

VOR Sender und Empfänger - Dokumentation

Glossar / Begriffsdefinitionen

- VOR Quartal (quarter): 90° Segment des VOR
- VOR-Master: VOR-Sender, der den Nordimpuls vorgibt
- VOR-Slave(s): VOR-Sender, die dem Master folgen
- Nordimpuls: Funksignal, das den Beginn einer VOR-Umdrehung anzeigt
- Norden: bezeichnet nicht magnetisch oder geographisch Nord, sondern ist im Spielfeld definiert!
- IR-Strahl (IR beam): Umlaufender Lichtstrahl
- VOR Segmente (segments): Anzahl der umlaufenden Segmente (Anzahl IR-LEDs)

Die VOR-Technologie

Die hier verwendete Technologie zur Positionsbestimmung ist dem sog. VOR aus der Luftfahrt entlehnt. Hierbei werden bekannte Punkte angepeilt, um seine eigene Position berechnen zu können.

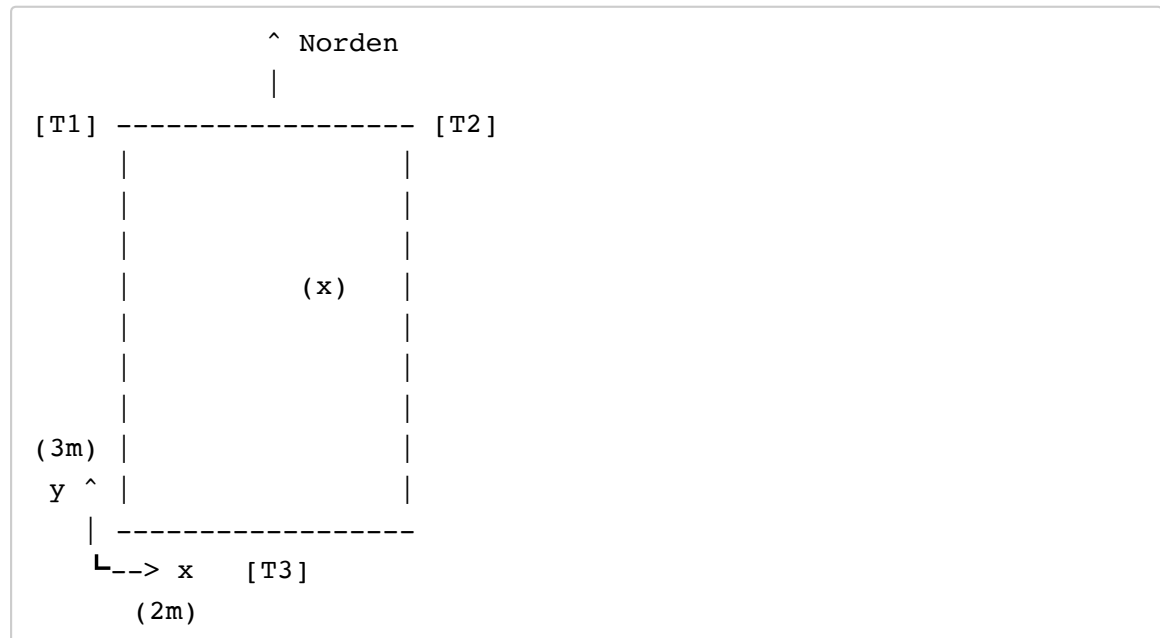
Damit nun aber keine Kompasspeilungen vorgenommen werden müssen, gibt es einen Trick: Die Funkfeuer/ Sendetürme senden ein Umlaufendes Signal mit einem relativ kleinen Abstrahlwinkel. In dieser Implementation wird das umlaufende Signal mit IR-LEDs realisiert. Um nun aus diesem umlaufenden Strahl eine Winkelinformation zu erhalten, muss es noch eine Referenz geben. Diese wird durch einen allseitig abgegebenen Nordimpuls realisiert. Nun kann die Zeit zwischen dem Nordimpuls und der Messung des Strahls gemessen werden, die proportional zum Abstrahlwinkel ist. Die Peilung ist also immer zum Empfänger hin.

Da die Positionen der Sendetürme bekannt sind, kann aus mindestens zwei Peilungen eine Position ermittelt werden.

Damit auch der Fall abgedeckt ist, dass ein Hindernis die Sicht auf einen Sendeturm verdeckt, gibt es drei Sendetürme.

Positionsberechnung

Spielfeld und Koordinatensystem



Zuordnung Winkel - Sendeturm

Je nach dem, aus welchem Winkel (bezogen auf Norden) der Strahl eintrifft, kann dieser eindeutig zu einem Sendeturm zugeordnet werden.

- a) 0°- 90°: Sender 3
- b) 90°-180°: Sender 1
- c) 180°-270°: Sender 2
- d) 270°-360°: Sender 3

Also muss in der Software die aus der Zeit berechnete Winkelangabe dem richtigen Sendeturm zugeordnet werden.

Berechnung

Prinzipiell ergibt sich aus der Geometrie erst einmal ein LGS mit drei Gleichungen.

$$\begin{aligned}\text{Sender 1: } g_1(x) &= m_1 \cdot (x) + 3 && \text{mit: } m_1 = \tan(90^\circ - \alpha_1) \\ \text{Sender 2: } g_2(x) &= m_2 \cdot (x-2) + 3 && \text{mit: } m_2 = \tan(90^\circ - \alpha_2) \\ &\Rightarrow g_2(x) = m_2 \cdot x - (2 \cdot m_2 - 3) \\ \text{Sender 3: } g_3(x) &= m_3 \cdot (x-1) + 0 && \text{mit: } m_3 = \tan(90^\circ - \alpha_3) \\ &\Rightarrow g_3(x) = m_3 \cdot x - (1 \cdot m_3)\end{aligned}$$

Andererseits kann nicht angenommen werden, dass sich alle drei Strahlen an einem Punkt schneiden, da schon die Anzahl der IR-LEDs nicht für eine hohe Genauigkeit ausreicht.

Daher müssen alle möglichen Schnittpunkte einzeln berechnet werden. Bei drei Sendern also drei Schnittpunkte, wenn ein Sender verdeckt ist, nur zwei Schnittpunkte.

Algorithmus

Zur Berechnung der Position müssen folgende Schritte durchgeführt werden:

- Zeit zwischen Nordimpuls und Empfang des IR-Strahls messen
- aus der Zeit den entsprechenden Winkel ermitteln
- aus dem Winkel (oder der Zeit) auf den Sendeturm zurückschließen
- aus dem Winkel und dem Sendeturm ergibt sich eine Geradengleichung
- mit mind. zwei Geraden lässt sich eine Position errechnen

Ermittlung des Winkels:

Bei einer Umlauffrequenz des IR-Strahls von `frequenz` Hz und einer Timer-Zeit (seit dem Nordimpuls) von `timer` Mikrosekunden ergibt sich ein Winkel `winkel` in Grad nach:

$$\text{winkel} = 360 * \text{timer} / (1000000/\text{frequenz})$$

Berechnung der Geradenparameter:

Die Geradenparameter ergeben sich aus den oben genannten drei Gleichungen.

Allgemein gilt jedoch mit einer Turmposition von `(pos_x | pos_y)` und einem Winkel `alpha`:

Bekannt: m (abhängig von α)

$$\begin{aligned} g(x) &= m * (x - \text{pos}_x) + \text{pos}_y \\ \Rightarrow g(x) &= m * x - m * \text{pos}_x + \text{pos}_y = \underset{\text{Steigung}}{m * x} + \underset{\text{Y-Achsenabschnitt}}{(\text{pos}_y - m * \text{pos}_x)} \end{aligned}$$

$$\begin{aligned} \Rightarrow m &= \tan(90^\circ - \alpha) \\ b &= \text{pos}_y - \tan(90^\circ - \alpha) * \text{pos}_x \end{aligned}$$

Schnittpunkt zweier Geraden:

Der Schnittpunkt zweier Geraden lässt sich berechnen, wenn die Steigung und der Y-Achsenabschnitt bekannt sind. Die Indizes `_1` und `_2` sollen hier die beiden Geraden unterscheiden.

für Geradengleichungen der Form: $y(x) = m * x + b$

$$\begin{aligned} x &= \frac{b_2 - b_1}{m_1 - m_2} \\ y &= \frac{m_1 * b_2 - m_2 * b_1}{m_1 - m_2} \end{aligned}$$

Implementierung als C Funktion:

```
void CalcIntersection (float m_1, float m_2, float b_1, float b_2, float *p_x, float *p_y)
{
    *p_x = (b_2 - b_1) / (m_1 - m_2);
    *p_y = ((m_1 * b_2) - (m_2 * b_1)) / (m_1 - m_2);
}
```

Mittelung der Werte

Wenn ein Signal von allen drei Türmen empfangen wird, müssen auch drei Schnittpunkte der Geraden berechnet werden. Für die Berechnung des Mittelwertes bzw. -punktes gibt es potenziell zwei Methoden:

- Der Flächenschwerpunkt des Dreiecks
- jeweils der Mittelwert der X- und Y-Koordinaten der drei Punkte

Wir haben uns für die einfache Methode - Mittelwert errechnen - entschieden. Der Flächenschwerpunkt spiegelt zwar für das ermittelte Dreieck besser den Mittelunkt wieder, hilft jedoch nicht unbedingt dabei der realen Position näher zu kommen.

Nachdem nun nur noch ein Punkt zur Verfügung steht, wird dieser noch einmal über 5 Werte gemittelt bevor die Koordinaten an das Fahrzeug übergeben werden. So wird eine Datenrate von 2 Hz erreicht. Weitere Erklärungen dazu im nächsten Kapitel "Programmablauf".

Programmablauf

Diese Parameter können eingestellt werden und werden im folgenden als Platzhalter verwendet:

- AVG_ANGLE 3
- AVG_VALUES 2

Um eine vorhersehbare Ausführungszeit zu haben, wird folgender Ablauf verwendet:

- (AVG_ANGLE-1) Nordimpuls-Perioden Daten sammeln
- danach eine Nordimpuls-Periode die Daten verarbeiten

So werden Schwankungen in der Ausführungszeit, während die Daten gesammelt werden, vermieden - die Abtastzeit der IR-Signale bleibt also gleich.

Hauptprogramm

- Wenn Nordimpuls
 - dann Timer starten
 - Counter++
- Wenn Counter im Intervall [1, (AVG_ANGLE-1)]
 - Wenn irgendeine IR-LED empfangen
 - aktuellen Timerwert auslesen
 - Winkel aus Timer berechnen
 - aus dem Winkelbereich den korrekten Turm ermitteln
 - Winkel in die Turminstanz speichern
- Wenn Counter = AVG_ANGLE
 - Wenn mind. 2 Türme Werte haben
 - Geradenparameter von der Turminstanz erhalten
 - Schnittpunkt(e) errechnen
 - Werte über AVG_VALUES Mal sammeln
 - wenn AVG_VALUES gesammelt: Werte übertragen/ ausgeben
 - sonst: weiter sammeln
 - Counter = 0

C++-Klasse für die VOR-Sender (Türme)

Die Turm-Klasse mittelt alle erhaltenen Winkel, wenn ein neuer Winkel eingegeben wird.

Erst beim Abrauf der Geradenparameter werden diese aus dem gemittelten Winkelwert errechnet. Dann wird auch der Winkel wieder zurückgesetzt

Werteausgabe

Aufgrund der Nordimpuls-Frequenz von 50 Hz werden auch mit dieser Frequenz neue Daten gesammelt. Zur Glättung werden jeweils `(AVG_ANGLE-1)` Werte (in `AVG_ANGLE` Zyklen) zusammengefasst.

Wie oben schon kurz beschrieben, können die errechneten Positionen noch einmal über `AVG_VALUES` Werte gemittelt werden, bevor diese ausgegeben werden.

Zusätzlich kann mit `EN_FILTER_ARR` eine Filterung von unplausiblen Werten dazugeschaltet werden. Diese Option verwendet einen gleitenden Durchschnitt, daher sollte dann die Mittelung der Werte mit `AVG_VALUES` relativ klein eingestellt werden, da sonst die Berechnung für die Bewegung des Fahrzeugs zu träge ist.

Bei der Filterung wird mit `filterErrorCnt` noch geprüft, ob die Position zu häufig außerhalb der Toleranz lag. Dadurch wird verhindert, dass die Filterung an einer "falschen" Position festhängt. Der Ablauf ist dann wie folgt:

- wenn `filterErrorCnt < FILTER_ERR_THRES`
 - Mittelwert über das `filterArray` bilden
 - wenn Position weniger als `FILTER_DIFF_MAX` vom Mittelwert abweicht:
 - Werte im `filterArray` um eine Stelle nach hinten schieben
 - Position für die Ausgabe bleibt die aktuell errechnete Position
 - `filterErrorCnt = 0`
 - sonst:
 - Mittelwert für die Ausgabe verwenden
 - `filterErrorCnt` hochzählen
- wenn `filterErrorCnt >= FILTER_ERR_THRES`
 - `filterArray` auf die aktuelle Position setzen --> d.h. reset

Die Serielle Datenausgabe (bzw. Datenübergabe) wurde von dem Team für das Fahrzeug entwickelt und getestet.

Hardware

Dieses VOR-System besteht aus drei Sendern, und zwei Empfängern. Die Sender sind dabei miteinander gekoppelt und decken die 360° nur ein Mal ab. Es können beliebig weitere Empfänger hinzugefügt werden.

Sender Hardware

Der Sender besteht aus:

- Arduino nano
- 433 MHz Sender
- (ein oder zwei) 8-Bit Schieberegister 74HC595 ??
- (8 oder 16) IR-LEDs mit 40 kHz Modulation
- (8 oder 16) Transistoren für die LEDs

Empfänger Hardware

Der Empfänger besteht aus

- Arduino nano
- 433 MHz Empfänger
- (8 oder 16) IR-Empfänger mit 40 kHz Modulation

433MHz Strecke

Der Nordimpuls wird über eine 433 MHz Funkstrecke übertragen. Diese hat eine Verzögerung zwischen Sendereingang und Empfängerausgang von 40µs.

Schnittstelle zum Roboter

Dem Roboter werden die Positionsdaten über die Serielle Schnittstelle (UART) des Arduino übergeben.

Das Datenformat ist folgendes:

- Startzeichen @
- X-Koordinate (in cm) als 16 Bit unsigned int (zwei Bytes nacheinander)
- Y-Koordinate (in cm) als 16 Bit unsigned int (zwei Bytes nacheinander)

```
Serial.print("@");  
Serial.write((char)x>>8); Serial.write((char)x);  
Serial.write((char)y>>8); Serial.write((char)y);
```