

Constraints

Gruppe: Bort

Martin Schröder (502301)

Frank Bicking (504628)

1 Aufgabe A

Zunächst einmal werden folgende Festlegungen bezüglich des Roboters und Feldes getroffen:

1. Der 0° Winkel der Kamera befindet sich horizontal wie vertikal genau in der Mitte des Bildes.
2. Der Koordinatenursprung der X- und Y-Achse des Feldes liegt in der oberen linken Ecke.
3. Das Kamerabild wird insofern als gefiltert angenommen, als dass die Drehung des Kopfes herausgerechnet wird, d.h. dass die Horizontlinie immer gerade und horizontal ist.

Die Stellung des Roboters wird durch drei unabhängige Variablen eindeutig bestimmt: Horizontale und vertikale Position sowie die Drehrichtung. Im folgenden sei die horizontale Position durch die Variable X , die vertikale durch die Variable Y und die Drehrichtung durch die Variable ψ bestimmt. Durch die auf Abbildung 5 auf Seite 10 wie folgt gegebene Feldgeometrie ergeben sich dann die Constraints denen X , Y und ψ unterliegen.

Die Constraints lassen sich zunächst einmal in zwei Kategorien einteilen: Constraints für die XY-Achse und Constraints für den Winkel ψ . Wir werden im folgenden diese Constraints nacheinander angeben, und begründen.

1.1 Constraints für den Winkel ψ

Zunächst lässt sich feststellen, dass die XY Position des Roboters nicht ausreicht, um die Stellung des Roboters komplett zu bestimmen. Daher muss man zusätzlich noch den Drehwinkel des Roboters auf dem Feld bestimmen. Diese wird mit dem Winkel ψ angegeben, wobei 0° als der Winkel angesehen wird, bei dem der Roboter parallel zur horizontalen Achse des Spielfelds steht und nach rechts in Richtung des gelben Tores schaut. Der Winkel steigt dabei im Uhrzeigersinn an. Insgesamt taugen 6 Landmarken als die einfachsten und direktesten Arten den Winkel ψ zu bestimmen: Die vier Flaggen, die

beiden Tore und der Mittelkreis. Die Linien auf dem Feld sind nicht ohne weiteres für die Bestimmung des Winkels geeignet, da sie zu viele Symmetrien ermöglichen. Erst die Orte an denen sich zwei Linien senkrecht treffen sind für die Erkennung des Winkels geeignet, jedoch schwieriger als die anderen Landmarken zu nutzen, wie man den uns gegebenen Probeaufnahmen deutlich erkennen kann. Insgesamt würde man dadurch noch 14 weitere Landmarken gewinnen. Da wir jedoch davon ausgehen, dass man immer mindestens eine der 7 zuerst genannten Landmarken sieht, werden wir uns auf diese beschränken.

Zuerst bestimmen wir die Winkel zwischen Roboter und Landmarke als relativen Wert. Dadurch ergibt sich für den Peilwinkel ϕ :

$$C_\phi = (\phi : \phi = 27,5^\circ - \frac{b \cdot \lambda}{r'} + C_angle) \quad (1)$$

Dazu die folgende graphische Veranschaulichung:

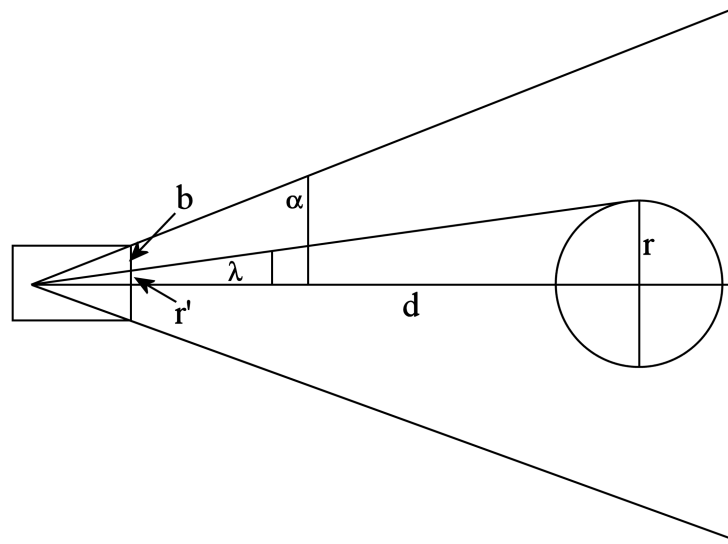


Abbildung 1: Variablenlegende für Flaggen

Hierbei ist b die Anzahl an Pixeln zwischen dem linken Rand des Bildes und der Mitte der Landmarke, λ der halbe Winkelanteil der Flagge am Bild¹ und r' die Hälfte der Breite der Flagge auf dem Bild in Pixeln. " C_angle " ist die Schwenkrichtung der Kamera, wobei vorne 0° , links 90° und rechts -90° ist. Damit ergibt sich der Winkel vom linken Rand des Bildes zur Mitte der Flagge.

Die Erklärung für diese Formel bieten die Winkelsätze, nach denen gilt: $\frac{r'}{\lambda} = \frac{b}{\alpha}$. Durch Umstellung erhalten wir: $\alpha = \frac{b \cdot \lambda}{r'}$. Da wir festgelegt haben, dass 0° der Kamera in der Mitte des Bildes liegen, müssen wir anschließend noch $27,5^\circ - \alpha$ berechnen, um den Winkel ins Verhältnis zum Kameramittelpunkt zu setzen, da die Kamera ja einen horizontalen Öffnungswinkel von 55° aufweist.

¹Also die Breite der Flagge auf dem Bild in Pixeln geteilt durch die Gesamtbreite des Bildes in Pixeln multipliziert mit dem Öffnungswinkel der Kamera, hier: 55° , wobei das Ergebnis durch zwei geteilt wird.

Wie man auf Abbildung 1 sehen kann, indem man den Kreis zu einer Ellipse verformt indem man die horizontale “Tiefe” des Kreises verringert, gilt diese Formel nicht nur für die Flaggen, sondern auch für den Kreismittelpunkt. Obwohl ihm eine wichtige Eigenschaft der Flaggen fehlt, nämlich die im wesentlichen gleichbleibende Form unabhängig vom Betrachtungswinkel und -distanz, können wir dennoch dieselbe Formel nutzen, da sich seine Form nur in der Tiefe, jedoch nicht in der Breite ändert, wenn sich Winkel oder Distanz ändert.

Anders verhält es sich mit dem Toren. Hierbei wird das Tor je nach Winkel mehr oder weniger gestaucht. Dennoch lässt sich auch hier der Peilungswinkel relativ leicht bestimmen. Zunächst jedoch dazu eine Skizze:

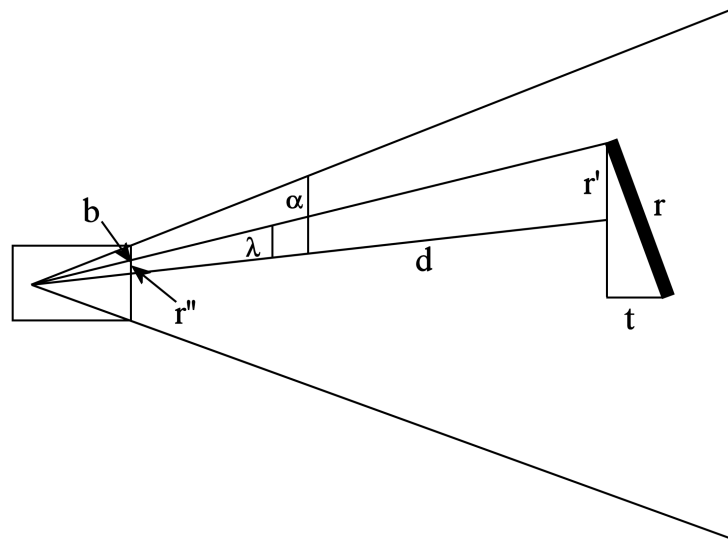


Abbildung 2: Variablenlegende für Tore

Wie man sieht, stimmt die abgebildete Längen r' und r'' und die echte Länge r nicht mehr überein, was den Mittelpunkt des Objekts das wir zur Peilung nutzen wollen verschiebt. Diesen Unterschied können wir jedoch in den meisten Fällen ignorieren, da der Fehler der dadurch entsteht nur bei sehr großen Winkeln zum Lot auf das Tor wirklich zum tragen kommt. Dementsprechend setzen wir hier einfach $r' = r$ und nutzen erneut denselben Constraint auf den Peilungswinkel.

Aus dem so berechneten Peilungswinkel ϕ eines Objektes können wir aber noch nicht den globalen Drehwinkel ψ bestimmen. Man könnte sich nämlich um eine Landmarke drehen, ohne notwendigerweise den Peilungswinkel zu ändern. Zur Festlegung des globalen Drehwinkels benötigen wir also mindestens die Peilwinkel zweier Objekte, oder den Peilwinkel eines Objektes und die momentane Position auf dem Feld. Wie wir im nächsten Abschnitt sehen werden, lässt sich die momentane Position selbst dann bestimmen, wenn wir unseren globalen Drehwinkel noch nicht kennen. Daher können wir hier einfach annehmen, dass uns die Position des Roboters bereits bekannt ist. Sobald wir dies festgelegt haben, ergeben sich für den Winkel ψ folgende Constraints, wobei $\phi(X)$

die Funktion ist, die den Peilungswinkel zur jeweiligen Landmarke angibt und x bzw. y die horizontale und vertikale Position des Roboters darstellt.

$$C_{\psi_1} = (\psi : [(y \leq 2000 \wedge \psi = 270^\circ - \tan^{-1}(\frac{x}{2000 - y}) - \phi(GoalBlue)) \vee (y > 2000 \wedge \psi = 90^\circ + \tan^{-1}(\frac{x}{2000 + y}) - \phi(GoalBlue))]) \quad (2)$$

Die Begründung für diese Formel findet man durch Betrachtung von Abbildung 6 auf Seite 10. Je nachdem ob wir uns ‘über’ oder ‘unter’ dem Tor befinden, ergibt sich eine leicht modifizierte Berechnung des globalen Drehwinkels, was das ‘oder’ erklärt. Ebenso kleine Modifikationen ergeben die Constraints für die restlichen Landmarken, die ohne Abbildung verbleiben.

$$C_{\psi_2} = (\psi : [(y \leq 2000 \wedge \psi = 270^\circ + \tan^{-1}(\frac{6000 - x}{2000 - y}) - \phi(GoalYellow)) \vee (y > 2000 \wedge \psi = 90^\circ - \tan^{-1}(\frac{6000 - x}{2000 + y}) - \phi(GoalYellow))]) \quad (3)$$

$$C_{\psi_3} = (\psi : [(x \leq 1650 \wedge \psi = \tan^{-1}(\frac{y}{1650 - x}) - \phi(FlagNW)) \vee (x > 1650 \wedge \psi = 180^\circ - \tan^{-1}(\frac{y}{x - 1650}) - \phi(FlagNW))]) \quad (4)$$

$$C_{\psi_4} = (\psi : [(x \leq 4350 \wedge \psi = \tan^{-1}(\frac{y}{4350 - x}) - \phi(FlagNE)) \vee (x > 4350 \wedge \psi = 180^\circ - \tan^{-1}(\frac{y}{x - 4350}) - \phi(FlagNE))]) \quad (5)$$

$$C_{\psi_5} = (\psi : [(x \leq 1650 \wedge \psi = 360^\circ - \tan^{-1}(\frac{4000 - y}{1650 - x}) - \phi(FlagSW)) \vee (x > 1650 \wedge \psi = 180^\circ + \tan^{-1}(\frac{4000 - y}{x - 1650}) - \phi(FlagSW))]) \quad (6)$$

$$C_{\psi_6} = (\psi : [(x \leq 4350 \wedge \psi = 360^\circ - \tan^{-1}(\frac{4000 - y}{4350 - x}) - \phi(FlagSE)) \vee (x > 4350 \wedge \psi = 180^\circ + \tan^{-1}(\frac{4000 - y}{x - 4350}) - \phi(FlagSE))]) \quad (7)$$

Es existiert in diesem Ansatz noch ein 7. Constraint, der für den Mittelpunkt, aber dieser hat nicht weniger als 4 Fallunterscheidungen, je nachdem in welchem der 4 Quadranten des Feldes der Roboter steht. Da sein Aufbau im wesentlichen klar sein sollte, können wir ihn daher vorerst ignorieren. Als Anmerkung sei eingefügt, dass man die Berechnungen auch alle auf eine Formel bringen kann, indem man zunächst X und Y so transformiert, als befände sich der Roboter im 1. Quadranten um dann anschließend den so erhaltenen Winkel zu jeweils 0°, 90°, 180° und 270° hinzuaddiert bzw. subtrahiert.

1.2 Constraints für die XY-Achse

Wie man auf der Abbildung 5 auf Seite 10 erkennt, unterliegen die X und Y Position des Roboters zunächst einmal folgenden trivialen Constraints, die sich aus der Geometrie

des Spielfelds, unabhängig von der Position des Roboters ergeben.

$$C_X = (X : 0 < X < 6000) \quad (8)$$

$$C_Y = (Y : 0 < Y < 4000) \quad (9)$$

Zu jeder Landmarke kann man zunächst nur die Distanz ermitteln. Wie man in Abbildung 1 auf Seite 2 erkennt, gilt für die Flaggen und den Mittelkreis folgendes:

$$\tan(\lambda) = \frac{r}{d} \Rightarrow d = \frac{r}{\tan(\lambda)}$$

Dabei kann man λ dadurch ermitteln, indem man die Breite des Objekts in Pixeln (r' bzw. r'') durch die Gesamtbreite des Bildes in Pixeln teilt und dann mit der Hälfte des Öffnungswinkels (hier: $27,5^\circ$) multipliziert. Bei den Toren als Landmarken ergibt sich das Problem, dass die Abbildung des Tores nicht dieselbe Form haben muss wie das Tor selbst, wie man auf Abbildung 2 erkennt. Da der Unterschied zwischen beiden Werten jedoch erneut nur bei großen Winkeln zum Lot auf das Tor zum Tragen kommt, ignorieren wir diesen Fehler zunächst und berechnen die Distanz d genauso wie bei den Flaggen, wobei wir r' auf r setzen.

Da wir nunmehr die Distanz zu einer jeden aufgezeichneten Landmarke kennen, können wir Constraints erstellen, die die Geometrie des Spielfeldes ausnutzen, um die X und Y -Werte zu beschränken, wobei $d(\text{Marke})$ die soeben berechnete Distanz (in *Millimetern*) zu der jeweiligen Landmarke ist.

$$C_{XY1} = (X, Y : d(\text{GoalBlue}) = \sqrt{X^2 + (2000 + Y)^2}) \quad (10)$$

$$C_{XY2} = (X, Y : d(\text{GoalYellow}) = \sqrt{(6000 + X)^2 + (2000 + Y)^2}) \quad (11)$$

$$C_{XY3} = (X, Y : d(\text{FlagNW}) = \sqrt{(1650 + X)^2 + Y^2}) \quad (12)$$

$$C_{XY4} = (X, Y : d(\text{FlagNE}) = \sqrt{(4350 + X)^2 + Y^2}) \quad (13)$$

$$C_{XY5} = (X, Y : d(\text{FlagSW}) = \sqrt{(1650 + X)^2 + (4000 + Y)^2}) \quad (14)$$

$$C_{XY6} = (X, Y : d(\text{FlagSE}) = \sqrt{(4350 + X)^2 + (4000 + Y)^2}) \quad (15)$$

$$C_{XY7} = (X, Y : d(\text{Middle}) = \sqrt{(3000 + X)^2 + (1800 + Y)^2}) \quad (16)$$

Es sei angemerkt, dass diese Lösung nur bei hohen Auflösungen der Kamera genaue Lösungen für die Constraints bietet, da die Distanzmessung stark von der zur Verfügung stehenden Auflösung abhängt und somit mit einem hohen Fehler gerechnet werden muss. Bei niedrigen Auflösungen empfiehlt es sich daher, eher aus den Peilungswinkeln zu drei Landmarken die Position zu berechnen, was hier aber nicht weiter ausgearbeitet wurde.

2 Aufgabe B

Auf den uns vorgegebenen Aufnahmen sind drei der in Aufgabe A als zweckmäßig herausgearbeiteten Landmarken zu erkennen. Dabei handelt es sich:

1. im um 90 Grad nach links gerichteten Bild um die blau/magenta markierte Flagge im südwestlichen Bereich des Spielfeldes (im Folgenden mit "FlagSW" abgekürzt),
2. im nach vorne gerichteten Bild um die magenta/blau markierte Flagge im nordwestlichen Bereich ("FlagNW"),
3. und schlussendlich im um 90 Grad nach rechts gerichteten Bild um die magenta/gelb markierte Flagge im nordöstlichen Bereich ("FlagNE").

Für alle drei Fälle ist also der Peilungswinkel zu einer Flagge zu ermitteln. Die zu diesem Zweck benötigte Formel haben wir bereits in Teilaufgabe A hergeleitet:

$$C_\phi = (\phi : \phi = 27,5^\circ - \frac{b \cdot \lambda}{r'} + C_angle)$$

$$\lambda = \frac{r' \cdot \text{Oeffnungswinkel}}{2 \cdot \text{Bildbreite}}$$

Die zur Berechnung von λ erforderliche Bildbreite (280 Pixel) und der horizontale Öffnungswinkel der Kamera (55°) sind vorgegeben.

$$C_\phi = (\phi : \phi = 27,5^\circ - \frac{b \cdot 55^\circ}{2 \cdot 208px} + C_angle)$$

Zur Berechnung des Peilungswinkels ist demzufolge b , der Abstand zwischen dem rechtem Bildrand und der Mitte der Flagge abzuschätzen. r' fällt hingegen komplett heraus, wir messen diesen Wert aber trotzdem, da wir ihn bei der Distanzmessung nochmals benötigen. Dass die vorgegebenen Bilder unserer dritten in Aufgabe A getroffenen Festlegung (die Horizontlinie ist gerade) nicht genügen, beheben wir durch Einsatz einer Bildbearbeitungssoftware per hand. Dadurch erhalten wir folgende Werte von b :

Richtung	Landmarke	b	r'
90° links	FlagSW	115px	10px
Mitte	FlagNW	44px	13px
90° rechts	FlagNE	75px	10px

Daraus ergeben sich nun folgende Peilungswinkel die uns nach den Formeln der Aufgabe A der Imageprozessor vermutlich liefern würde.

$$C_\phi(\text{FlagSW}) = (\phi : \phi = 27,5^\circ - \frac{115 \cdot 55^\circ}{2 \cdot 208px} + 90^\circ = 102,296^\circ)$$

$$C_\phi(\text{FlagSW}) = (\phi : \phi = 27,5^\circ - \frac{44 \cdot 55^\circ}{2 \cdot 208px} + 90^\circ = 21,683^\circ)$$

$$C_\phi(\text{FlagSW}) = (\phi : \phi = 27,5^\circ - \frac{75 \cdot 55^\circ}{2 \cdot 208px} + 90^\circ = -72,416^\circ)$$

Wie man sieht, und wie es bereits in Aufgabe A erwähnt wurde, sind alle diese Werte aufgrund der Rasterung durch Pixel mit einem mehr oder minder großen Fehler behaftet. So sorgt eine Abweichung von einem Pixel bei der Messung der Entfernung vom rechten Bildrand zur Mitte der Flagge bereits für eine Abweichung von $0,13^\circ$ beim berechneten Peilungswinkel, was sich anschließend direkt auf den Drehwinkel auswirkt. Noch viel verheerender ist hingegen der Fehler bei der Distanz. Hier bewirkt ein Pixel mehr von 13 auf 12 Pixeln bei 208 Pixeln Bildbreite bereits eine Erhöhung der Distanz von 1749,6 auf 1895,5 Millimeter.

Aufgrunddessen müssen wir bei der Implementation der Constraints aus Aufgabe A darauf achten, ein passendes Fehlermaß für die errechneten Winkel und vor allem für die errechnete Distanz zur Landmarke einzuführen. Bei der Distanz wäre es also anzuraten, den Fehler so zu wählen, dass die Distanz eher über- als unterschätzt wird, d.h. man nimmt die gemessenen Pixelwerte, errechnet die Distanz wie in Aufgabe A und erhöht die berechnete Distanz dann um einen bestimmten Prozentsatz. Alternativ dazu kann man den Fehler auch getrennt auf b , r' und den Peilungswinkel ϕ anwenden. Ähnliches gilt für die Berechnung des Drehwinkels aus Peilungswinkel und der errechneten Position, auch dort müssen wir ein passendes Fehlermaß wählen und den Winkelspielraum zu erhöhen, um nicht in die Lage zu kommen durch zu kleine Constraints eventuell gar keine Position bzw. gar keinen Drehwinkel zu erhalten.

3 Aufgabe C

Folgende Anmerkungen seien zu Kompilierung unseres Constraint-Propagierungsmechanismus gemacht: Das Programm ist in Java 1.5 unter Netbeans 5.0 bzw. 5.5 geschrieben worden. Eine fertig kompilierte *.jar* Datei liegt der Abgabe bei und kann durch den Befehl:

```
java -jar ConstraintProp.jar
```

ausgeführt werden. Der abgegebene Quellcode ist ein handangepasster Export aus NetBeans und sollte durch jeden Java 1.5 Compiler mit der folgenden Anweisungen kompilier- und ausführbar sein, die man im Verzeichnis ausserhalb des Ordners “constraintprop” ausführt

```
javac constraintprop\Interval.java
javac constraintprop\Constraint.java
javac constraintprop>Main.java
```

```
java constraintprop/Main
```

Es ist von äußerster Wichtigkeit, dass diese Anweisungen direkt von ausserhalb des Verzeichnisses “*constraintprop*” ausgeführt werden, da diese Klassen in einer Package liegen. Der Einfachheit halber liegen der Zip-Datei in der dieses Dokument zu finden gewesen sein sollte der Unterordner “*constraintprop*” sowie die beiden Dateien “*make.bat*” und “*run.bat*”. Sofern die ausführbaren Dateien von von Java “*java.exe*” und “*javac.exe*” im

Die Ausgabe mit den von uns vordefinierten Werten, welche wir oben berechnet und gemessen haben, sollte dann wie folgt aussehen:

Angle Interval = <0.0 ; 2.7632052717574225>

Angle Interval = <2.2660705923939837 ; 2.7632052717574225>

Angle Interval = <2.2660705923939837 ; 2.3371490818564125>

[illegible]

8

Hierbei stellt sich nun die Frage, warum der Y-Wert so stark eingeschränkt wurde, der X-Wert hingegen noch sehr breitgefächert ist. Der Grund hierfür liegt in der schlechten Distanzmessung. Aufgrund der niedrigen Auflösung gleichen sich die berechneten Distanzen zur südwestlichen und nordöstlichen Flagge. Deshalb müssen wir mit einem großen Fehler von 10% auf die Distanz rechnen. Dabei ergibt sich für die Distanzen folgende Darstellung, wobei die schwarzen Kreise hier die “perfekte” kreisförmige Constraintbildung darstellt und die blauen Kästen die tatsächlich verwendete rechteckige Annäherung. Dabei erkennt man bereits warum die X- und Y-Werte so liegen wie sie nuneinmal liegen.

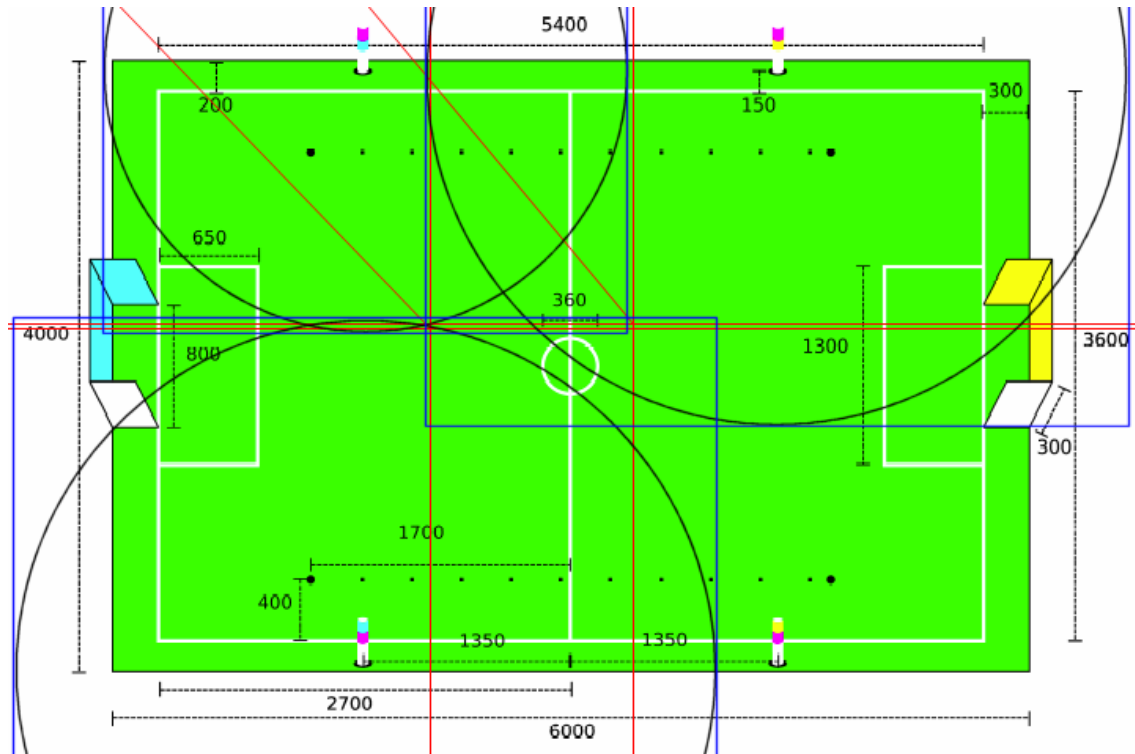


Abbildung 4: Veranschaulichung der Fehler

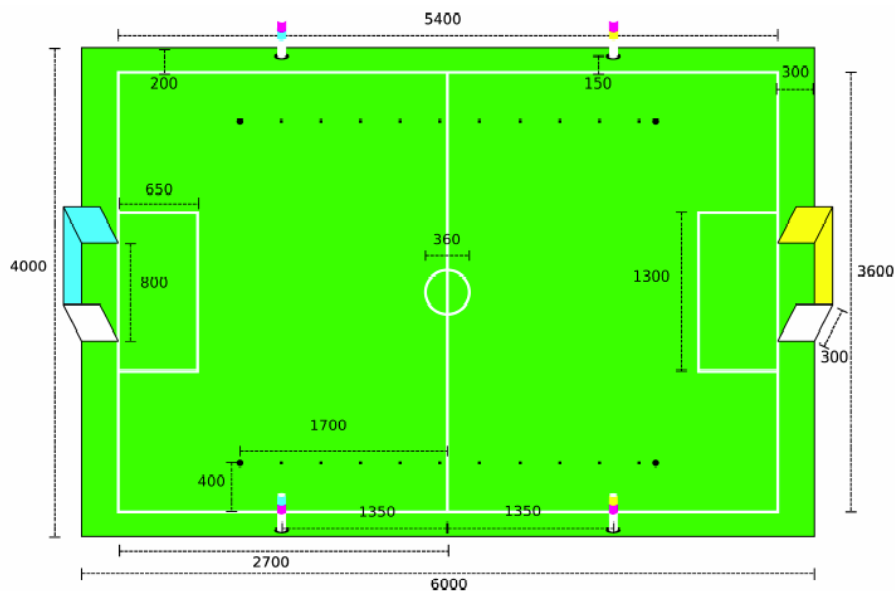


Abbildung 5: Übersicht des Spielfelds

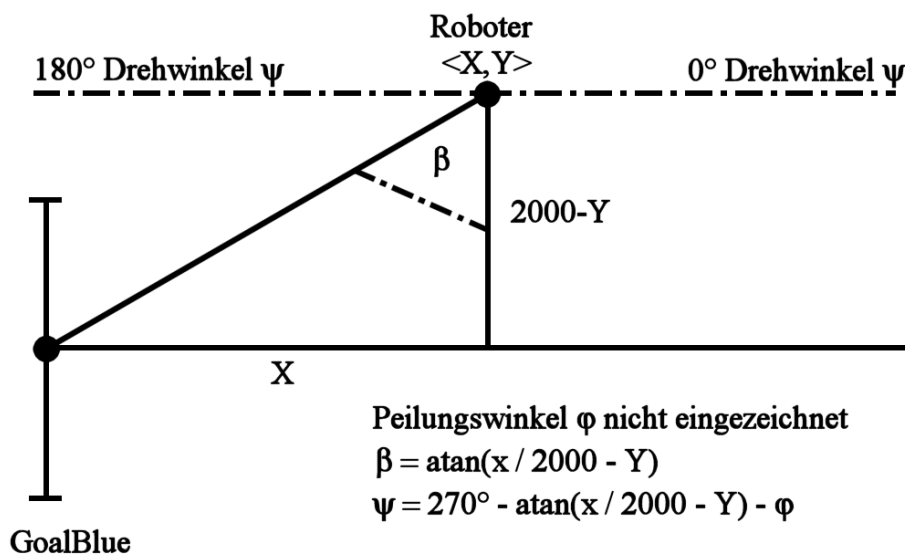


Abbildung 6: Drehwinkelberechnung - Blaues Tor