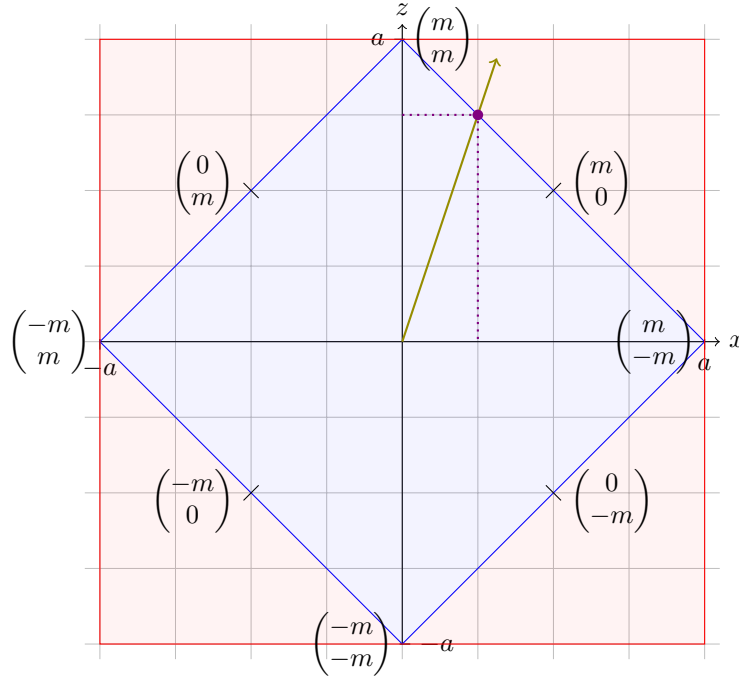


T.H.O.M.A.S. Umrechnung von Joystick-Auslenkungen in Motorwerte



a ist die Maximalauslenkung des Joysticks. Da dieser Werte zwischen -32767 und 32767 annehmen kann, ist $a = 32767$.

Die Motorgeschwindigkeit lässt sich als Vektor angeben: $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$, wobei $-m \leq v_1, v_2 \leq m$.
 v_1 sei hierbei der linke Motor, v_2 der rechte.

m ist hierbei der Maximalwert der Motorsteuerung, also $m = 255$, da die Motoren nur Werte zwischen einschließlich -255 und 255 akzeptieren.

Die Joystick-Auslenkungen im blau eingefärbten Bereich lassen sich nun leicht in Motorgeschwindigkeiten umrechnen:

$$f \begin{pmatrix} x \\ z \end{pmatrix} = \frac{m}{a} \cdot \begin{pmatrix} x + z \\ -x + z \end{pmatrix}$$

Ein größeres Problem stellt der rot markierte Bereich dar, denn für diesen definiert unser Modell keine Geschwindigkeitswerte, bzw. erzeugt ungültige. Der einfachste Weg ist hier, sämtliche Auslenkungen im roten Bereich einfach

auf den blauen Bereich zu reduzieren. Dazu wird einfach eine Linie (olivgrün in der Zeichnung) zum Ursprung gezogen und der Geschwindigkeitswert genommen, der am Schnittpunkt (violetter Punkt) der Linie mit der Grenze des blauen Bereichs liegt.

Das führt dann zu folgender Berechnung der Motorgeschwindigkeit für alle Auslenkungen $\begin{pmatrix} x \\ z \end{pmatrix}$, die nicht im blauen Bereich liegen:

$$g \begin{pmatrix} x \\ z \end{pmatrix} = \frac{m}{|x| + |z|} \cdot \begin{pmatrix} x + z \\ -x + z \end{pmatrix}$$

Offensichtlich wird g also nur verwendet, wenn der Joystick im „roten Bereich“ ist, ansonsten f . Wie kann also erkannt werden, wann f und wann g verwendet werden muss?

Die Antwort ist bereits in der Formel von g : Wenn die Summe $|x| + |z|$ größer als a wird, befinden wir uns im roten Bereich.

Warum? Betrachte die blauen Linien, die die Kanten des blauen Bereichs darstellen.

Auf diesen Linien gilt immer: $|x| + |z| = a$.

Innerhalb der blauen Fläche gilt: $|x| + |z| < a$.

Die Betragsstriche sind hier erforderlich, damit bei den negativen Koordinaten keine merkwürdigen Dinge geschehen.

Und damit wäre der mathematische Teil erledigt, schauen wir also einmal kurz ins Programm:

[MotorControl.h]

```
const float _joystickMaxAxis = 32767.0f; // = a
const float _joystickScale = 255.0f; // = m
const float _joystickNoPowerZone = _joystickScale * 0.1f;
const float _joystickAxisInvConv = _joystickScale /
    _joystickMaxAxis; // = m/a
```

`_joystickNoPowerZone` ist der Bereich, in dem die R-Achse berücksichtigt wird. Der Faktor 0.1 ist hier gewählt, damit die R-Achse bei bis zu 10% Joystick-Auslenkung berücksichtigt werden kann (der Joystick kann nur schwierig ganz exakt in der Mitte gehalten werden, deshalb die 10% Toleranz).

Der restliche Code sollte selbsterklärend sein, wenn statt der mathematischen Symbole die gegebenen Variablennamen eingesetzt werden.

Ein Hinweis zur „Korrektursumme“: Hierbei handelt es sich einfach um die Summe $|x| + |z|$.

[MotorControl.cpp]

```
// Joystick-Korrektursumme berechnen
corrSum = static_cast<float>(abs(_joystickAxis[0]) + abs(
    _joystickAxis[1]));
```