

## 第 1 章

### 1.1 はじめに

GPS(Global Positioning System : 全世界的測位システム)は、高度約 20,000km の円軌道を飛行する人工衛星を用いた 3 次元位置決定のためのシステムである。GPS はカーナビゲーションをはじめ、携帯電話、船舶航法システム、土木工事、移動体の自律制御など幅広い分野で使用され、その利用分野は今後も広がると考えられる。

2000 年 5 月 2 日 13 時 (JST) まで SA (Selective Availability) と呼ばれる測位精度を劣化させる操作がなされていた。その精度は、単独測位では水平方向で約 100m であった。測位精度を上げるために種々の補正システムが世界中で運用されている。現在 GPS 測位の測位精度は SA が解除され、受信機、アンテナ等の周辺機器の進歩により大幅に改善されている。

また、GPS と類似した航法システムでロシア連邦が運用している GLONASS (Global Navigation Satellite System) や欧州が開発している GALILEO、日本が開発、運用を計画している準天頂衛星などを GPS を組み合わせられることにより、測位精度のさらなる向上が期待される。

## 1.2 研究の目的と背景

近年、災害救助や監視、探査のために自律型ロボットの製作が活発化してきている。また、GPS は位置情報を容易に取得できることから、GPS 測位を用いた移動体の自律制御技術の応用として、構内自動監視システム用小型自律無人車や農業で用いるコンバインの自動制御など様々な開発が行われている。それに伴い自律型ロボットのコンテストの数も増えてきている。

そこで GPS 受信機より得られた位置情報を使い、与えられた目的の位置 ( waypoint ) を自律的に通過するロボットカーの制御アルゴリズムの製作を行った。

この目的を達成するために、まずロボットカーの制御系が局座標で設計されていることから GPS 受信機により得られた座標を局座標に変換する。

次にロボットカーの進行方向と目的地の方位角がずれているとき、ロボットカーの軌道を修正するために比例制御を用いる。また、この比例制御の制御ゲインをシミュレータを使用して調整する。

最後に GPS 受信機により得られた位置情報には大気遅延が含まれているため、数メートルの誤差がでてしまうのでこれを補正する。これらを用いて自律型ロボットカーの製作を行った。

## 第2章

### 2.1 座標変換

ロボットカーのナビゲーションには局地(East North Up)座標系で設計されている。しかし、GPS 受信機で取得された位置情報は楕円体座標で出力される。そこで楕円体座標を局地座標に変換することを考える。

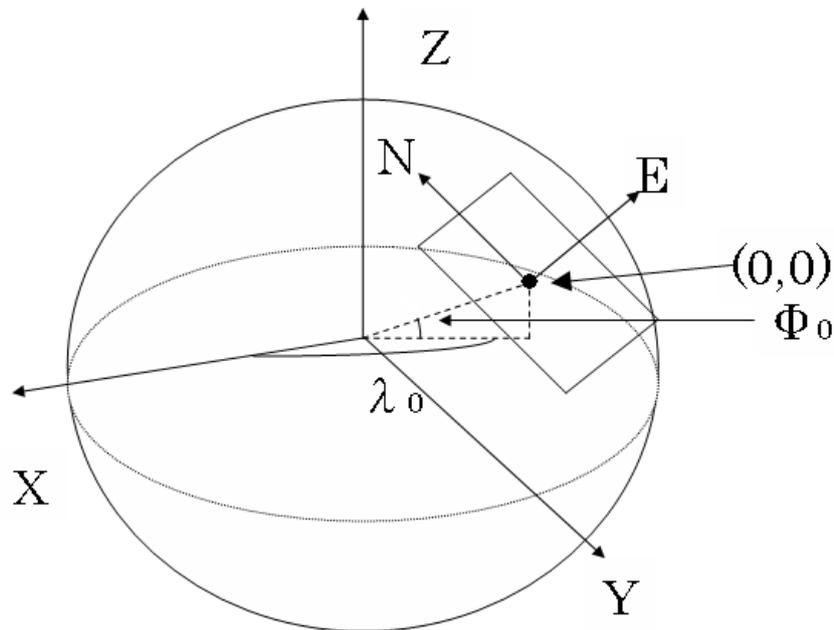


図 2-1 直交座標系と局地座標系

はじめに楕円体座標(  $\phi_0$ ,  $\lambda_0$  ) を直交座標( $x, y, z$ )に変換する方法について述べる。楕円体の中心は直交座標系の原点と、自転軸はZ軸と一致しているものとする(図 2-2)。

距離  $N$  を点  $P$  から子午線楕円に降ろした垂線の点  $P'$  と  $Z$  軸との間の長さとして定義する。楕円体座標が緯度  $\phi$ 、経度  $\lambda$ 、楕円体高  $h$  で表されるとき、点  $P$  の直交座標( $x, y, z$ )は、

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (N + h) \cos \phi \cos \lambda \\ (N + h) \cos \phi \sin \lambda \\ (N(1 - e^2) + h) \sin \phi \end{bmatrix} \quad (2.1)$$

ここで、 $N$  は卯酉線曲率半径、 $e$  は離心率である。 $a$  を赤道半径、 $b$  を極半径、 $f$  を扁平率とすると、

$$f = (a - b) / a \quad (2.2)$$

$$e^2 = f(2-f) \quad (2.3)$$

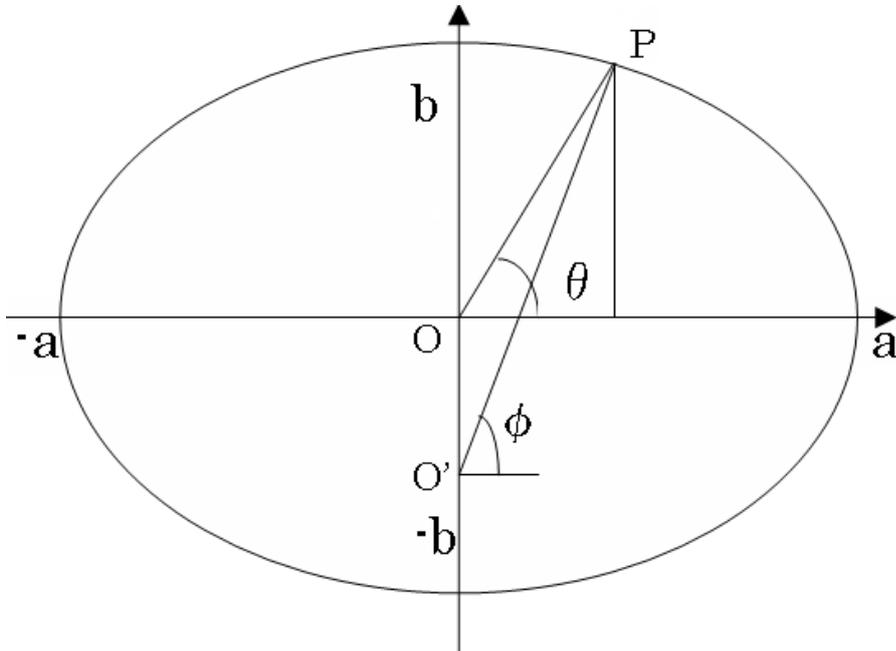


図 2-2

次に卵酉線曲率半径  $N$  を求める。まず、法線の式を求める。楕円の関数  $f(x, y)$  は

$$f(x, y) = x^2/a^2 + y^2/b^2 - 1 \quad (2.4)$$

となる。これを  $x, y$  でそれぞれ偏微分すると、

$$\partial f(x, y)/\partial x = 2x/a^2 \quad (2.4.a)$$

$$\partial f(x, y)/\partial y = 2y/b^2 \quad (2.4.b)$$

点  $P$  の座標は  $(a \cos \theta, b \sin \theta)$  なのでこれを代入し、法線の式をパラメータ  $t$  で表すと

$$x = 2t \cos \theta/a + a \cos \theta \quad (2.5.a)$$

$$y = 2 \sin \theta/b + b \sin \theta \quad (2.5.b)$$

ここで交点  $G$  の座標を求める。 $G$  の  $x$  座標は  $0$  なので(2.5.a)式の  $x$  に  $0$  を代入して

$$t = -a^2/2 \quad (2.6)$$

求めた  $t$  を(2.5.b)に代入すると、

$$y = (b^2 - a^2) \sin \theta/b \quad (2.7)$$

となり、 $y$  座標が求まる。よって点  $P$  の座標は  $(a \cos \theta, b \sin \theta - (b^2 - a^2) \sin \theta / b)$  となり、三平方の定理により  $N$  の長さを求めることができる。

$$N^2 = (a \cos \theta)^2 + (b \sin \theta - (b^2 - a^2) \sin \theta / b)^2 \quad (2.8)$$

(2.8) 式において、地球中心角  $\theta$  で表されているので、 $\theta$  を  $\phi$  に変換する。

$\tan \phi = a \tan \theta / b$  であることから  $N$  は

$$N = a / (1 - e^2 \sin^2 \phi)^{\frac{1}{2}} \quad (2.9)$$

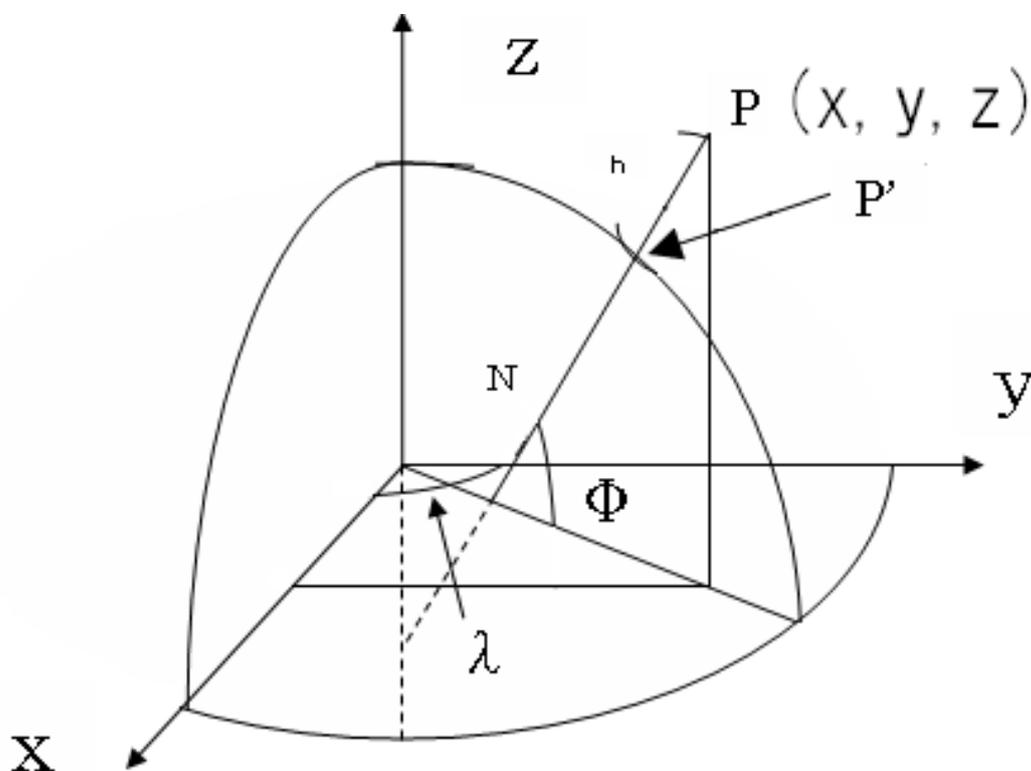


図 2-2 直交座標系

求めた直交座標  $(x, y)$  を局座標 (ENU 座標) に変換する。図 2.1 より、まず  $z$  軸を角度  $(\lambda + 90^\circ)$  だけ回転させ、 $x$  軸を角度  $(90^\circ - \phi)$  だけ回転させる変換行列  $R$  を考えると次式により求めることができる。

$$\mathbf{R} = \begin{bmatrix} -\sin \lambda_0 & \cos \lambda_0 & 0 \\ -\sin \phi_0 \cos \lambda_0 & -\sin \phi_0 \sin \lambda_0 & \cos \phi_0 \\ \cos \phi_0 \cos \lambda_0 & \cos \phi_0 \sin \lambda_0 & \sin \phi_0 \end{bmatrix} \quad (2.10)$$

上式の変換する回転行列  $R$  を直交座標  $(x, y, z)$  にかけることによって局地座標系で表すことができる。

## 2.2 局地座標変換の簡略化

このように、厳密な局地座標変換では、三角関数や浮動小数点演算が多くなり、8ビットマイコンの処理能力では演算負荷が大きすぎる。

そこで、ロボットカーの移動範囲が十分に狭いものと仮定して三角関数を近似して、局地座標変換を行う。

### 2.2.1 緯度方向の変位

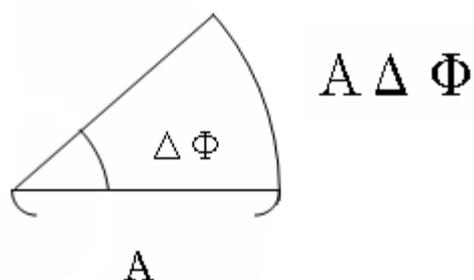


図 2.2-1 緯度方向の変位

まず、緯度方向の変位から求める。地球を完全な球体と仮定し、下図 2.2-2 より A を赤道半径、 $\Delta\phi$  を緯度緯度差とすると、弧度法により緯度方向の変位  $\Delta N$  は

$$\Delta N = A \Delta \phi \quad (2.2.1)$$

ここで  $\Delta\phi$  はきわめて小さい値であることから  $\Delta N$  は直線とみなすことができる。

### 2.2.2 経度方向の変位

次に経度方向の変位を求める。(2.2.1)式を導いたとき同様、地球を完全な球体と仮定して経線の間隔が極めて近いことから平行と仮定して弧度法を用いる。しかし緯度によって円の半径が変化するので図 2.2-2 のように基準点の緯度  $\phi_0$  を用いて  $z$  軸から  $\lambda_0$  までの長さを求め、それに経度差  $\Delta\lambda$  を掛ける。よって経度方向の変位  $\Delta E$  は

$$\Delta E = A \times \Delta\lambda \times \cos \phi_0 \quad (2.2.2)$$

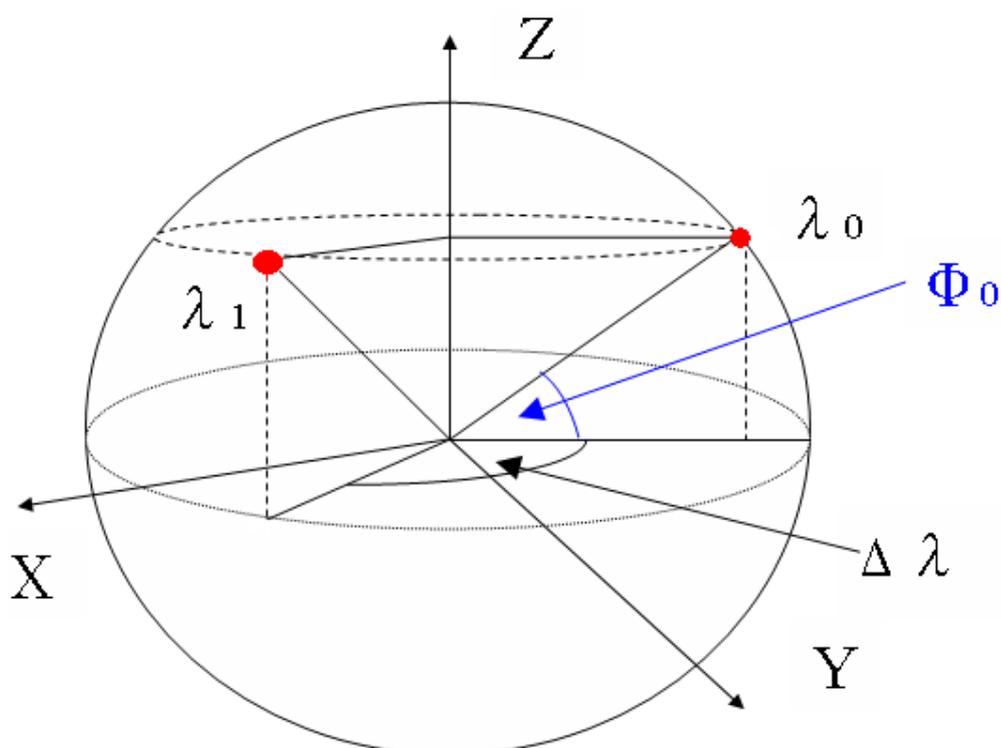


図 2.2-2 経度方向の変位

2.1 で厳密に求めた局地座標変換と 2.2 で使用した簡略化した座標変換を東京海洋大学、越中島キャンパスにあるテニスコートの座標を用いて比較してみる。

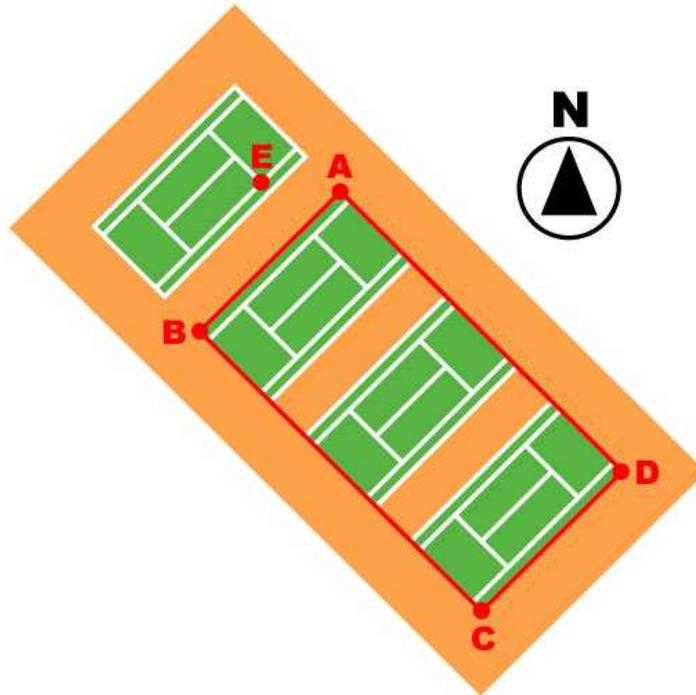


図 2.2-3

上図の A,B,C,D,E の緯度、経度座標は下表 2.2-1 のようになる。E を基準点としたときの厳密な座標変換を表 2.2-2 に示す。また、簡略化した座標変換を表 2.2-3 に示す。さらに厳密な局地座標と簡略化した局地座標の差を表 2.2-4 に示す。座標変換の誤差が一番大きいのは C の North 方向で約 16cm であった。ここで、waypoint から 2m の範囲に入ればいいことからロボットカーのナビゲーションには十分な精度が得られることが確認できた。

	緯度	経度
A	35.66501	139.7934
B	35.66486	139.7932
C	35.66457	139.7936
D	35.66472	139.7937
E	35.66502	139.7933

表 2.2-1 テニスコートの座標

	East(m)	North(m)
A	9.054401	-1.4313
B	-7.95883	-17.9856
C	23.15223	-49.8957
D	40.11114	-33.3747

表 2.2-2 厳密な局地座標

	East(m)	North(m)
A	9.044036	-1.43602
B	-7.94971	-18.0449
C	23.1256	-50.0604
D	40.06508	-33.4849

表 2.2-3 簡略化した局地座標

	East(m)	North(m)
A	0.010365	0.004721
B	-0.00912	0.059289
C	0.026629	0.164675
D	0.046059	0.110203

表 2.2-4 座標変換誤差

## 第3章 ロボットカーの制御

### 3.1 フィードバック制御による軌道修正

ロボットカーの制御では、GPS 受信機から得られる位置、速度、進行方向の情報を用いる。これらの情報はNMEAフォーマットで出力される。

NMEAとはNational Marine Electronics Associationの略でデータ転送の標準フォーマットである。すべてアスキーフォーマットで出力され、文字、カンマ、数字から成る。このNMEAフォーマットのなかのGPGGAで位置情報を、GPVTGで速度、進行方向の情報を得る。表3.1-1はNMEAフォーマットのGPGGA、GPVTGの出力例ある。

\$GPGGA,031713.00, <u>3539.90320</u> ,N, <u>13947.60261</u> ,E,1,09,0.95,4.9,M,39.4,M,,*52
( 1 )                      ( 2 )
\$GPVTG, <u>93.42</u> , T, ,M, <u>0.022</u> , N, <u>0.040</u> , K,A*05
( 3 )                      ( 4 )                      ( 5 )
( 1 ): 北緯 35 度 39.90320 分
( 2 ): 東経 139 度 47.60261 分
( 3 ): 真北からの進行方向 93.42 度
( 4 ): 速度(ノット表示)
( 5 ): 速度 ( km/h 表示 )

表 3.1-1 NMEA フォーマット例

これらの情報を用いて、軌道修正を行う。図 3.1-1 はロボットカーの制御モデルを表している。 $v$  はロボットカーの速度、 $\theta$  は目的地の方位角、 $\phi$  は操舵角である。

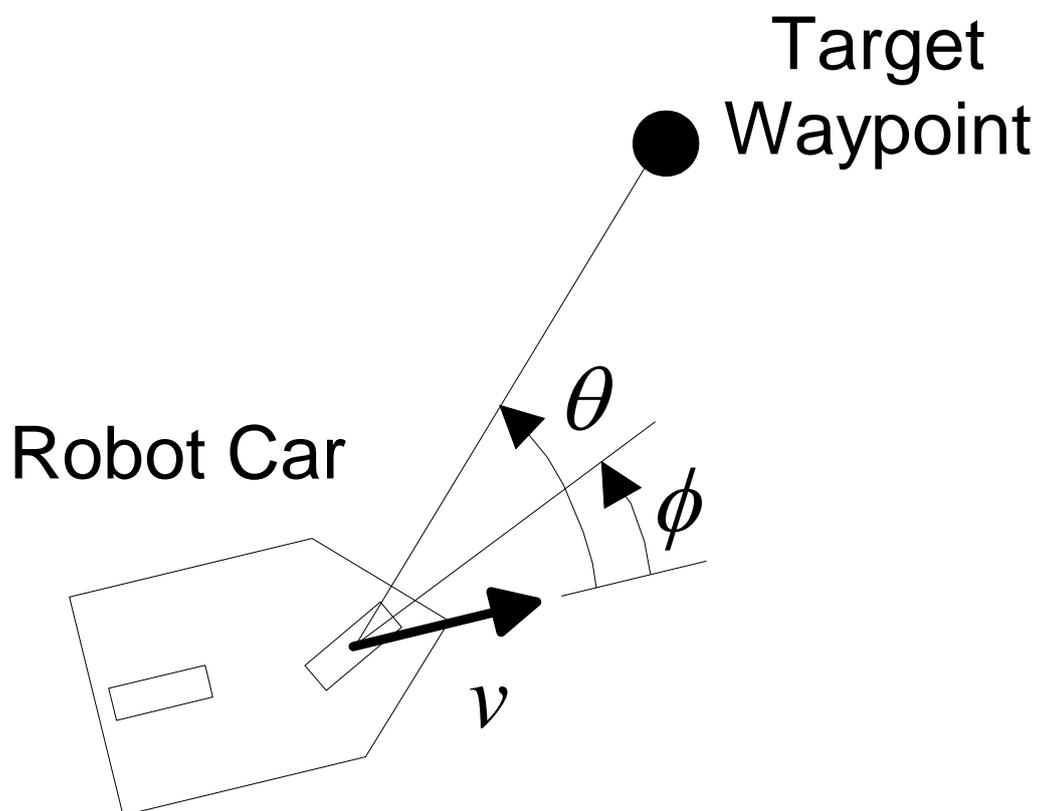


図 3.1-1

ロボットカーの進行方位角が目的地の方位角とずれているとき、目的地に向かわせるために、ロボットカーの起動を修正する必要があります。

もし、目的地の方位角と同じ角度で舵を切るとドリフトしたりするので、急に目的地の方位角にしようとしなくてゆっくりと方位角に近づけるようにする。つまり、操舵角を目的地の方位角に比例した大きさになるようにして、徐々に調整する制御方法、比例制御を行う。

操舵角  $\phi$  は 3.1-1 式により求めることができる。ここで  $K$  は制御ゲイン、 $\theta$  は目的地の方位角である。

$$\phi = K \times \theta \quad (3.1-1)$$

### 3.2 シミュレーション

この制御ゲイン $K$ を調整するために、野外での実験により求めようとする、毎回、制御ゲインを変えて走行を行わなければならないと時間がかかる。また、ロボットカーに対する操舵角の影響が未知なため、衝突や転倒の危険があり故障する可能性がある。そこで、ソフトウェアシミュレータを製作し $K$ を調整した。

図 3.1-2 にシミュレーションに使用したロボットカーの数式モデルを示す。モデルを単純化するために、速度一定と仮定し後輪で駆動し、前輪で操舵行う二輪モデルを用いる。

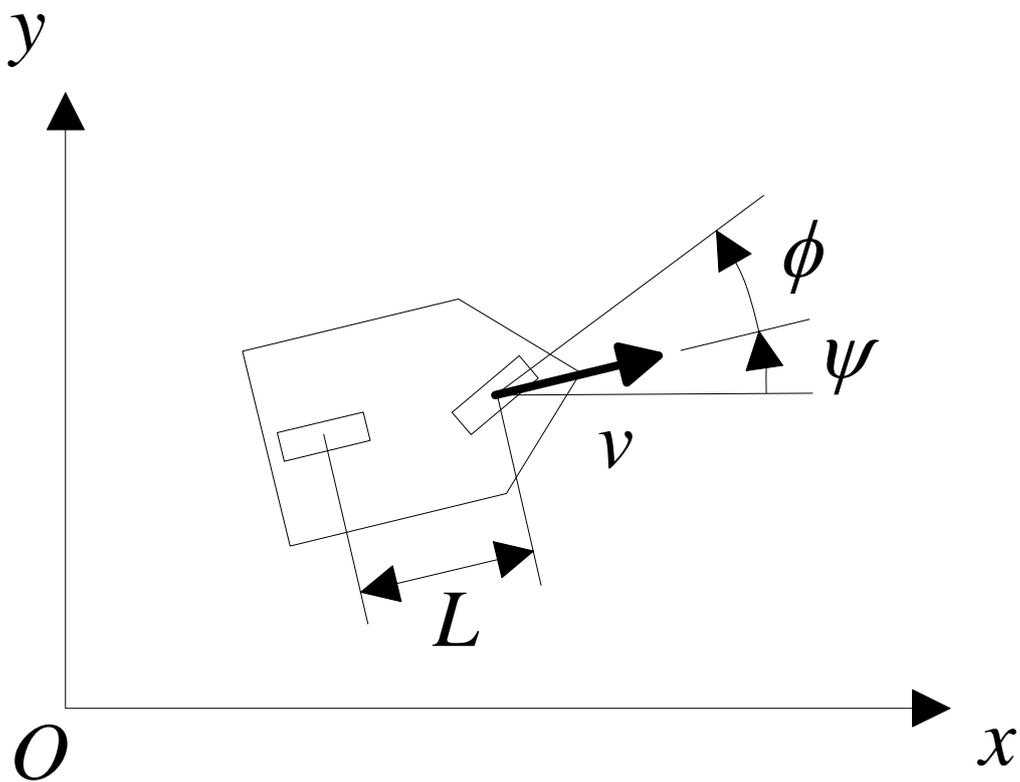


図 3.1-2 ロボットカーの運動モデル

ここで  $v$  は速度、 $\psi$  は  $x$  に対するロボットカーの姿勢、 $\phi$  は操舵角、 $L$  はトレット（軸距）である。また、このロボットカーの運動方程式は 3.2-1 式で表すことができる。

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v \cos \psi \\ v \sin \psi \\ v \tan \phi / L \end{bmatrix} \quad (3.2-1)$$

3.2-1 式の運動方程式に 3.1-1 の制御則を代入し、数値積分することで、フィードバック制御されたロボットカーの軌道制御をシミュレーションした。  
 下図にシミュレーション結果を示す。赤い三角が waypoint を表し、青い線がロボットカーの軌跡、赤い矢印が進行方向を示している。

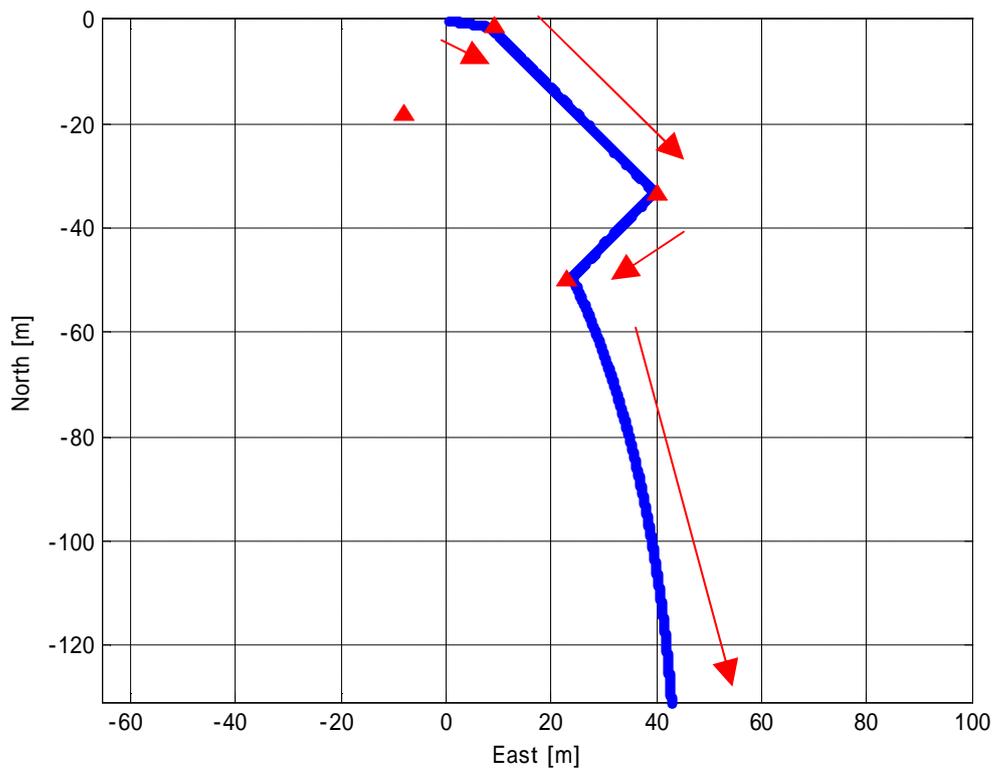


図 3.2-1 制御ゲイン  $K=-0.9$  のとき

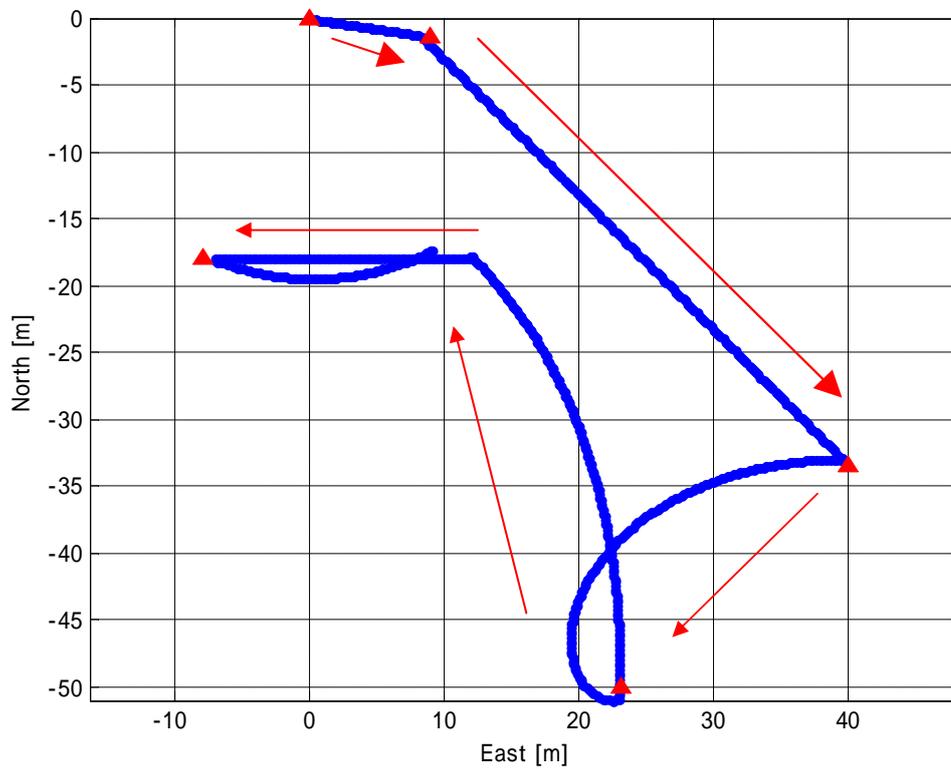


図 3.2-2 制御ゲイン  $K=-0.8$  のとき

図 3.2-1、図 3.2-2 のように制御ゲインが大きいと急ハンドルとなり、ロボットカーが不要な旋回運動をしてしまう。

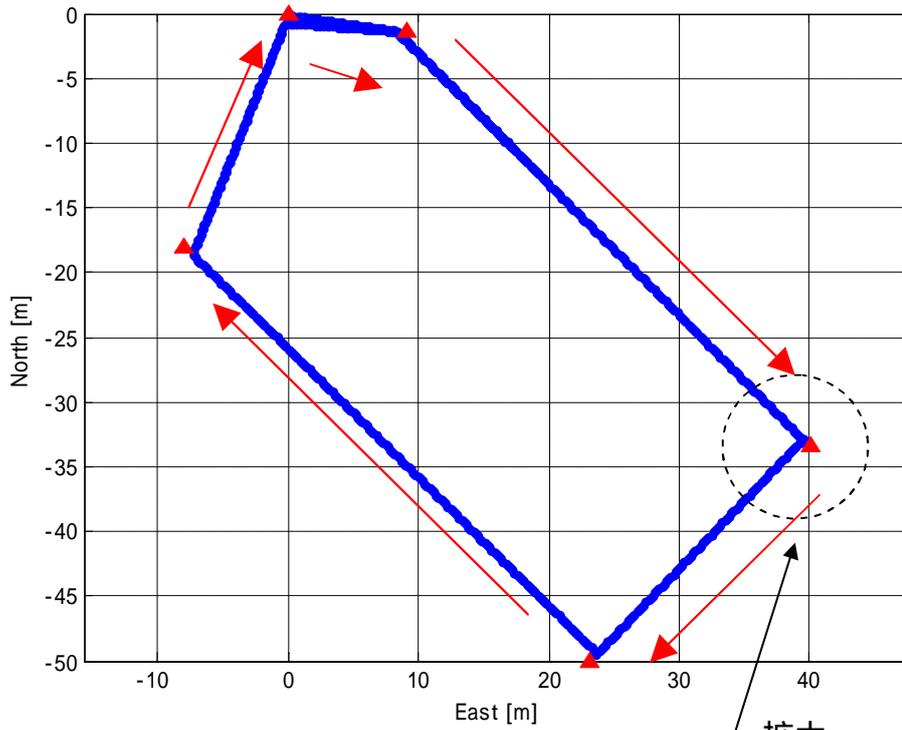


図 3.2-3 制御ゲイン  $K=-0.7$  のとき

拡大

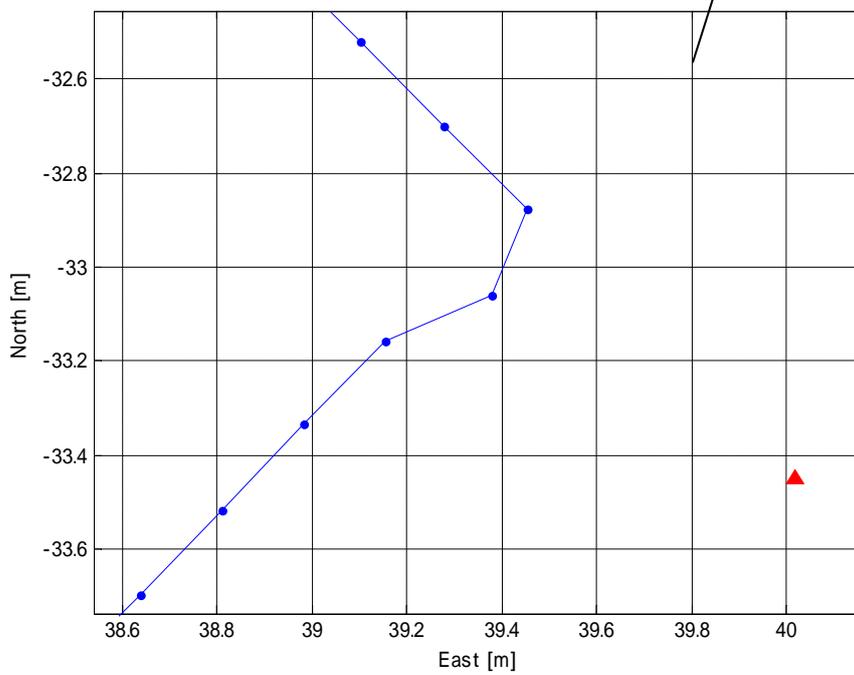


図 3.2-4 図 3.2-3 を拡大したもの

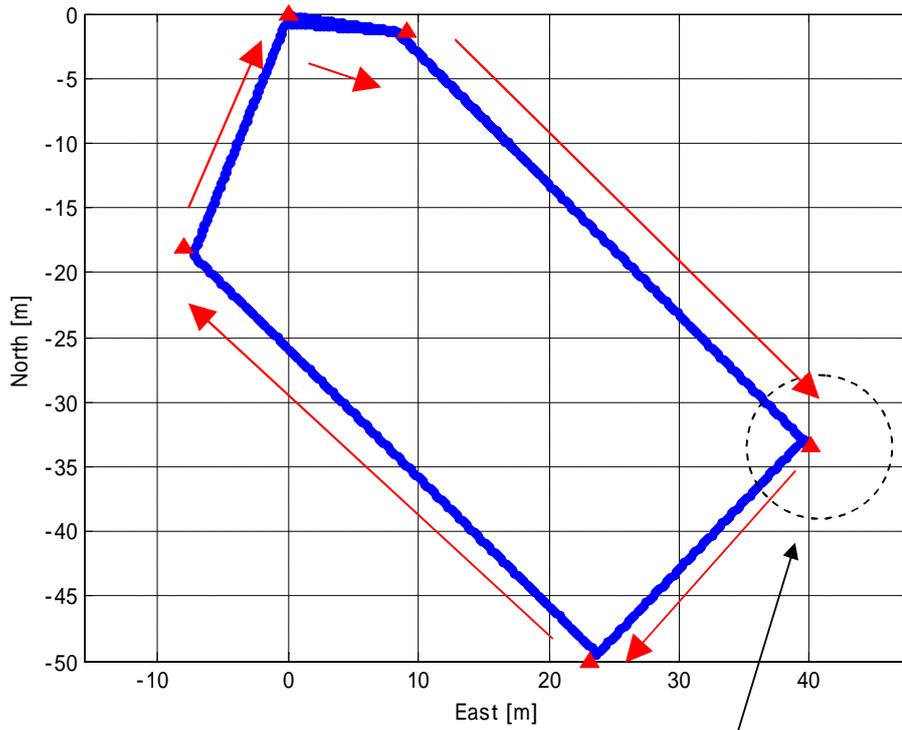


図 3.2-5 制御ゲイン  $K=-0.6$  のとき

拡大

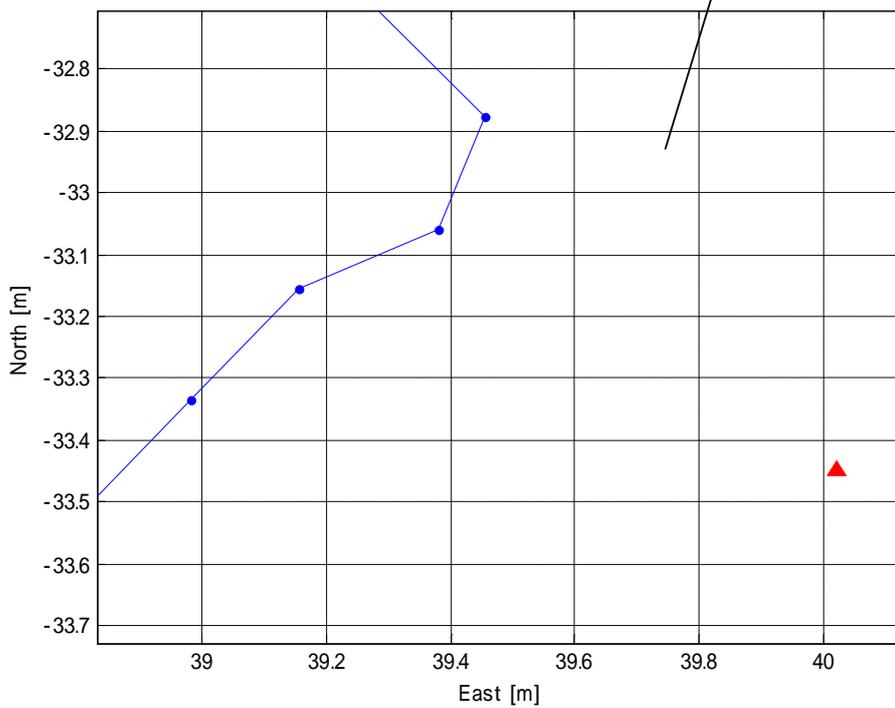
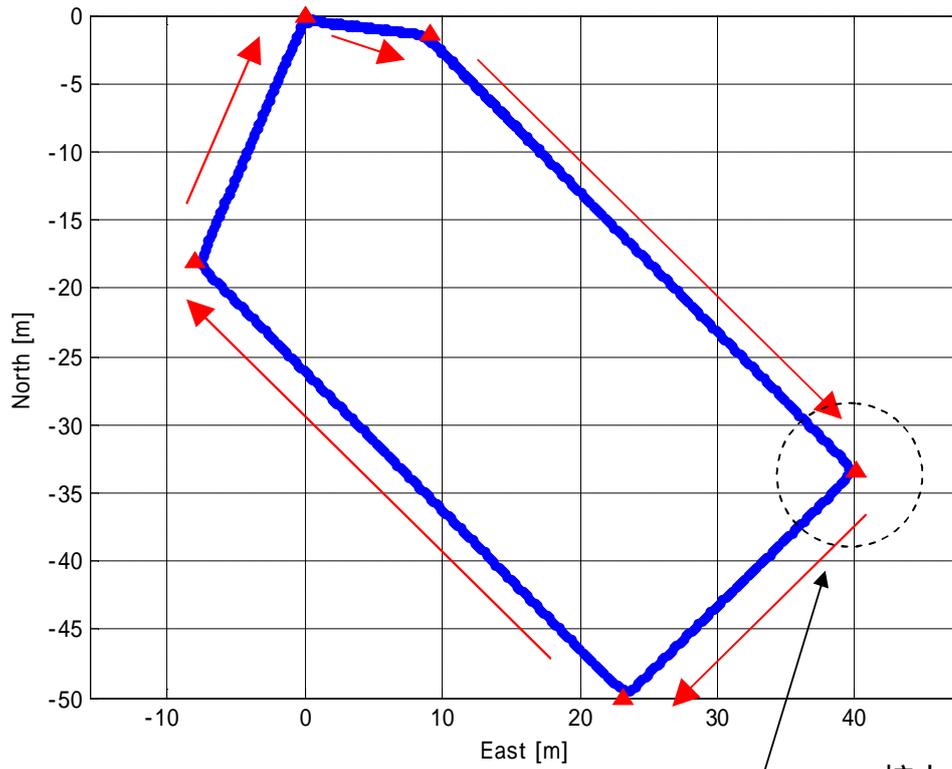


図 3.2-6 図 3.2-5 を拡大したもの

制御ゲイン  $K=0.7$ 、 $0.6$  にしたとき、図 3.2-3、図 3.2-4 のようにうまく制御されているように見えるがそれぞれ 2 つめの矢印の先の部分を拡大してみると直角に近い角度で曲がっているため実際に走らせたとき危険である。制御ゲインを  $0.5$ 、 $0.4$  としたときも同様の理由である。



拡大

図 3.2-6 制御ゲイン  $K=-0.3$  のとき

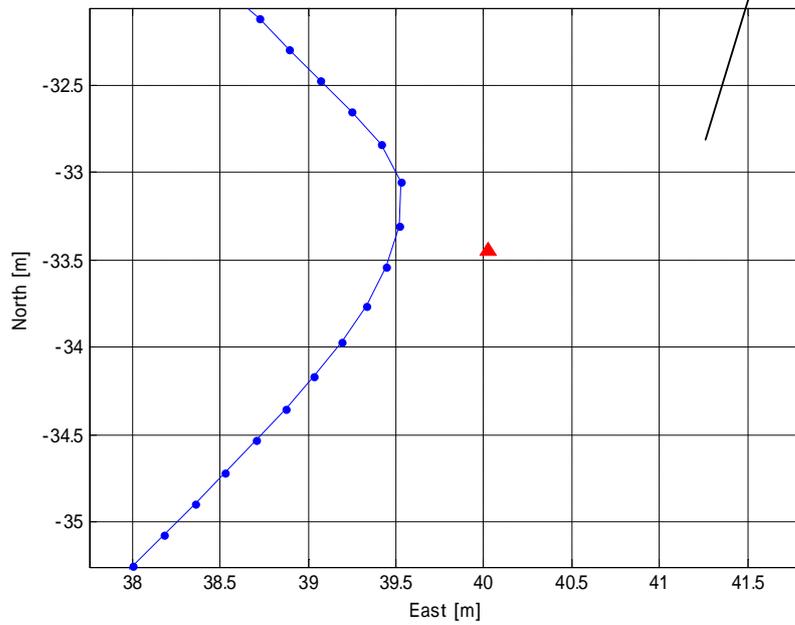


図 3.2-7 図 3.2-6 を拡大したもの

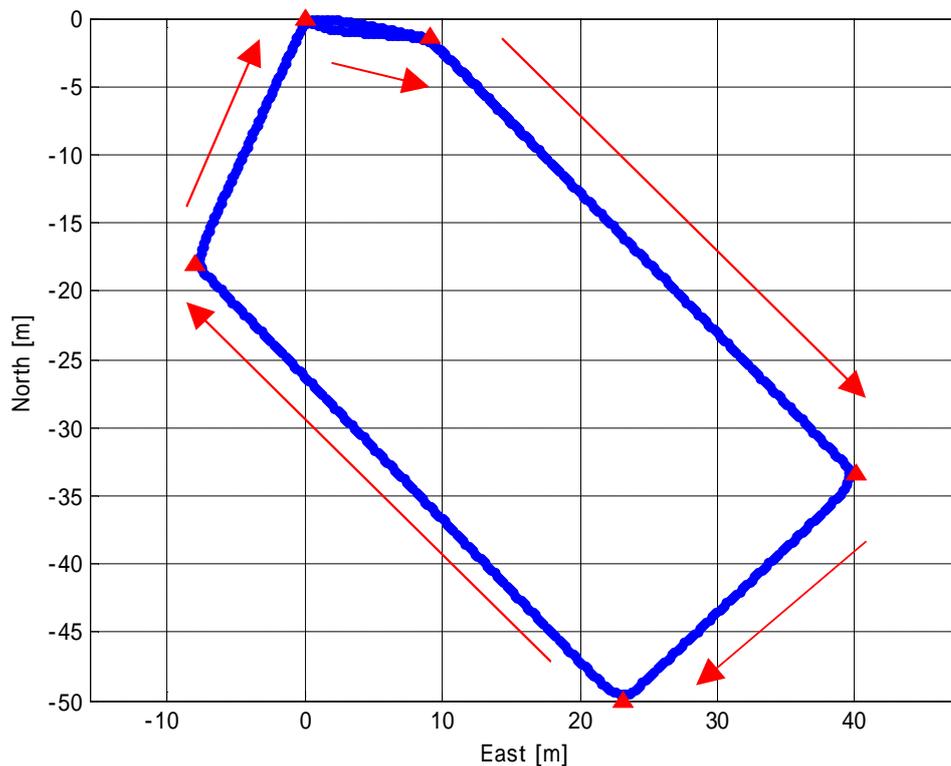


図 3.2-8 制御ゲイン  $K=-0.2$  のとき

制御ゲイン  $K = -0.3$  としたとき、図 3.2-6、図 3.2-7 より滑らかにそうこうしているのがわかる。また制御ゲイン  $K = 0.2$  としたとき、 $K = 0.3$  のときと同様に滑らかな走行をしているように見えるが目的地の方位角にゆっくりと近づいていくため、時間がかかってしまう。座標(0,0)からスタートして、ゴール(0,0)まで戻ってくる時間は  $K = 0.3$  のとき、138.75sec、 $K = 0.2$  のとき、139.75sec である。以上のことから制御ゲイン  $K$  を 0.3 とした。

## 第 4 章 GPS 受信機の測位誤差の補正

### 4.1 ユーザー位置計算

衛星  $i$  と受信機との間の距離の測定値を  $r_i$  [m] と書くことにする。これは正確な距離  $\rho_i$  [m] に受信機の時計の進み  $\delta$  [s] による誤差が加わって測定されたものであることから擬似距離 (pseudorange) と呼ばれ、 $r_i = \rho_i + c\delta = \rho_i + s$  の関係にある。受信機の時計が進んでいると、測距信号が到着する時間に本来よりも進んだ時刻が観測されるため、見かけ上余分に時間が掛かったことになり、距離の測定結果が長めとなる。

衛星の位置は既知として、測定した擬似距離から受信機位置を求めることを考える。衛星と受信機との正確な距離  $\rho_i$  は、受信機位置を  $(x, y, z)$ 、衛星  $i$  の位置を  $(x_i, y_i, z_i)$  とすると、

$$\rho_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \quad (4.1-1)$$

と表される (単位はすべて [m])。可視衛星数が  $N$  のとき、擬似距離と本来の距離との関係より、受信機位置  $X$  を求めるためには、連立方程式

$$\begin{aligned} r_1 &= \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} + s \\ r_2 &= \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} + s \\ &\vdots \\ r_N &= \sqrt{(x_N - x)^2 + (y_N - y)^2 + (z_N - z)^2} + s \end{aligned} \quad (4.1-2)$$

を、 $x, y, z, s$  についてとけばよい。3次元の位置を決めるためには未知数は最低3つとなるが、式 4.1-1 では受信機の時計誤差  $\delta$  も未知数として扱われているため未知数は合計4つとなり、これを解くには最低4つの方程式が必要となる。したがって、4個以上の衛星を使用して擬似距離を測定する必要がある。

連立方程式 4.1-2 のそれぞれの式は、二乗すると球面を表す式になる。幾何学的には、受信機位置  $x, y, z$  はこれらの球面の交点として求めることができる。正しく距離が測定されていれば4つの球面は1点で交わるはずだが、擬似距離は受信クロック誤差  $s = c\delta$  だけ長く測定されているから、1点では交わらない。このため、図 4.1-1 のように球面が1点で交わるように  $s$  を変化させて球の大きさを調節すると、その時の交点として受信機位置が求められる。5個以上の衛星を使用する場合はそれでも1点では交わらないので、受信機位置  $x, y, z$  と各球面との距離の二乗和が最も小さくなるように  $s$  を調整する (最小二乗法)。

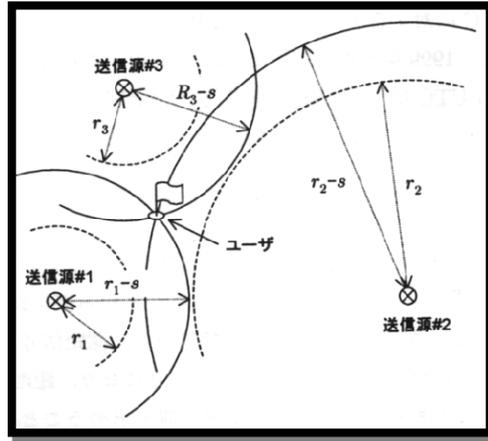


図 4.1-1 球面の交点として位置を求める

式 4.1-2 は非線形の連立方程式であるため、通常は適当な初期値のまわりで線形化をおこない、逐次近似法により解を得る。いかにその手順を示す。

- ( 1 )  $x, y, z, s$  について適当な初期値  $x^0, y^0, z^0, s^0$  を用意する。
- ( 2 )  $x^0, y^0, z^0, s^0$  としたときに擬似距離として測定されるべき値を計算する。

$$\begin{aligned}
 r_1^0 &= \sqrt{(x_1 - x^0)^2 + (y_1 - y^0)^2 + (z_1 - z^0)^2} + s^0 \\
 r_2^0 &= \sqrt{(x_2 - x^0)^2 + (y_2 - y^0)^2 + (z_2 - z^0)^2} + s^0 \\
 &\vdots \\
 r_N^0 &= \sqrt{(x_N - x^0)^2 + (y_N - y^0)^2 + (z_N - z^0)^2} + s^0
 \end{aligned}
 \tag{4.1-3}$$

- ( 3 ) 実際に測定された擬似距離  $r_i$  に対して、残差  $\Delta r_i = r_i - r_i^0$  を求める。
- ( 4 )  $x^0, y^0, z^0, s^0$  をこの残差に相当する分だけ修正すれば、正しい解に近づくことができる。このためには、 $r_i$  の  $x, y, z, s$  による偏微分

$$\begin{aligned}
 \frac{\partial r_i}{\partial x} &= -(x_i - x^0)/r_i^0, & \frac{\partial r_i}{\partial y} &= -(y_i - y^0)/r_i^0, \\
 \frac{\partial r_i}{\partial z} &= -(z_i - z^0)/r_i^0, & \frac{\partial r_i}{\partial s} &= 1
 \end{aligned}
 \tag{4.1-4}$$

を用いる。  $x^0, y^0, z^0, s^0$  の変化量を  $\Delta x, \Delta y, \Delta z, \Delta s$  と書くと、

$$\begin{aligned}
\Delta r_1 &= \frac{\partial r_1}{\partial x} \Delta x + \frac{\partial r_1}{\partial y} \Delta y + \frac{\partial r_1}{\partial z} \Delta z + \frac{\partial r_1}{\partial s} \Delta s \\
\Delta r_2 &= \frac{\partial r_2}{\partial x} \Delta x + \frac{\partial r_2}{\partial y} \Delta y + \frac{\partial r_2}{\partial z} \Delta z + \frac{\partial r_2}{\partial s} \Delta s \\
&\vdots \\
\Delta r_N &= \frac{\partial r_N}{\partial x} \Delta x + \frac{\partial r_N}{\partial y} \Delta y + \frac{\partial r_N}{\partial z} \Delta z + \frac{\partial r_N}{\partial s} \Delta s
\end{aligned} \tag{4.1-5}$$

という連立方程式が得られる。 $\Delta r_i$  は分かっているから、この連立方程式を  $\Delta x, \Delta y, \Delta z, \Delta s$  について解けばよい。

(5) 得られた  $\Delta x, \Delta y, \Delta z, \Delta s$  を初期値として与えた  $x^0, y^0, z^0, s^0$  に加えて、初期値を変更する。

$$\begin{aligned}
x^1 &= x^0 + \Delta x, & y^1 &= y^0 + \Delta y, \\
z^1 &= z^0 + \Delta z, & s^1 &= s^0 + \Delta s,
\end{aligned} \tag{4.1-6}$$

(6) 初期値を  $x^1, y^1, z^1, s^1$  に更新して、(2) に戻る。以上の手順を  $\Delta x, \Delta y, \Delta z, \Delta s$  が十分に小さくなるまで繰り返す。

このような手順で解を求めることが出来る。通常はそれほど厳密な初期値は必要なく、数回程度の繰り返しで正しい解に収束する。

手順(4)の方程式の解法を紹介する。取り扱いを簡単にするために方程式を行列により表現する。ベクトル  $\Delta X = [\Delta x \ \Delta y \ \Delta z \ \Delta s]^T$ 、 $\Delta R = [\Delta r_1 \ \Delta r_2 \ \cdots \ \Delta r_N]^T$  を利用すると、(4)の方程式は、

$$G \Delta X = \Delta R \tag{4.1-7}$$

と簡単に書くことが出来る。行列  $G$  は、

$$G = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial z} & \frac{\partial r_1}{\partial s} \\ \frac{\partial r_2}{\partial x} & \frac{\partial r_2}{\partial y} & \frac{\partial r_2}{\partial z} & \frac{\partial r_2}{\partial s} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial r_N}{\partial x} & \frac{\partial r_N}{\partial y} & \frac{\partial r_N}{\partial z} & \frac{\partial r_N}{\partial s} \end{bmatrix} = \begin{bmatrix} \frac{-(x_1 - x)}{r_1} & \frac{-(y_1 - y)}{r_1} & \frac{-(z_1 - z)}{r_1} & 1 \\ \frac{-(x_2 - x)}{r_2} & \frac{-(y_2 - y)}{r_2} & \frac{-(z_2 - z)}{r_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{-(x_N - x)}{r_N} & \frac{-(y_N - y)}{r_N} & \frac{-(z_N - z)}{r_N} & 1 \end{bmatrix} \tag{4.1-8}$$

で、GPS 衛星と受信機の相対的な位置関係により決まる。式 4.1-7 の連立方程式の解は、方程式が 4 つ ( $G$  が  $4 \times 4$  の正方行列の場合) であれば、 $G$  の逆行列を求めることで得られる。

$$\Delta X = G^{-1} \Delta R \quad (4.1-10)$$

方程式が 5 つ以上 (測位に使用する衛星が 5 つ以上) ある場合は未知数よりも方程式のほうが数が多いことになる。これは、「過剰決定の状態」と呼ばれ、実際式 4.1-10 の左辺には誤差  $\varepsilon_i$  が含まれており、

$$G \Delta X + \varepsilon = \Delta R \quad (4.1-11)$$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix} \quad (4.1-12)$$

となる。そこで、この誤差  $\varepsilon_i$  の二乗和を最小にするように解を決定する。これを「最小二乗法」と呼ぶ。まず式 4.1-11 を

$$\varepsilon = \Delta R - G \Delta X \quad (4.1-13)$$

と書き直す。このとき誤差  $\varepsilon$  の二乗和  $f$  は、

$$\begin{aligned} f &= \sum_{i=1}^N (\varepsilon_i)^2 \\ &= \varepsilon^T \varepsilon \\ &= (\Delta R - G \Delta X)^T (\Delta R - G \Delta X) \\ &= \Delta R^T \Delta R - 2 \Delta R^T G \Delta X + \Delta X^T (G^T G) \Delta X \end{aligned} \quad (4.1-14)$$

で表すことができる。ここで  $T$  は転置行列を示す。 $f$  を最小にするために変数  $\Delta X$  で偏微分して極値を求めると、

$$\begin{aligned} \frac{\partial f}{\partial \Delta X} &= -2 \Delta R^T G + 2 \Delta X^T (G^T G) = 0 \\ \Delta X^T (G^T G) &= \Delta R^T G \end{aligned} \quad (4.1-15)$$

となり、両辺の転置をとり、 $(AB)^T = B^T A^T$  という性質と  $(G^T G)$  が対称行列であることから、

$$(G^T G) \Delta X = G^T \Delta R \quad (4.1-16)$$

となる。これを「正規方程式」と呼び、 $\Delta X$ について解けば、

$$\Delta X = (G^T G)^{-1} G^T \Delta R \quad (4.1-17)$$

と最小2乗解が得られる。得られた $\Delta X$ を更新量とする。

## 4.2 誤差要因

GPS 受信機は、側距信号を用いて衛星と受信機との距離を測定し、測位計算を実行する。このとき衛星の位置情報に誤差があると、直接的に測位誤差となって現れることになる。

GPS 衛星の軌道は正確に予測されて航法メッセージとして放送されているが、現在の技術水準では数 m ~ 10m 程度の誤差は避けることができず、この予測誤差が衛星軌道誤差として残ることになる。

GPS の信号は衛星から受信機まで伝搬してくる間に通過する媒体による影響を受けている。伝搬距離は受信機真上約 20,000km(図 4.1.1 衛星 1)から水平線上約 26,000km(図 4.1.1 衛星 2)に及ぶ。

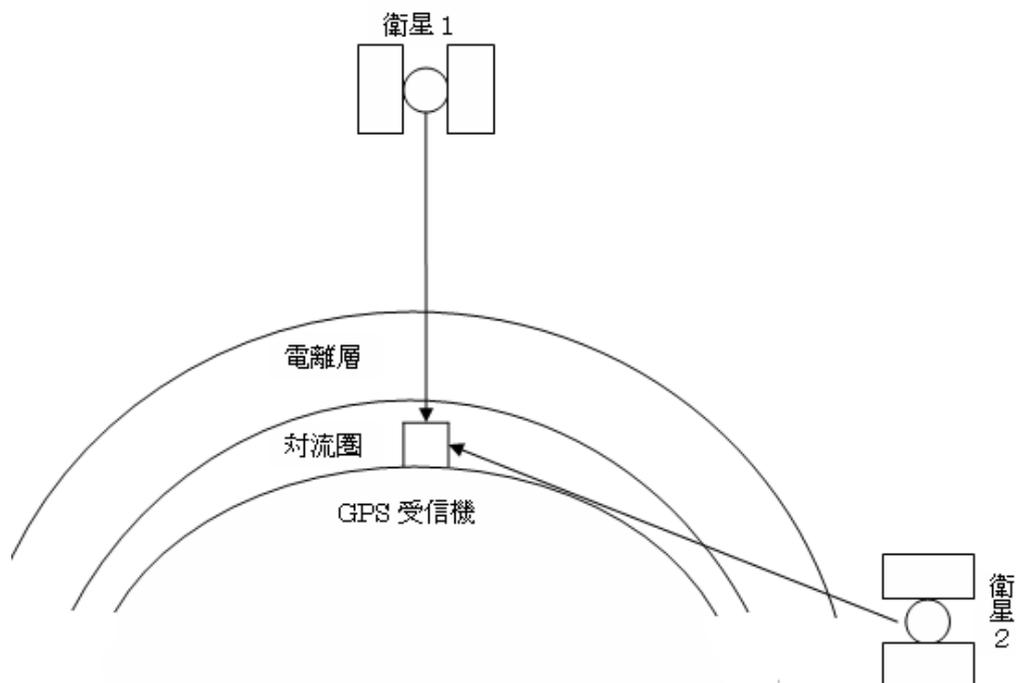


図 4.1.1 衛星位置と伝搬距離

GPS 衛星から送信された信号は受信機に到達する過程で電離層を通過する。電離層とは自由電子やイオンの密度の大きい層のこといい、分子や原子の状態にある空気中の酸素や窒素が紫外線や X 線により電離される。その後、対流圏を通過し受信される。対流圏とは対流が起こりやすい場所で、曇や雨をふらせるなどの天気に影響を及ぼしている。ここで信号の伝搬速度は光速 ( $c = 299,792,458 \text{ m/s}$ ) である。そのため、これらの大気を通過することにより伝搬速度が遅れて受信機に伝わる。

受信機と衛星の位置の関係によって伝搬距離が変化する。もしも図 4.1.1 の衛星 2 (水平線上の衛星) を受信することができた場合、衛星 1 (頭上の衛星) よりも大気の通過距離が長くなる。また、仰角の低い衛星からの信号を受信すると図 4.1.2 のように直接受信される信号と壁などに反射されて受信する信号が表れる。このような二つ以上の経路を通過して受信機に伝わる現象をマルチパスという。

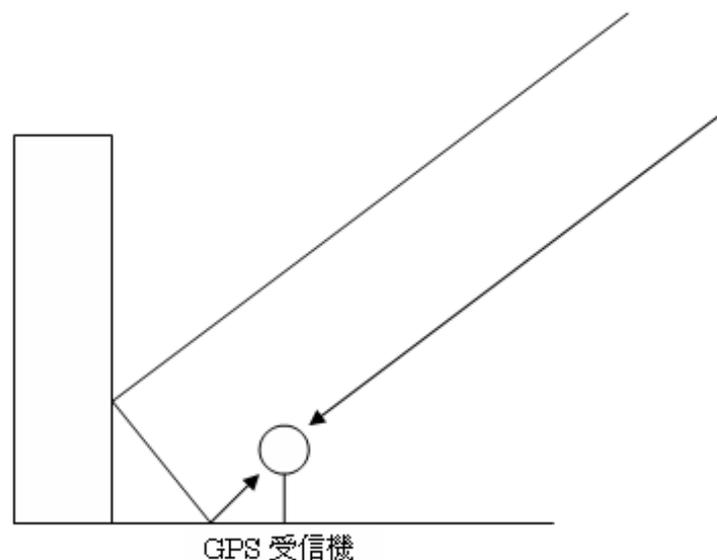


図 4.1.2 マルチパスのイメージ

仰角の低い衛星を受信すると大気による伝搬時間の遅れやマルチパスの影響により測位誤差が生じてしまう。

### 4.3 測位誤差の補正

#### 4.3.1 MSAS による補正

2007年9月にMSASの正式な運用が開始されたので、MSASを使用して測位誤差が補正できるのか試してみた。

MSASとはMTSAT Satellite-based Augmentation Systemの略で、地上に設置された受信局でGPS信号を受信し、信号に含まれる誤差を補正するためのGPS補強情報を作成し、赤道上空3万6千kmの軌道にあるMTSAT(Multi-functional Transport Satellite)を經由してGPS受信機に送られるシステムのことである。また、このような静止衛星を利用した補強システムの総称をSBAS(Satellite-Based Augmentation System)という。

図4.3-1は2007年11月10日の正午、GPS受信機、AEK-4Tを用いて台車に受信機をのせSBASの機能をon、offにして図2.2-3にあるA D C Bの順にラインに沿って歩いて一周したときの結果である。

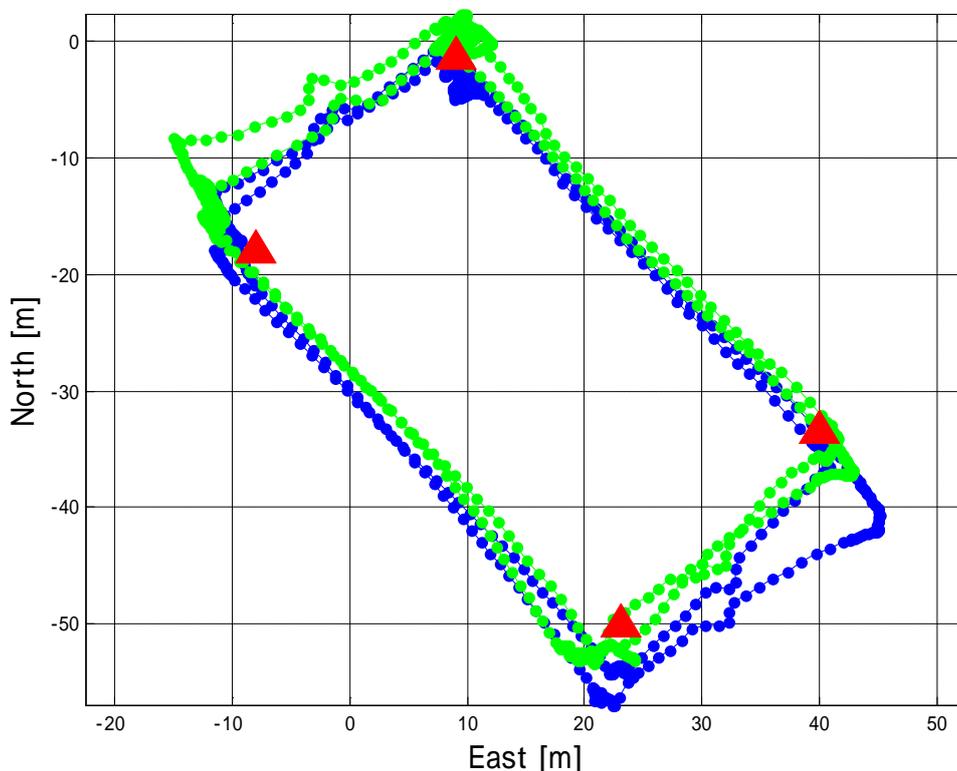


図 4.3-1 SBAS の ON/OFF

緑のラインが SBAS の ON にしたとき、青のラインが SBAS を OFF にしたとき赤の三角形が waypoint、赤の矢印進行方向である。この図をみるかぎり SBAS の補正はできていないように見える。そこで、受信機の SBAS の補正ができていないと仮定して SBAS の補正が可能な ublox 社製の受信機のをを用いて再度 SBAS を ON、OFF にし大学の第 4 実験棟の屋上のアンテナを使用して評価した。図 4.3-2 は X、Y 方向、図 4.3-3 は高さ方向の変位を表している。

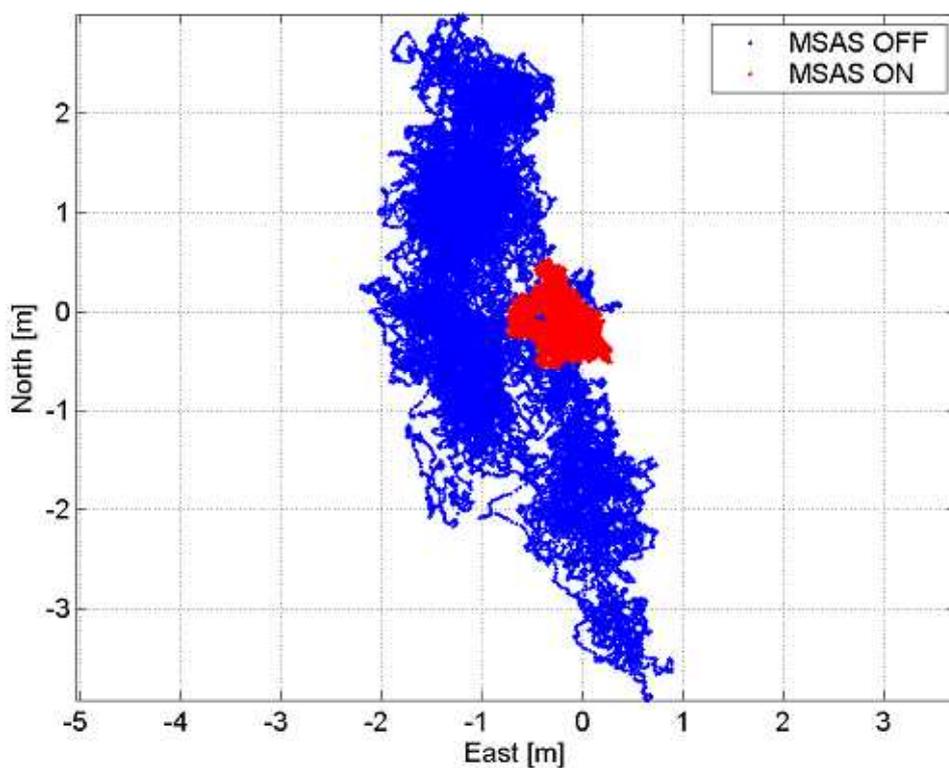


図 4.3-2 X、Y 方向の変位

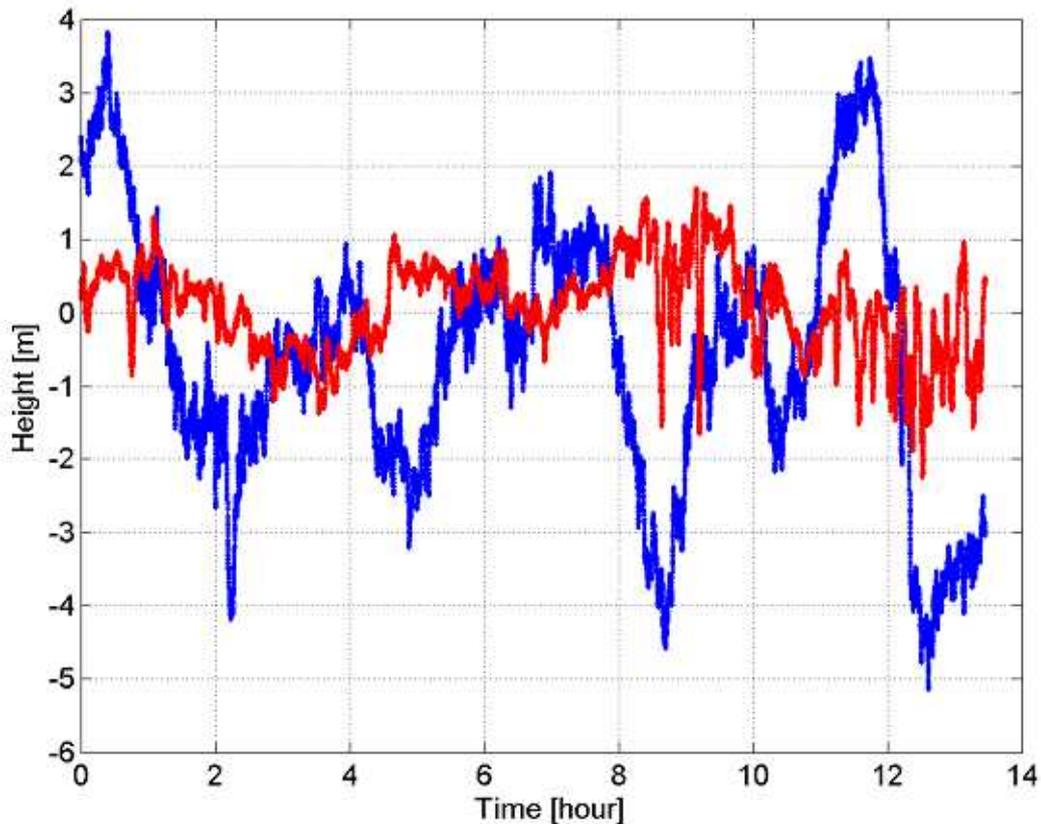


図 4.3-3 高さ方向の変位

図 4.3-2、図 4.3-3 より SBAS を ON にしたとき、X 方向に約  $\pm 1\text{m}$ 、Y 方向に約  $\pm 3.5\text{m}$  あった誤差が X、Y 方向ともに約  $\pm 30\text{cm}$  以内の誤差になった。また、高さ方向も約  $\pm 4\text{m}$  あった誤差が 1、2m の範囲に収まっている。このことから AEK-4T 受信機では SBAS の補正が利用できないことがわかった。

### 4.3.2 オフセット補正

MSAS の補正情報は u-blox 社製の AEK-4T では利用できなかった。そこで、ロボットカーの走行は短時間なので、大気遅延の変化が微小であると仮定しスタート地点が既知であることを利用して、その地点の測位誤差を補正值（オフセット）として用いることにより誤差を修正する。

まず、低い仰角の衛星の情報を取得すると、大気を通過する距離が長いので大気遅延の変化が一定ではない。また、マルチパスの影響も受けやすい。図 4.3-4 は 2007 年 11 月 12 日の正午、GPS 受信機、AEK-4T を用いて台車に受信機をのせ図 2.2-3 にある A D C B の順に仰角マスクをそれぞれ 5 度ずつ変えていきラインに沿って一周したときの結果で

ある。

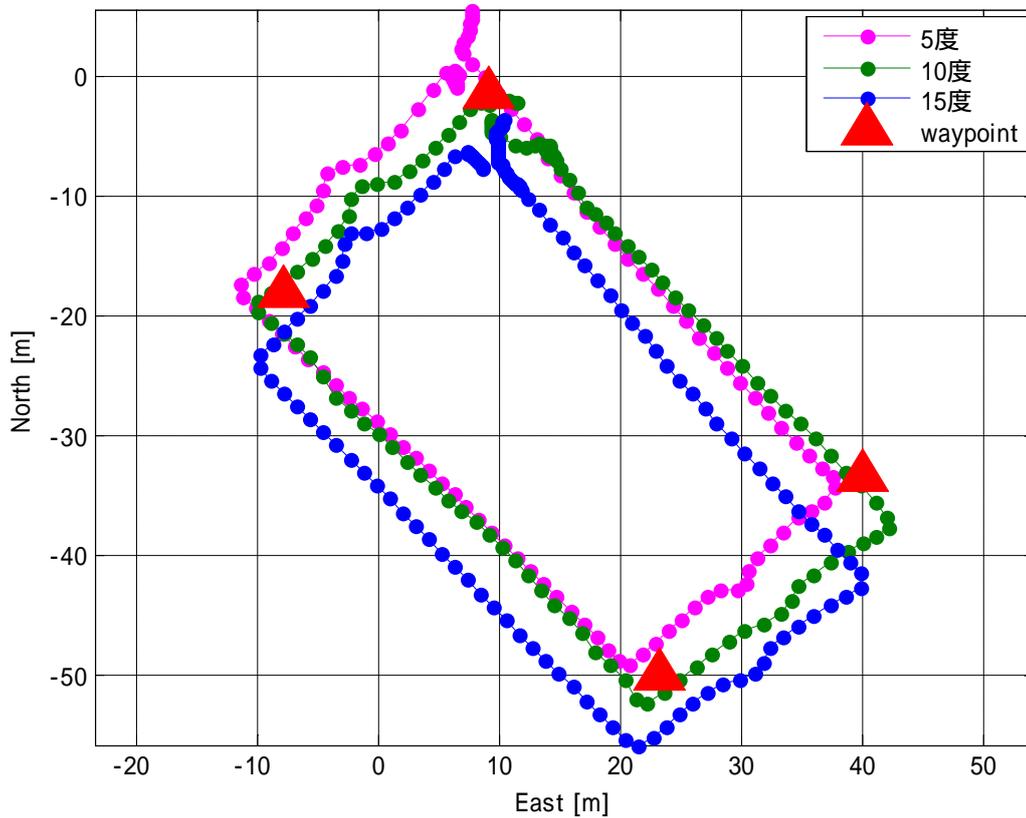


図 4.3-4 仰角 MASK を 5 度、10 度、15 度としたとき

仰角マスクを 5 度としたとき、受信した衛星数は 9、A と走行軌跡の四隅のうち A 点に近い角との誤差は約 3m、D との誤差は約 4m、C との誤差は約 3m、B との誤差は約 4m であった。

仰角マスクを 10 度としたとき、受信した衛星数は 7、A と走行軌跡の四隅のうち A 点に近い角との誤差は約 1m、D との誤差は約 4m、C との誤差は約 2m、B との誤差は約 2m であった。

仰角マスクを 15 度としたとき、受信した衛星数は 5、A と走行軌跡の四隅のうち A 点に近い角との誤差は約 3m、D との誤差は約 3m、C との誤差は約 3m、B との誤差は約 3m であった。また、waypoint をそれぞれ直線で結んだときの線と走行軌跡の線は平行している。

このことより、仰角マスクを大きくしていくと受信できる衛星数は減っていくが、走行軌跡のラインが直線に近づいていく。

この結果より、低い仰角の衛星の情報を受信すると測位誤差が大きくなり、また場所ご

とによって測位誤差が変化してしまう。そこで、15、20、25 と仰角をあげてみる。図 4.3-5 は図 4.3-4 と同じ方法で実験を行ったときの結果である。

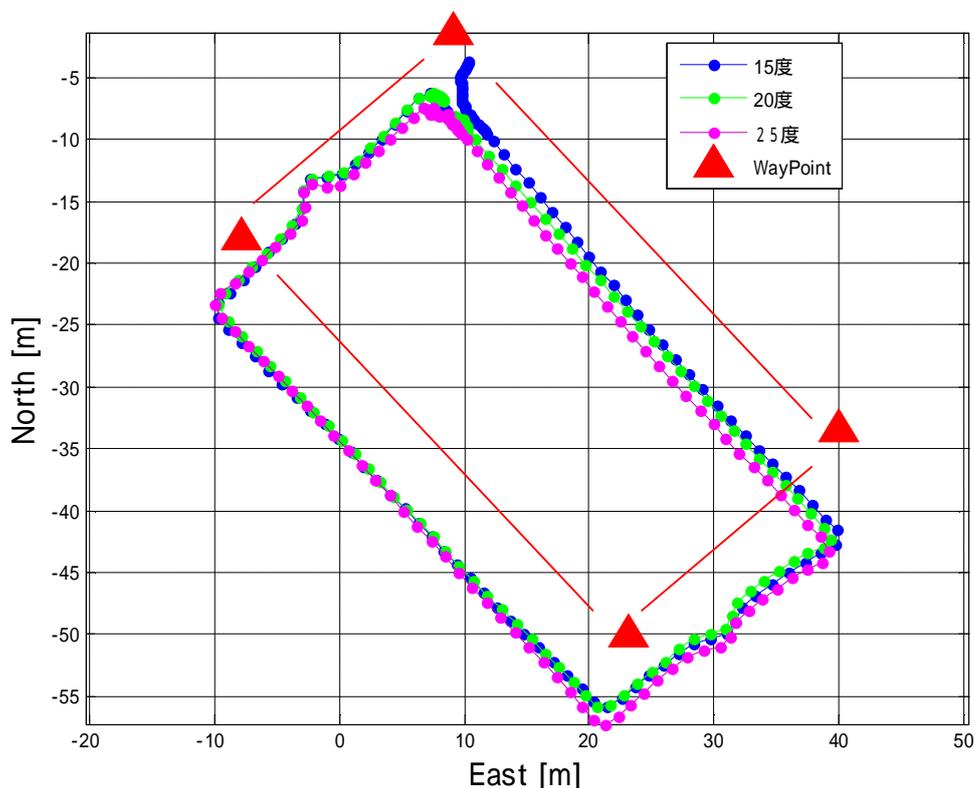


図 4.3-5 仰角 MASK 15 度、20 度、25 度するとき

仰角 15 度以上の衛星の情報を受信したとき、受信した衛星数の平均は 5 基、20 度するときの衛星数の平均は 5 基、25 度するときの衛星数の平均は 5 基であった。また、図 4.3-4 と比較すると GPS 受信機の位置情報によって描かれた軌跡とライン上の軌跡が平行していることから、一定の測位誤差でずれている。このことから、スタート地点の waypoint とそこでの GPS 受信機によって得られた位置情報との差をオフセットとし、その差だけ走行軌跡をずらしてやれば誤差は解消される。

図 4.3-6 は waypoint C と走行軌跡の四隅のうち C 点に近い座標との差をオフセットとして用いて、それを補正したときの結果である。

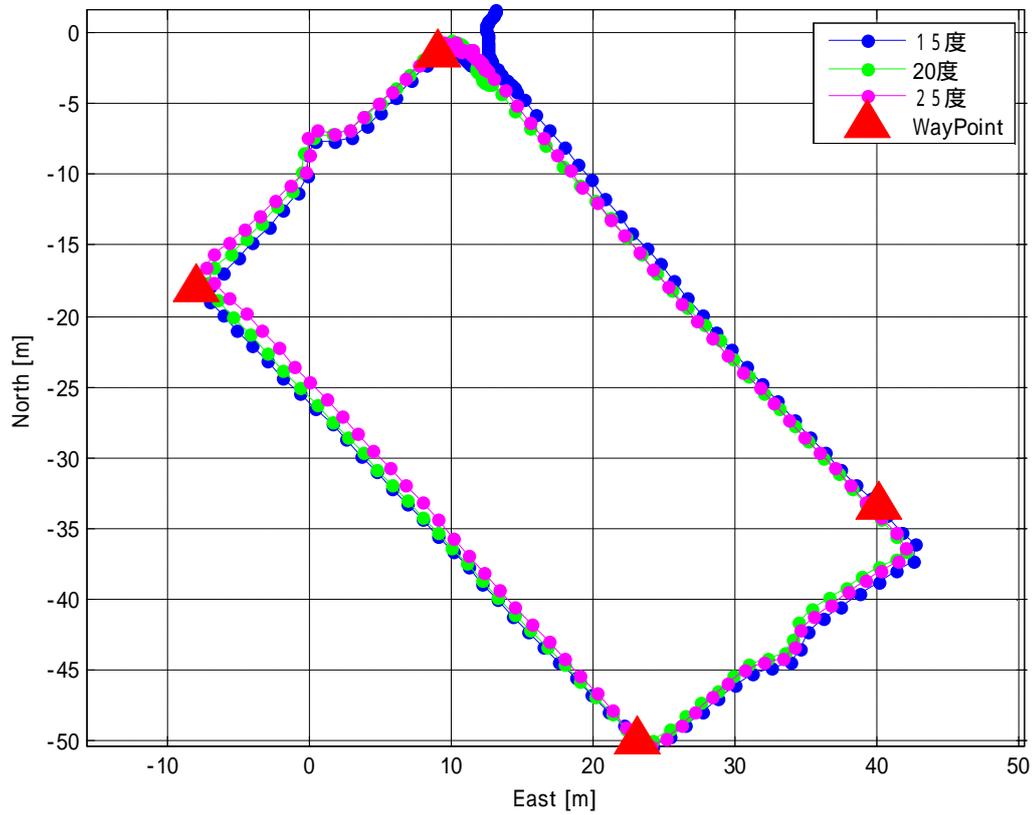


図 4.3-6 オフセット補正後

これにより、補正前に見られたオフセットは改善されており、WayPoint を通過出来た。

## 第5章 ロボットカーの走行結果

### 5.1 ロボットカーコンテストの競技ルール

走行競技場、図1に規定する競技場で行われる。競技場は、準備作業エリア、ホームエリア、競技フィールド、waypoint から成る。ただし、waypoint の数や位置は、競技開催当日にアナウンスされる。また、競技場のフロアは平坦な地面、アスファルト、または石畳とする。なお、競技場のサイズや形状は、開催場所により変更されることもある。

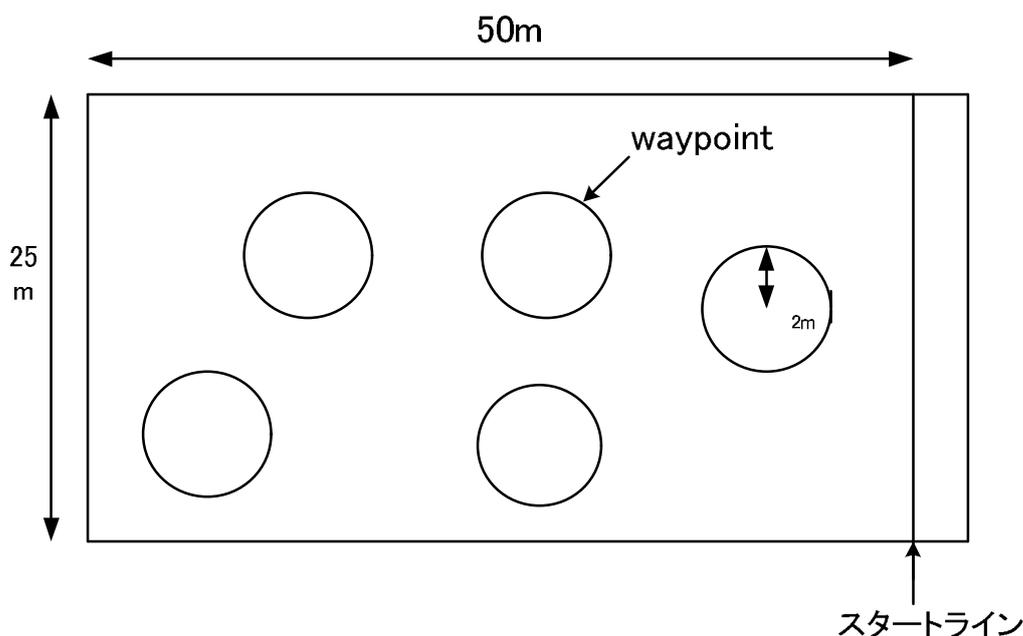


図1 競技場

参加するロボットカーは、競技中の姿勢の高さ 300mm、幅 450mm、奥行き 450mm 以下とし、重量は合計で 100kg 以下とする。また、ロボットカーは主な航法センサーとして GPS 受信機を用いることとする。これらレギュレーションは走行前の機体審査でチェックされ、満足していないロボットカーは走行競技に参加できない。また、すでに提供されている配信サービスの利用以外に、独自に基準点を設置し、補正データ等の配信を行うことを禁止する。

各競技の持ち時間は、準備時間をおよび競技時間を含めて最大 15 分とする。競技者は主審による競技開始の合図をもってロボットカーと共に準備作業エリアに入り、準備作業を始める。準備作業エリア以外での作業は禁止する。

準備作業を終えたロボットカーは、ホームエリア内から自立的にスタートラインを通過し、競技エリアに入らなければならない。ロボットカーがスタートラインを通過した時点

で競技時間の計測が開始される。ロボットカーは持ち時間以内に指定された waypoint を通過後、自立的にホームエリア内に戻ってこなければならない。つまり、準備時間が短ければ短いほど、競技時間を長く取ることができる。ロボットカーがスタートラインを通過した後は、ホームエリア内に戻るまで、人為的操作を一切くわえることはできない。

また、競技時間内であれば、競技者はリトライを宣言することができる。リトライは、それを主審が認めたときに適用される。リトライが認められた時点で、それまでの得点はゼロとなり、競技者はホームエリアからロボットカーを再度スタートさせることができる。

競技者は、主審による競技開始の合図をもってロボットカーと共に準備作業エリアに入り、準備作業を開始する。この準備作業では、waypoint の入力を始め、各センサーの初期化作業等を行うことができる。準備作業エリアには、作業のためのテーブルや電源が事前に用意されている。競技者は、準備作業に必要な道具類、パソコン等を準備作業エリア内に持ち込むことができる。なお、開催場所によっては、作業用の電源などが準備できないこともある。

準備作業を終えたロボットカーは、ホームエリア内に設置しなければならない。そのときの向きは任意であるが、ロボットカーの一部がスタートラインに接してはならない(図2参照)。また、ロボットカーは準備作業時間内に自立的にスタートラインを横切り、競技エリア内に入らなければならない。ただし、準備作業時間内であれば、競技者はロボットカーをホームエリア内の任意の位置に何度でも移動させることができる。

なお、ホームエリア内には、既知の waypoint が一点だけ準備される。ロボットカーはこの waypoint をスタート地点として選んでも良い。

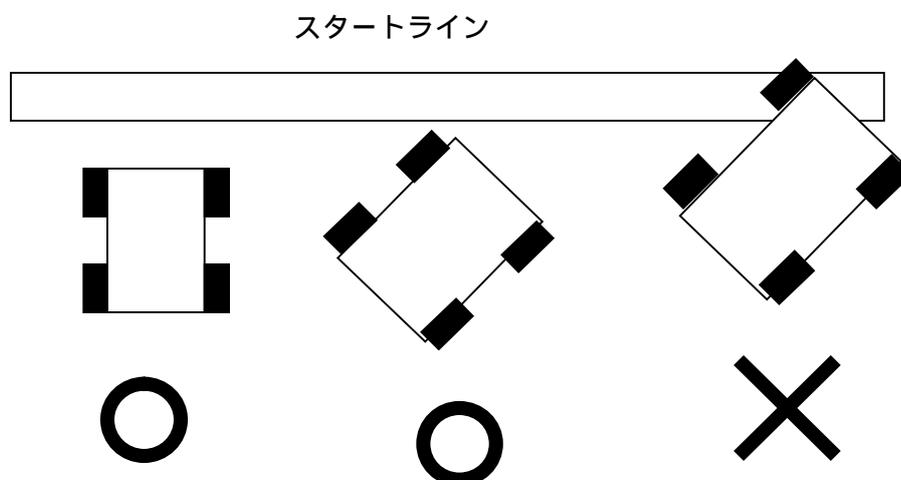


図2 スタート位置

### ロボットカーの自律走行

ロボットカーがスタートラインを通過した後は、ホームエリア内に戻り停止するまで、人為的操作を一切加えることができない。ロボットカーはしていされた waypoint を自立的に通過し、競技時間内にホームエリア内に戻らなければならない。

### 競技の終了

以下のいずれかの時点で競技終了とする。

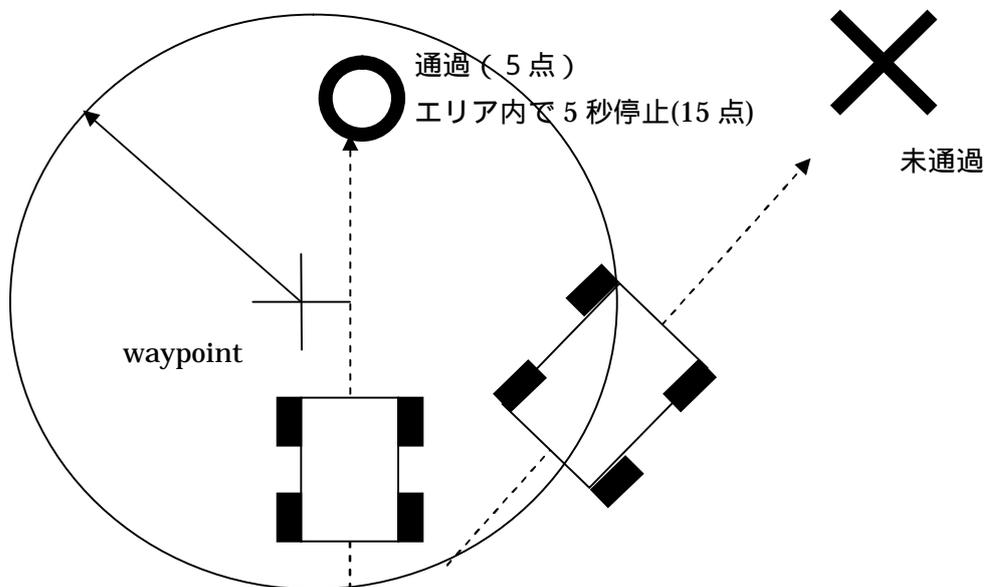
- (1) 競技開始後、15分が経過したとき
- (2) 走行を終えたホームエリア内に戻ったロボットカーに競技者が触れたとき
- (3) 競技者が終了を宣言したとき

### 得点

走行競技は waypoint を通過する正確さと速さを競い、競技終了時の合計ポイントを得点とする。合計ポイントにはペナルティも含まれる。さらに、高度な技術が要求されるが高得点を狙えることのできるチャレンジコースも用意される。

### waypoint の通過

waypoint を中心とする半径 2メートルの円内をロボットカー全体が通過することで 5点獲得さらにエリア内で 5 秒間以上停止した時点で 15 点加算される。(図3)ただし、一度通過した waypoint をその後通過しても、ポイントは加算されない。通過のみの waypoint 内にその後停止した場合は、通過ポイント無しの 15 点のみが加算される。



### 図3 waypointの通過

ロボットカーがスタートラインを通過した時点から、走行を終えホームエリア内に戻ってきたロボットカーに競技者が触れるまでを競技時間とし、以下のようにポイントが加算される。また、競技時間はリトライによってリセットされない。

- (1) 5分未満：10点
- (2) 5分以上10分未満：5点
- (3) 10分以上：0点

#### ペナルティ

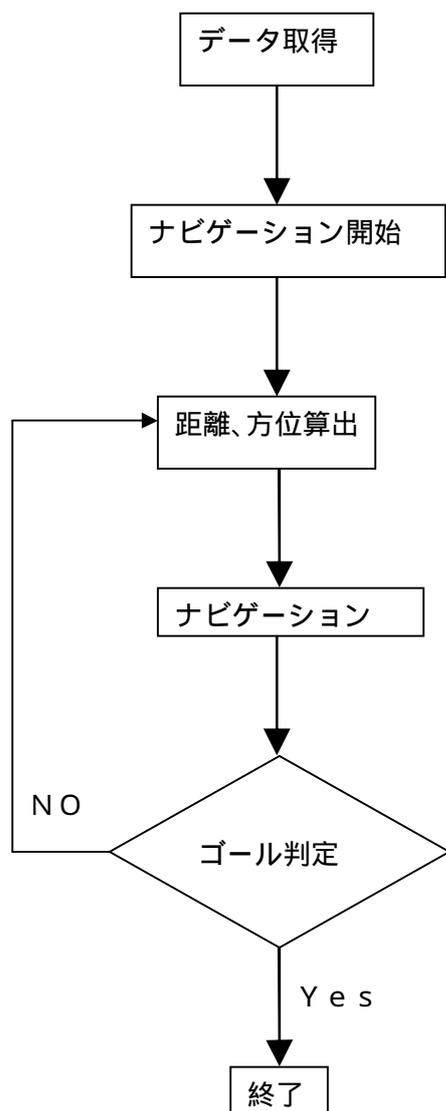
ロボットカーが競技時間以内に帰ってこれなかった場合、マイナス20点のペナルティとする。また、本競技のルールに違反した主審が判断し警告した場合、警告毎にマイナス10点のペナルティが課せられる。

#### リトライ

持ち時間内であれば、競技者はリトライを宣言することができる。リトライは競技者が主審に対して宣言し、それを主審が認めたときに適用される。リトライが認められると、ペナルティを除くすべての得点が無効となる。競技者は準備エリアでロボットカーの調整など行うことができるが、競技時間はリトライによってリセットされない。準備が整い次第、競技者は主審に再スタートを宣言し、主審の許可を得た後、ホームエリアからロボットカーを再スタートさせることができる。

## 5.2 ロボットカーの制御アルゴリズム

第2章から第4章までのことを用いてロボットカーを制御する。制御用ソフトウェアはC言語を用いて開発した。制御は以下のチャートのようにになっている。



ロボットカーをスタート地点に置き、電源を入れる。しばらくするとGPS受信機から位置情報が取得される。スタート地点が既知であることから、受信機から位置情報との差をオフセットとする。また、今の位置から目的地までの距離を算出する。マイコンのスイッチを入れることによりロボットカーは動き始める。GPSから真北から方位角を取得し、これを用いて目的地までの方位角を算出、操舵角を制御ゲイン  $K=0.3$  により調整し

waypoint へ向かう。もしもロボットカーが waypoint の 1 m 以内に入っていれば 5 秒間停止する。さらに再度 2m 以内に入っているか調べ、入っていれば次の waypoint にきりかわり、入っていなければ同じ waypoint で停止し判断を行う。3 回この判断を行い、3 回と入っていないと判断されれば、次の waypoint に進む。

### 5.3 ロボットカーの走行結果

実験用に設置した waypoint 使用して、ロボットカーを走行させた結果を図 5.3-1 に示す。

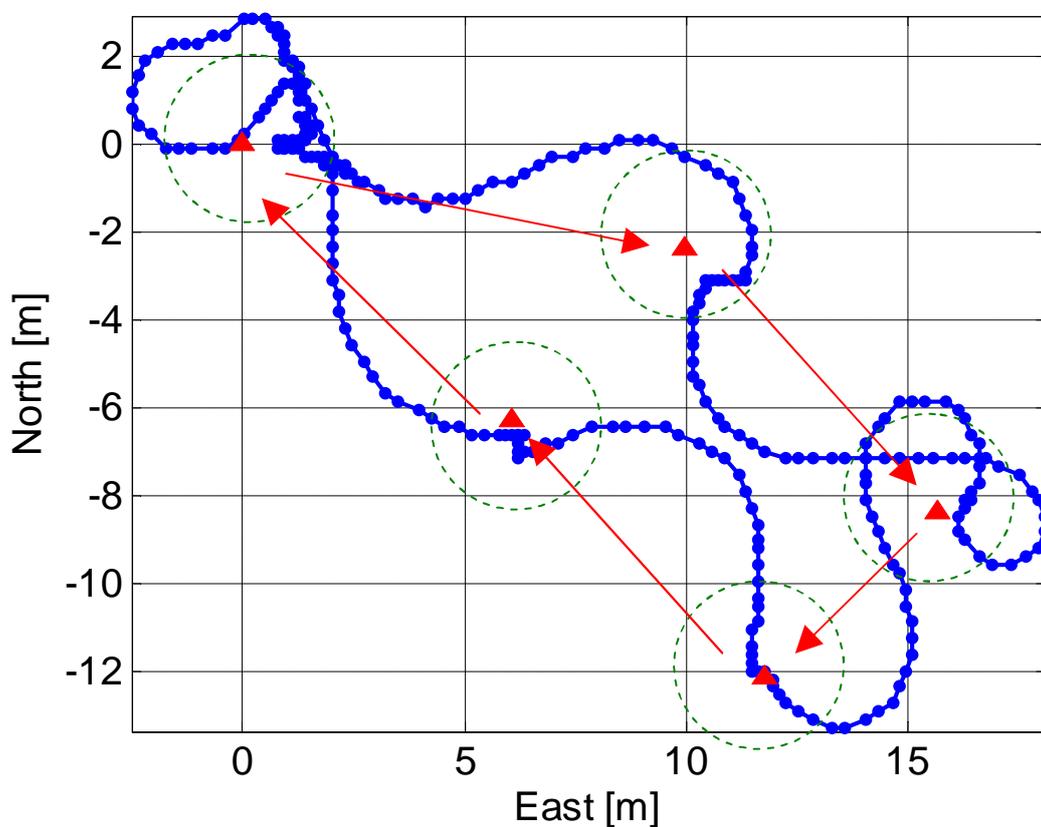


図 5.3-1 ロボットカーの走行結果

赤の三角形が waypoint、赤い矢印が目標とする走行のライン、青い線が実際の走行軌跡、緑の円が waypoint を中心とした半径 2m の円である。

## 第6章 総括

### 6.1 結論

図 5.3-1 の結果から実験用に設置した Waypoint を自律的に通過させることができた。また、この図からはわかりにくいですが Waypoint から 2m 以内で一時停止した。しかし、まっすぐ waypoint に向かいたいのには蛇行しながら waypoint に向かっている。これは、GPS 受信機によって現在の位置情報と前の位置情報を用いて方位角を求めているため正確な方位角が得られないことが原因と考えられる。また、GPS 受信機の位置情報を用いたオフセット補正を行うと、受信できる衛星の数によって精度が変わってしまうので今回のロボットカーコンテストのように短時間の走行なら問題ないが長時間の走行となる場合や長距離を走行する場合は、受信できるの衛星数や大気遅延量の変化によってオフセット補正が機能しない場合が考えられる。

### 6.2 今後の課題

今回は GPS の情報をのみを使用して、ロボットカーを制御したが、方位角が GPS 受信機の精度によって変化してしまうなど正確な値を得ることができない。そこで、ジャイロや地磁気センサーを用いてより正確な方位角をもとめることにより、スムーズな走行を実現できると考えられる。また、速度も加速度センサーなどを使い、正確に算出することによって、位置の推測が可能となる。また、今回速度が 1m/s と一定であることから操舵角の比例制御のみしか行わなかったが今後は微分 (I)、積分 (D) を加えて、PID 制御を行い、速度を始め加速させていき、目的地の近くで減速し停止するような速度制御を実行すれば、より速く、正確かつスムーズに waypoint を通過していき、図 5.3-1 のような蛇行する結果が改善される。

## 謝辞

本研究において、あらゆる面でご指導ご鞭撻頂きました東京海洋大学情報通信工学研究室の安田明生教授に心から深く感謝申し上げます。

本研究において、GPSの基楚、有益な御示唆をしていただいた久保信明準教授に心より感謝いたします。

講師の海老沼先生には一から本研究の取り組み方、データの取り方、プログラミングの基本技術をご指導いただきました。心より感謝いたします。

最後に情報工学研究室の皆様心から感謝の意を表します。

## 参考文献

- ( 1 ) Pr atop Misra and Per Enge 日本航海学会 GPS 研究会訳  
精説 GPS 基本概念・測位原理・信号と受信機
- ( 2 ) 出村公成、簡単！実践！ロボットカーシミュレーション、森北出版
- ( 3 ) 土屋淳・辻宏道、新・GPS 測定の基礎、社団法人 日本測量会

- 第 1 章 序論
  - 1.1 はじめに
  - 1.2 研究の目的と背景
- 第 2 章 座標変換
  - 2.1 局地座標変換
  - 2.2 局地座標変換の簡略化
- 第 3 章 ロボットカーの制御
  - 3.1 フィードバック制御による軌道修正
  - 3.2 シミュレーション
- 第 4 章 GPS 受信機
  - 4.1 ユーザー位置計算
  - 4.2 誤差要因
  - 4.3 測位誤差による補正
- 第 5 章 ロボットカーの走行結果
  - 5.1 ロボットカーコンテストの競技ルール
  - 5.2 ロボットカーの制御アルゴリズム
  - 5.3 ロボットカーの走行結果
- 第 6 章 総括
  - 1.1 結論
  - 1.2 今後の課題

卒業論文

平成19年度

(2007)

自律型 GPS ロボットカーの制御

東京海洋大学  
海洋工学部 海事システム工学科

0421017 川寄洋平

指導教官 久保信明 準教授

