

Hochschule München

Projektarbeit Wackelroboter



Betreuer	Prof. Dr. Norbert Nitzsche
Verfasser	Valentyn Chepil, Stephan Morongowski, Severin Schendel, Aleksandar Stoiljkovic, Timo Veit
Fachbereich	Maschinenbau - MBB
Abgabetermin	28.2.2018

Inhaltsverzeichnis

1	Eigenbau eines Motortreibers	1
1.1	Anforderungen an den zu entwickelnden Treiber	1
1.2	Vergleich quelloffener Treiber	1
1.2.1	VESC	1
1.2.2	oDrive	2
2	Definition eines roboterfesten Koordinatensystems	3
3	Navigation	4
3.1	Sensorik	4
3.2	Inkrementalgeber	4
4	Auslegung Kurvengeschwindigkeit	7

Abbildungsverzeichnis

1	Roboterkoordinatensystem (Quelle: [roboterKS_Bild])	3
2	Rotation um den Momentanpol (Quelle: eigene Darstellung)	4
3	Kräfte bei der Kurvenfahrt (Quelle: eigene Darstellung)	7

1 Eigenbau eines Motortreibers

Aufgrund der schlechten Verfügbarkeit eines zur feldorientierten Regelung von Synchronmotoren geeigneten Motortreibers wurde überlegt, einen solchen Treiber innerhalb der Projektarbeit zu entwickeln und zu bauen. Zur Eruierung des zeitlichen und kostenmäßigen Aufwandes wurden zunächst wahrscheinliche Anforderungen festgelegt und dann bereits im Internet frei verfügbare Lösungen untereinander verglichen.

1.1 Anforderungen an den zu entwickelnden Treiber

Einfaches Design, Erweiterbarkeit, quelloffen.

Technische Daten:

Motorspannung	bis 25V
Strom	dauerhaft 30A, Spitze 50A
Schnittstellen	I2C, USB, PWM

Die Herstellungskosten sollten 80,- € nicht übersteigen.

1.2 Vergleich quelloffener Treiber

1.2.1 VESC

Der VESC - vector electronic speed control [**vesc**] ist ein openSource Projekt von Benjamin Vedder. Der Treiber wurde ausgelegt, um ein Skateboard mit einem bürstenlosen Gleichstrommotor anzutreiben und ist in der Lage, 240A Spitzenstrom und 50A Dauerstrom bei bis zu 60V Versorgungsspannung zu liefern. Er verfügt über eine umfangreiche Konfigurationssoftware, über die vom PC aus alle nötigen Voreinstellungen vorgenommen werden können. Das ganze Projekt wirkt auf den ersten Eindruck sehr ausgereift.

Technische Daten:

Microcontroller	STM32F4
MOSFET Treiber	DRV8302
MOSFETS	6 IRFS7530
Motorspannung	8V - 60V
Strom	dauerhaft 50A, Spitze 240A
Schnittstellen	PPM signal (RC servo), analog, UART, I2C, USB, CAN-Bus
Größe	40mm mal 60mm

Kosten im Eigenbau für zwei Stück:

Bauteile	ca. 120,- €
Platinen	ca. 100,- €
Lötzubehör	ca. 20,- €
Summe	ca. 240,- €

Beschaffungskosten für zwei fertig aufgebaute Platinen: **ca. 280,-€**

1.2.2 oDrive

Der ODrive ist eine Entwicklung verschiedener Personen, da grundsätzlich jeder an der Entwicklung des Treibers teilnehmen kann. Dieser Treiber steuert bis zu zwei Motoren mit je ca. 100A Spitzenstrom. Das Projekt befindet sich noch in der Entwicklungsphase. Die Konfiguration des Treibers muss in die Firmware kompiliert werden.

Technische Daten:

Microcontroller	STM34F4
MOSFET Treiber	DRV8301
MOSFETS	28 NTMFS4935NT1G
Motorspannung	8V - 30V
Strom	dauerhaft ?, Spitze > 100A
Schnittstellen	USB, CAN, UART, PWM, and step/dir interface
Größe	110mm mal 50mm

Kosten im Eigenbau pro Stück:

Bauteile	ca. ???,- €
Platinen	ca. ???,- €
Lötzubehör	ca. 20,- €
Summe	ca. ???,- €

Beschaffungskosten als fertig aufgebaute Platine: **ca. 120,-€**

2 Regelung von Schrittmotoren

Da das Verwenden eines Schrittmotors zur Debatte stand, wurde hierzu eine Recherche angestellt. Die Unterschiede zwischen einem BLDC-Motor und einem Schrittmotor sind im wesentlichen:

BLDC-Motor

Verwendung von Drehstrom / 3 Phasen

niedrige Polpaarzahl

Schrittmotor

meist Verwendung von 2 Phasen

hohe Polpaarzahl

2.1 Resume

Zur feldorientierten Vektorregelung bei BLDC Motoren wurden einige Erklärungen und die zugehörigen Berechnungen gefunden. Bei Schrittmotoren wird üblicherweise der Winkel geregelt, zur genauen Realisierung wird das sogenannte Microstepping verwendet. Dabei entstehen aber schwankende Momente. Eine Vorgehensweise um die Vektorregelung beim Schrittmotor mit 2 Phasen umzusetzen wurde nicht gefunden, ist aber grundsätzlich möglich. Ein weiteres Problem ergibt sich aus der hohen Polpaarzahl. Bei 50 Polpaaren und einer einigermaßen guten Nachbildung des elektrischen Winkels mit 50 Stützstellen ergibt sich eine notwendige Decoderauflösung von 2500 Impulsen pro Umdrehung. Entsprechende Decoder sind wiederum teurer, dies macht den Preisvorteil des kostengünstigeren Schrittmotors zu nichte. Die notwendige, hohe Auflösung führt ebenfalls dazu, dass zur Steuerung schon bei geringen Drehzahlen sehr hohe Frequenzen gebraucht werden.

Pole $p = 50$

Stützstellen $s = 50$

$U = 500 \text{ U/min}$

Frequenz am Decoder:

$f = U / 60 * p * s = 20,8 \text{ kHz}$

3 Definition eines roboterfesten Koordinatensystems

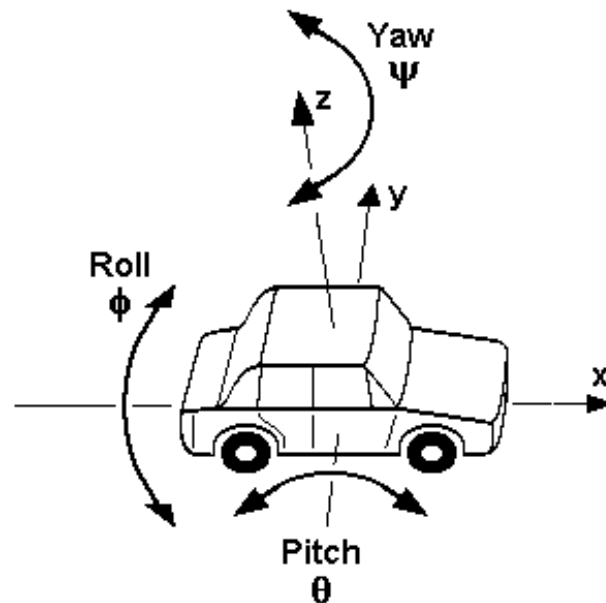


Abbildung 1: Roboterkoordinatensystem
(Quelle: [roboterKS_Bild])

Zur einheitlichen Bezeichnung wurde das roboterfeste Koordinatensystem (im Folgenden mit KS bezeichnet) wie in Abbildung 1 zu sehen, ähnlich ISO 88551 festgelegt. Entgegen der häufig anzutreffenden Bezeichnungen der Winkel um die jeweiligen Koordinatenachsen werden im Folgenden die Winkel um

- die x-Achse mit α
- die y-Achse mit β
- die z-Achse mit γ

bezeichnet.

Der Ursprung des KS liegt auf der Radachse mittig zwischen den beiden Rädern. Das KS ist ein rechtshändiges KS.

Die Verdrehung der jeweiligen Räder wird mit φ_l bzw. φ_r bezeichnet. Dabei bezeichnet l das in positive x-Richtung blickend links liegende Rad und r das in positive x-Richtung blickend rechts liegende Rad.

4 Navigation

Wie navigiert man in der Ebene?

4.1 Sensorik

Zur Verfügung stehen folgende Sensoren:

- Gyros in allen drei Raumachsen
- Beschleunigungssensoren in allen drei Raumachsen
- Inkrementalgeber der Räder
- Ultraschallsensoren

4.2 Inkrementalgeber

Zur Bestimmung der Position des Roboters in einem festen Koordinatensystem können die Inkrementalgeber der Antriebsmotoren benutzt werden.

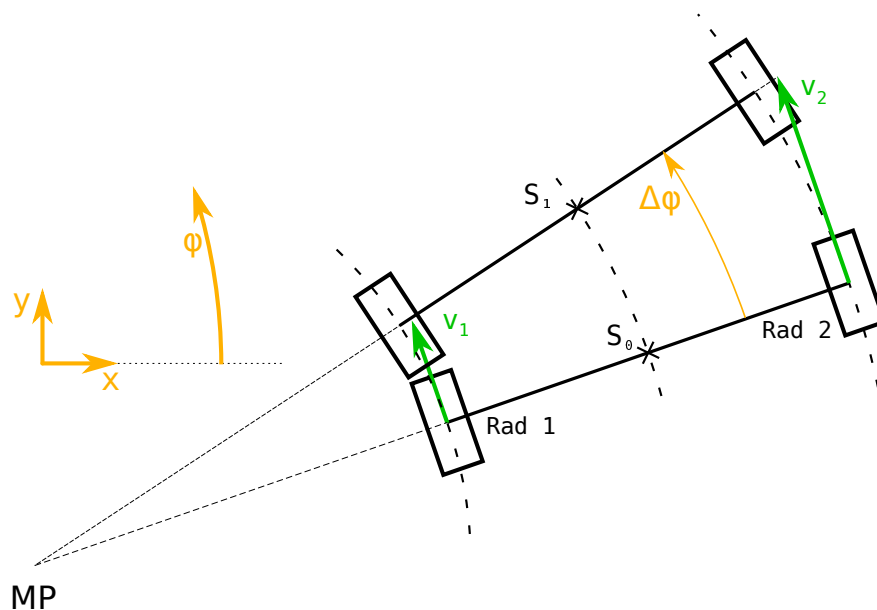


Abbildung 2: Rotation um den Momentanpol
(Quelle: eigene Darstellung)

Zur Ermittlung der aktuellen Position ist in Abbildung 2 die Kinematik einer Kurvenfahrt dargestellt, betrachtet für einen gedanklich sehr kleinen Zeitabschnitt. In diesem wird angenommen, dass die Geschwindigkeiten v_1 und v_2 beider Räder konstant bleiben. Dies sorgt für die Vereinfachung, dass der Roboter sich auf einer Kreisbahn um einen für die betrachtete Zeitspanne konstanten Momentanpol bewegt. Im Folgenden

sind die durch den Inkrementalgeber erfassten Bogenlängen mit arc bezeichnet. Es gilt:

$$arc_{R1} = \Delta\gamma \cdot r_{R1} \quad (1)$$

$$arc_{R2} = \Delta\gamma \cdot r_{R2} \quad (2)$$

$$r_{R1} = r_{R2} + l_a \quad (3)$$

$$\Delta arcs = arc_{R2} - arc_{R1} \quad (4)$$

$$r_S = r_{R2} - \frac{1}{2}l_a \quad (5)$$

Aus (1), (2), (3), (4) und (5) folgen:

$$\Delta\gamma = \frac{1}{l_a} \cdot \Delta arcs \quad (6)$$

$$r_S = \frac{arc_{R2}}{\Delta\gamma} - \frac{1}{2}l_a \quad (7)$$

Durch Aufsummieren von $\Delta\gamma$ nach jeder Auswertung der Inkrementalgeber kann somit ein ungefährender Absolutwinkel der Roboterachse zu einem festen KS berechnet werden.

Zur Bestimmung des Schwerpunktes in xy-Koordinaten wird die Verschiebung von S_0 zu S_1 mit trigonometrischen Funktionen berechnet und anschließend aufsummiert. Es gilt:

$$\vec{S}_1 = \vec{S}_0 + \begin{pmatrix} -\sin(\Delta\gamma) \cdot r_S \cdot \sin(\gamma_0) - (r_S - \cos(\Delta\gamma) \cdot r_S) \cdot \cos(\gamma_0) \\ \sin(\Delta\gamma) \cdot r_S \cdot \cos(\gamma_0) - (r_S - \cos(\Delta\gamma) \cdot r_S) \cdot \sin(\gamma_0) \end{pmatrix} \quad (8)$$

Die Berechnung im Mikrocontroller wurde wie in Listing 1 umgesetzt.

```
float deadReckonTotalPhi(int inkLeft, int inkRight) {
    static float totalGamma = 0.0f;
    static int fullTurns = 0;
    totalGamma = totalGamma + 1/l_a * (inkRight - inkLeft);

    // adjust for -2pi < totalGamma < +2pi
    if (totalGamma > 6.283185307) {
        totalGamma -= 6.283185307;
        fullTurns += 1;
    }
    else if (totalGamma < -6.283185307){
        totalGamma += 6.283185307;
        fullTurns -= 1;
    }
    return totalGamma;
}
```

```
}
```

Listing 1: deadReckonTotalPhi

5 Auslegung Kurvengeschwindigkeit

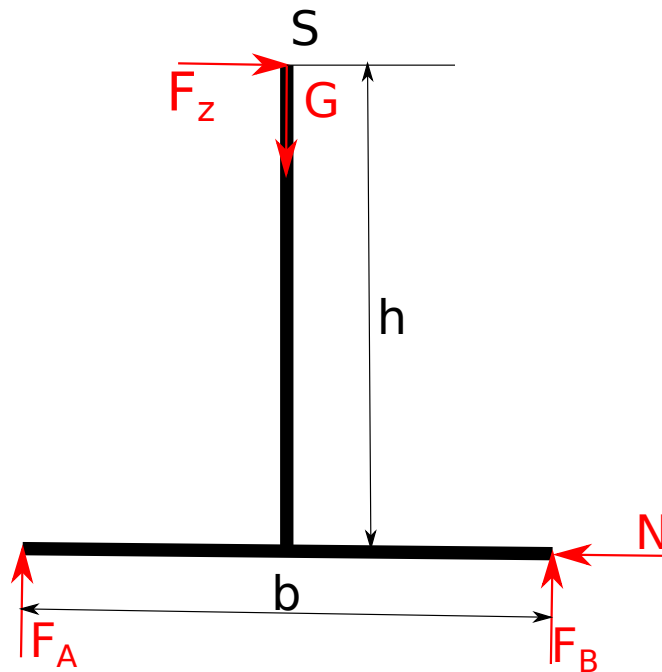


Abbildung 3: Kräfte bei der Kurvenfahrt
(Quelle: eigene Darstellung)

Bei der Kurvenfahrt darf der Roboter nicht seitlich umkippen. Um dies zu gewährleisten, muss das Verhältnis zwischen Geschwindigkeit und Kurvenradius passen. Dazu soll der innere Reifen noch mindestens die Hälfte der Last aufnehmen wie im Stand also ein Viertel der Gewichtskraft.

Es gilt:

$$F_A = \frac{1}{4} m \cdot g \quad (9)$$

$$G = m \cdot g \quad (10)$$

$$F_z = m \cdot \ddot{x} \quad (11)$$

$$N = m \cdot \frac{v^2}{r} \quad (12)$$

$$F_A + F_B = G \quad (13)$$

$$N = F_z \quad (14)$$

$$F_A \cdot b = F_z \cdot h \quad (15)$$

Hieraus folgt:

$$\ddot{x} = \frac{v^2}{r} \quad (16)$$

$$\frac{1}{4}m \cdot g \cdot b = m \cdot \frac{v^2}{r} \cdot h \quad (17)$$

Und daraus die Zusammenhänge von Geschwindigkeit zu Radius

$$v = \sqrt{\frac{gb}{4h} \cdot r} \quad (18)$$

$$r = \frac{4h}{gb} \cdot v^2 \quad (19)$$

Der notwendige Reibwert μ zwischen Reifen und Boden ergibt sich aus der Haftbedingung.

$$\mu \cdot mg = m \cdot \frac{v^2}{r} \quad (20)$$

$$\mu = \frac{v^2}{rg} \quad (21)$$