Name: \_\_\_\_\_\_\_\_\_\_ Klasse: \_\_\_ Datum: \_\_\_\_\_\_\_\_

# Lösungsblatt Aufgabe 4

# Einparkhilfe

## Programmieraufgaben

**1. Parklücke finden**

Der Einparkvorgang lässt sich mit demontiertem Servo-Hebel leicht per Hand nachvollziehen. Die Länge der erforderlichen Parklücke kann am Einfachsten mit Hilfe entsprechender Bleistift-Markierungen auf der Fahrbahn ausgemessen werden (siehe Abbil­dung).



*Parallel\_Parking\_Model.jpg*

*Breite* der Parklücke: Die Parklücke muss mindestens so breit sein wie das Fahrzeug, also $b$ = 14,5 cm, zuzüglich 2 cm bis zur Fahrbahnbegrenzung.

Die Lücke sollte zusätzliche 1,5 cm breiter sein, da die Reifen beim Lenkeinschlag über die Fahrzeugbreite hinausragen.

Ist der Ultraschallsensor so montiert, dass er ab dem äußeren Rand des Fahrzeugs misst, sind weitere 2 cm bis zur Seitenlinie sowie deren Breite (2 cm) hinzuzuaddie­ren.

Also befindet sich rechts neben dem Fahrzeug eine aus­reichend breite Lücke, wenn der Sensor **mindestens 22 cm** bis zum nächsten Objekt misst.

*Länge* der Parklücke: Beim Einparkvorgang in zwei Zügen muss sich der Mittelpunkt des Differentials zunächst etwa 17,5 cm mit maximalem Lenkeinschlag nach rechts rückwärts bewegen, bevor er die Mitte der Seitenlinie kreuzt, anschließend weitere 17,5 cm mit maximalem Lenkeinschlag nach links. Zu dieser Strecke von 35 cm sind die Länge des Fahrzeughecks (5,5 cm, gemessen von der Mitte des Differenzials bis zum äußersten Ende des Fahrzeugs) zu addieren und der „Start-Abstand“ von 2 cm (zwischen der Position des Differenzials beim Start des Einparkvorgangs und der Stelle, an der das linke Vorderrad den Seitenstreifen kreuzt) zu subtrahieren. Damit summiert sich die benötigte Mindestlänge der Parklücke auf

$$l=38,5 cm$$

Aus der Mindestlänge der Parklücke $l$ lässt sich die Zahl der für diese Distanz erforderlichen Impulse $i$ nach der folgenden Formel bestimmen (siehe Aufgabe 1):

$$i=l·\frac{1}{20,5}·\frac{13}{7}·63,9 \frac{1}{cm}≈l· 5,789 \frac{1}{cm}$$

Berück­sichtigen wir den angepassten Umrechnungsfaktor aus dem Test in Aufgabe 1 so korrigiert sich die Umrechnung zu

$$i=l·\frac{100}{0,0017}≈l· 5,882 \frac{1}{cm}$$

Damit erfordert die gesuchte Parklückenlänge $l=38,5 cm$ also $i≈227$ **Impulse**.

*Streuung* des Ultraschall-Signals: Wenn Hindernisse im Abstand von 4 cm rechts neben der Fahrbahnbegrenzung stehen, beträgt der Abstand zum Ultraschallsensor 8 cm. Durch den Streuwinkel des Ultraschallsensors wird die Parklücke erst erkannt, wenn die Mitte des Ultraschallsensors bereits 3 cm in die Parklücke „hineinragt“. Das Ende der Parklücke wird entsprechend bereits 3 cm vor der Stelle erkannt, an der die Sensormitte den äußeren Rand des Hindernisses erreicht. Misst der Sensor also über eine Strecke von 32,5 cm (oder $i'≈191$ **Impulse**) eine Breite von mindestens 22 cm, ist die gefundene Parklücke lang genug.

Für die Messung der Parklückenlänge ist es irrelevant, an welcher Stelle der Ultra­schallsensor seitlich am Fahrzeug angebracht ist. Für den Start des Einparkvorgangs muss das Fahrzeug so zum Stehen kommen, dass sich die Hinterachse (bzw. das Differential) genau 2 cm hinter dem Ende der Parklücke (also dem äußer­sten Rand des Hindernisses) befindet. Dazu muss es, nachdem der Ultraschallsensor das Ende der Parklücke erkannt hat,

* weitere 3 cm (um die das Ende der Parklücke zu früh erkannt wird)
* plus obige 2 cm
* zuzüglich des Abstands von der Mitte des Ultraschallsensors zur Hinterachse

weiterfahren. Ist der Ultraschallsensor beispielsweise 7 cm vor der Hinterachse montiert, muss das Fahrzeug nach Erkennen des Parklückenendes weitere 12 cm (oder **71 Impulse**) fahren.

1b. Zustandsübergangsdiagramm:



*State-Transistion\_Diagram\_Find\_Parking\_Space.drawio*

1c. Programm (Beispiel) Parklückensuche:



*Find\_Parking\_Space.ft*

**2. Einparkmanöver**

2a. Die Bestimmung der zurückzulegenden Impulse gelingt am Einfachsten mit dem Interface-Test: Das Fahrzeug wird manuell eingeparkt; dabei werden die vom Encoder geliefer­ten Impulse im Interface-Test abgelesen und notiert. Für das Zurücksetzen sind – mit nach rechts bzw. nach links eingeschlagener Lenkung – **jeweils 125 Impulse** erforderlich.

2b. Damit das Fahrzeug zum Schluss in der Mitte des benötigten Einparkbereichs der Parklücke steht, muss es nach dem Zurücksetzen 5,75 cm vorfahren; das entspricht etwa **34 Impulsen**.

2c. Programm (Beispiel) Einparkmanöver:



*Parking\_Manoevre.ft*

Die Variablen „backward\_distance“ und „forward\_distance“ enthalten die zu fahrenden Impulse (125 bzw. 34). „minangle“ und „maxangle“ sind die maximal möglichen Einschlagwinkel des Servos im Servo-Halter (siehe Aufgabe 2); sie liegen etwa bei 130 bzw. 375. Für die kurze Pause nach einem Lenkeinschlag („pause“) sind 250 ms eine gute Wahl.

## Experimentieraufgaben

**1. Einparkassistent**

Eine Lösung der Aufgabe ist im Programm *Autonomous\_Parking.ft* zusammen­gefasst.

Der Blinker darf nicht automatisch aktiviert werden, da sonst mitten im Einparkvor­gang auf den linken Blinker gewechselt wird.

**2. Berechnung des Einparkvorgangs**

Der Mittelpunkt des Differentials bewegt sich beim Einparkvorgang exakt entlang zweier Kreissegmente der Länge $s$. Der Mittelpunkt $M$ des ersten Kreises entspricht bei unserer Achsschenkellenkung dem Mittelpunkt des Wen­dekreises: Die Verlänge­rungen der beiden vorderen Achsstummel der Lenkung und der Hinterrad­achse treffen sich exakt in diesem Punkt.



*Mathematical\_Model\_Parallel\_Parking.jpg*

Wir berechnen nun die Länge dieses Kreissegments $s$. Damit wissen wir, wie viele Encoder-Impulse wir benötigen, um dieses Segment abzu­fahren. Es gilt: $s$ verhält sich zum Kreisumfang ($=2π∙r$) wie $δ$ zu 360°, also:

$$s=\frac{π∙δ∙r}{180°}$$

Den Winkel $δ$ können wir durch den Radius $r$ ausdrücken, denn in dem rechtwinkligen Dreieck, das $M$ und die beiden Schnittpunkte der Kreissegmentlinien mit der Seitenlinie bilden, gilt:

$$\cos(δ=\frac{(r-d)}{r})$$

Für den Abstand $d$ vom Achsmittelpunkt zur (Mitte der) Seitenlinie nehmen wir an, dass sich das Fahrzeug etwa 2 cm links von der Fahrbahnbegrenzung befindet. Dann gilt bei einer Fahrzeugbreite von $b$ = 14,5 cm und mit unserer 2 cm breiten Seitenlinie:

$$d=10,25 cm$$

Den Radius $r$ des Wendekreises berechnen wir aus dem Dreieck, das von $M$ und den beiden Mittelpunkten der Vorder- und Hinterachse des Fahrzeugs gebildet wird. Den Achsabstand bezeichnen wir mit $a$:

$\tan(α=\frac{a}{r})$ , also: $r=\frac{a}{\tan(α)}$

Der Winkel $α$ entspricht dabei dem (maximalen) Einschlagswinkel unseres Servo-Hebels. Bei unserem Fahrzeug liegt der Einschlagswinkel der Lenkung bei jeweils etwa 38°. Den Achsabstand können wir messen: $a=15,5 cm$.

Wir erhalten schließlich (mit $α=38°$):

$$s=\frac{π}{180°}∙\arccos(\left(\frac{a-d∙\tan(α)}{a}\right))∙\frac{a}{\tan(α)}≈21,15 cm$$

Das entspricht etwa 122 Impulsen.

Jetzt können wir auch die benötigte Länge der Parklücke berechnen. Nach dem *Satz des Pythagoras* gilt:

$$r^{2}=\left(\frac{t}{2}\right)^{2}+\left(r-d\right)^{2}$$

Damit folgt (mit $α=38°$):

$$t=2∙\sqrt{\left(\frac{a}{\tan(α)}\right)^{2}-\left(\frac{a}{\tan(α)}-d\right)^{2}}≈34,73 cm$$

Zu diesem Wert müssen wir noch die Länge des Hecks des Fahrzeugs hinzurechnen (5,5 cm) und 2 cm abziehen, die wir über die Parklücke hinaus fahren. Unsere Parklücke sollte also mindestens 38,23 cm lang sein - das entspricht ziemlich genau dem experimentell bestimmten Wert.

Anlagen

# Einparkhilfe

## Erforderliches Material

* PC für Programmentwicklung, lokal oder über Web-Schnittstelle.
* USB-Kabel oder BLE- bzw. WLAN-Verbindung für die Übertragung des Programms auf den TXT4.0.
* Test-Fahrstrecke

## Weiterführende Informationen

[1] VDA: [*Automatisierung: Von Fahrerassistenzsystemen zum automatisierten Fahren*](https://www.vda.de/dam/vda/publications/2015/automatisierung.pdf). September 2015.

[2] Nina Tetzlaff: [*Autonome Straßenfahrzeuge*](https://www.dpma.de/docs/dpma/veroeffentlichungen/erfinderaktivitaeten/ea_2018_2019.pdf). Erfinderaktivitäten 2018/2019, DPMA, S. 62-81.

[3] VW: [*Park Assist Steering 2.0*](https://www.vdveer-engineering.nl/en/information/vag-ssp/ssp-vag/ssp-vag-en/331-ssp-494-park-assist-steering-20/download). Design and function. Service Training, 2021.