

Programmablauf

Als Eingabewerte erhält das Programm die Winkel $\alpha_{1,2}$ und $\beta_{1,2,3}$. Diese Winkel haben wir aufgrund der Formel (1) (Seite 1) bestimmt.

Für die Bestimmung der Winkel haben wir die Bildmaße des Aufgabenblattes auf A4 ausgedruckt zugrunde gelegt und mittels Formel (2) (Seite 1) bestimmt.

Das Programm ermittelt nun mit Hilfe der Winkel BETA 1,2,3 plus dem Fehlerintervall $I=[-0,25^\circ; +0,25^\circ]$ die fehlerbehafteten Entfernungen $F_{1,2,3}$ zu den Fahnen (Bild: links, nach vorn, rechts). Leider konnten wir aus Zeitgründen nicht mehr die Formel X (siehe Punkt Fehlerbetrachtung) mit einfließen lassen, so dass sich $F_{1,2,3}$ nur nach der Formel X (siehe Seite 1) berechnet.

Analog wird mit den Winkeln ALPHA 1,2 verfahren.

Darauf werden alle 5 maximal konsistenten Einschränkungen (die plus Fehlerabweichung der 5 Radian betreffend) der Roboterposition um die 3 Fahnen sowie 2 Mittelpunkte $M_{1,2}$ einzeln bestimmt. Der Schnitt aller 5 Flächen ist unsere maximal globale Einschränkung, die im Folgenden eingeschränkt wird.

Die Einschränkung wird dadurch realisiert, dass wir alle Schnittpunkte aller maximalen (durch Fehlerabweichung) Radian mit allen anderen maximalen Radian und alle Schnittpunkte aller minimalen (durch Fehlerabweichung) Radian mit allen minimalen Radian sammeln und vergleichen. Ausgenommen wurden die Schnittpunkte der Kreise um $M_{1,2}$.

Nun werden nur die x- und y-Werte herausgezogen, die die anfänglich bestimmte maximal globale Einschränkung am wenigsten einschränken.

Hier liegt auch leider ein großes Problem, da wir aus Zeitgründen keinen Algorithmus entwerfen konnten der die „Fehlerflächen“, in denen sich der Roboter befindet, schneiden kann. Ergo könnte man aus unseren Daten noch mehr an Information gewinnen um die Roboterposition wirklich weiter einschränken zu können.

Unser Ergebnis liegt bei einer Feldgröße von 579mm x 867mm.

Randbemerkungen:

Das C++ Programm wurde mit Visual Basic 2005 gefertigt und auf Suse 10 getestet.