn bis der vordere Ultraschallsensor einen Wert von 5 cm misst. Die Räder werden gerade ausgerichtet und ein Ausgleichsvorgang beginnt. Der

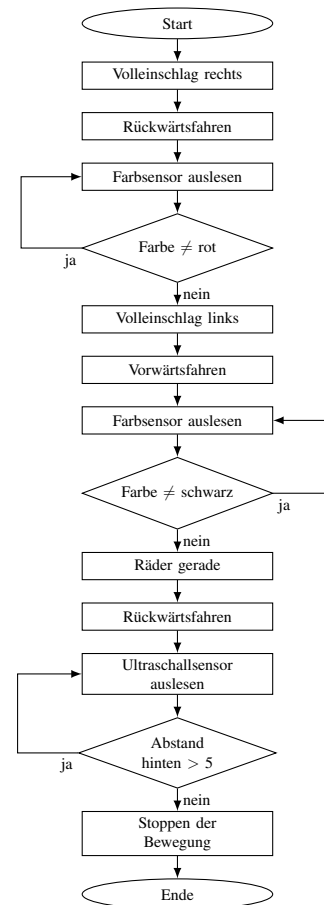


Abbildung 6. Programmablaufplan zum Algorithmus des Quereinparkens.

Ausgleichsvorgang sieht vor, dass der Roboter zum vorderen und hinteren Objekt den selben Abstand aufweist. Dabei messen die Ultraschallsensoren vorn und hinten kontinuierlich den Abstand. Es wird so lange rückwärts oder vorwärts gefahren, bis beide Sensoren den gleichen Wert aufweisen.

Fall 2: Die Parklücke ist größer als 50 cm, aber kleiner als 55 cm. Das Einparken erfolgt wie im ersten Fall, allerdings kann auf das Korrigieren innerhalb der Lücke verzichtet werden. Der Roboter steht nach dem Zurückfahren mit vollem Linkseinschlag gerade sobald der hintere Sensor einen Wert von 5 cm misst. Danach werden die Räder gerade ausgerichtet und der Ausgleichsvorgang beginnt.

Fall 3: Die Parklücke ist größer als 55 cm. Das Einparken erfolgt auch hier ähnlich zu den ersten beiden Fällen. Allerdings muss beim Zurückfahren mit vollem Linkseinschlag der vordere Ultraschallsensor zum stoppen der Bewegung eingesetzt werden, da der hintere Sensor aufgrund der Größe der Parklücke, immer noch große Werte misst, obwohl das Fahrzeug schon gerade ausgerichtet ist. Dies wäre auch der Fall wenn die Parklücke nur nach vorn eine Begrenzung aufweist. Die Bewegung wird bei einem Wert des vorderen Sensors von 21 cm gestoppt. Danach startet nicht wie in den anderen Fällen ein Ausgleichsvorgang, sondern das Fahrzeug fährt an das vordere Objekt bis auf einen Abstand von 10 cm heran.

D. Ausparkvorgang

Grundsätzlich werden beim Ausparkvorgang die Abläufe der einzelnen Einparkvorgänge umgekehrt. Das Fahrzeug erkennt allerdings selbständig, ob es quer oder längs in einer Parklücke steht. Dies wird über die gemessenen Abstände der drei Ultraschallsensoren festgelegt. Für das Längsausparken ist es wichtig, die drei Fälle vom Einparken zu berücksichtigen. Die Messungen erfolgen hierbei nur über den hinteren Ultraschallsensor. Der Roboter steht durch den Ausgleichsvorgang mittig in der Lücke. Misst dieser Sensor nun einen Abstand von unter 10 cm, wird die kleinsten Lücke identifiziert (III-C2 Fall 1). Wenn der Abstand zwischen 10 cm und 15 cm liegt, steht der Roboter in der Lücke wie im Fall 2 beschrieben (III-C2 Fall 2). Sollte nur der Abstand einen Wert von mehr als 15 cm betragen, ist das Objekt nach hinten weit entfernt oder nicht vorhanden, in dieser Weise im Fall 3 (III-C2 Fall 3) zu sehen.

E. Eingabe in die GUI

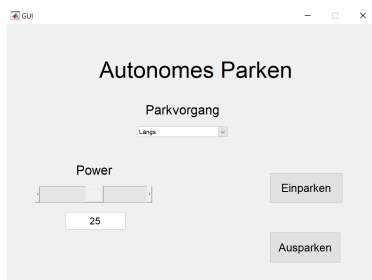


Abbildung 7. Eingabeoberfläche der GUI

In MATLAB gibt es die Möglichkeit eine grafische Benutzeroberfläche (*graphical user interfaces* auch „GUI“ genannt) zu entwerfen. „[Diese] ermöglichen die Point-and-Click-Steuerung von Softwareanwendungen“ [4] und können über das MATLAB Add-On „GUIDE“ entworfen werden.

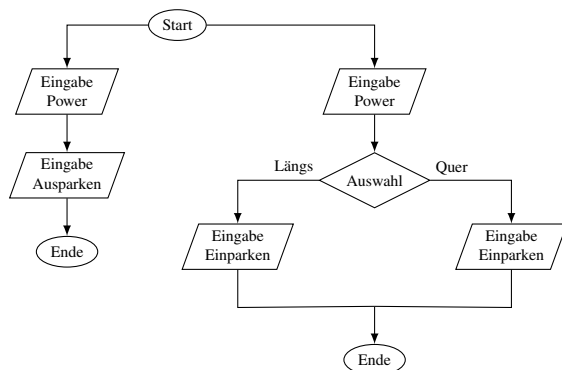


Abbildung 8. Programmablaufplan zur Eingabe in die graphische Benutzeroberfläche (GUI)

Hier wird eine GUI verwendet, um die Befehle zum Ein- und Ausparken nicht einzeln eingeben zu müssen. Wird die GUI (Abbildung 7) aufgerufen, kann mittels eines Pop-Up-Menüs ausgewählt werden ob der Einparkvorgang längs oder quer durchgeführt werden soll. Über einen Schieberegler kann

die Power des Motors ausgewählt werden. Das Programm wird mit den Schaltflächen „Ein- und Ausparken“ gestartet. In der Abbildung 8 ist der Ablauf der Eingabe dargestellt. Zum Ausparken ist die Eingabe über das Pop-Up-Menü nicht erforderlich.

IV. ERGEBNISDISKUSSION

Die Anfangs gesetzten Ziele wurden erfolgreich umgesetzt und es wurde ein Roboter entwickelt, welcher in der Lage ist, autonom ein- und ausparken. Problematisch beim Aufbau ist, dass keine Spurführung vorgesehen ist. Das Fahrzeug muss vor jedem Vorgang exakt ausgerichtet werden, damit es gerade an Lücken vorbei fährt. Zusätzlich konnte bei den ersten Versuchen der erforderliche Lenkradius nicht erreicht werden. Problematisch wurde dies beim Längseinparken, weshalb anfangs nur in Lücken größer 50 cm eingeparkt werden konnte. Dies wurde durch einen Umbau der ursprünglichen Lenkung korrigiert. Der resultierende erhöhte Lenkeinschlag ermöglichte es auch in Lücken unter 50 cm einzuparken. Ein weiterer Nachteil sind die Fahrbahnmarkierungen, welche als Hilfestellung zur Steuerung der Lenkung erforderlich sind, um einen erfolgreichen Parkvorgang zu gewährleisten.

V. ZUSAMMENFASSUNG UND FAZIT

Der entwickelte Roboter kann anhand von Messdaten durch Ultraschallsensoren einen Ein- und Ausparkvorgang steuern. Dabei erkennt er durch Vermessung die Größe einer Parklücke. Zusätzlich plant er selbstständig anhand dieser Daten wie der Vorgang des Parkens am effizientesten umzusetzen ist. Beim Ausparken erkennt er durch die Abstandswerte der Ultraschallsensoren, ob er längs oder quer zur Fahrbahn ausgerichtet ist.

In Zukunft sollten Lösungsansätze entwickelt werden, wie ohne Fahrbahnmarkierungen gearbeitet werden kann. Des Weiteren ist es sehr wichtig, dass dieses System mit einem autonomen Fahrsystem verbunden wird. Dies würde das Problem beseitigen, dass der Roboter keine Spurführung zum fahren hat. Somit könnte man problemlos überall vorbeifahren und der Roboter würde seine Fahrtrichtung nicht verlieren. Die Lenkung funktioniert für diesen Modellversuch angemessen präzise. Allerdings sollte diese dennoch überarbeitet werden. Möglicherweise können Alternativen zu Legobauteilen getestet werden, um in Zukunft genauere Lenkbefehle umsetzen zu können.

ANHANG

Im Rahmen der Abschlusspräsentation des Projektseminars wurde ein Video gedreht, welches die Funktion des Einparkroboters zeigt: <https://www.youtube.com/watch?v=eopOMXVead0>.

LITERATURVERZEICHNIS

- [1] RWTH AACHEN (Hrsg.): *RWTH - Mindstorms NXT Toolbox*. <https://www.mindstorms.rwth-aachen.de/trac/wiki/Download4.07>
- [2] VOLKSWAGEN AG (Hrsg.): *Der Parklenkassistent: Konstruktion und Funktion: Selbststudienprogramm 389*. <https://www.motor-talk.de/forum/aktion/Attachment.html?attachmentId=677825>. Version: 2007
- [3] *Fachkunde Kraftfahrzeugtechnik*. 30., neubearbeitete Auflage. Haan-Gruiten : Verlag Europa-Lehrmittel, 2013 (Europa-Fachbuchreihe für Kraftfahrzeugtechnik). – ISBN 978–3–8085–2240–0
- [4] MATH WORKS (Hrsg.): *Matlab GUI*. <https://de.mathworks.com/discovery/matlab-gui.html>