

オドメトリと座標系

1. オドメトリについて

今回説明する内容はロボット上のマイコンボード (sh-vel) とパソコン (ypspur-coordinator) で行われていることです。オドメトリ (odometry) とは車輪型移動ロボットにおける車輪やステアリングの回転角度の計算から、それぞれの移動量を求め、その累積計算からロボットの位置を推定する手法の総称です。

1.1 オドメトリの計算方法

ロボットの位置は一般的に

$$\begin{aligned}\theta(t) &= \int_0^t \omega(\tau) d\tau + \theta(t_0) \\ x(t) &= \int_0^t v_{xGL}(\tau) d\tau + x(t_0) \\ y(t) &= \int_0^t v_{yGL}(\tau) d\tau + y(t_0)\end{aligned}\quad (1)$$

の計算をすることで求めることができる。ここで、 $(x(t_0), y(t_0), \theta(t_0))$ はロボットの初期位置である。

1.2 独立2輪駆動型ロボットのキネマティクス (運動学)

本研究室では一般的に PWS (Power Wheeled Steering: 独立2輪駆動) 型のロボットが使用されている。PWS の場合は図 1 に示すキネマティクスとなっている。PWS は並進速度 v 、角速度 ω 成分しかもたず、横方向には動かないことから、世界座標系でのロボットの速度は

$$\begin{aligned}v_{xGL}(t) &= v(t) \cos(\theta(t)) \\ v_{yGL}(t) &= v(t) \sin(\theta(t))\end{aligned}\quad (2)$$

と表すことができる。この式 2 を式 1 に代入すると、オドメトリの計算式は、

$$\begin{aligned}\theta(t) &= \int_0^t \omega(\tau) d\tau + \theta(t_0) \\ x(t) &= \int_0^t v(\tau) \cos(\theta(\tau)) d\tau + x(t_0) \\ y(t) &= \int_0^t v(\tau) \sin(\theta(\tau)) d\tau + y(t_0)\end{aligned}\quad (3)$$

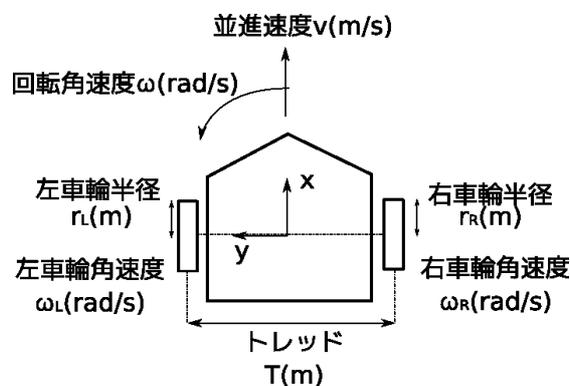


Fig. 1 Odometry

と表される。

これにより車輪から速度・角速度を知ることができれば、オドメトリが計算できるようになる。

1.3 コンピュータ内部での計算

実際にどのような計算をコンピュータで行っているかと言うと、時々刻々と以下の様な流れで計算を行っている。

- 1) 各車輪のエンコーダから値を取得し、各車輪の角速度を計算する。
各車輪の角速度 ω_R に関しては、左右動輪についているエンコーダを使用して、

$$\omega_R(t) = \frac{\text{count}(t) - \text{count}(t - \Delta t)}{\text{Encoder} \times \text{Gear} \times \Delta t} \times 2\pi \quad (4)$$

と表すことができる。このとき

count(t):時刻 t におけるエンコーダの値 [pulse]

Encoder: 1 回転当たりのエンコーダのパルス数 [pulse]

Gear: ギア比 (タイヤが 1 回転するときのエンコーダの回転数) [・]

Δt : サンプルング周期 [s]

である。左車輪も同様に計算できる。

- 2) ロボットの速度、角速度を求める。

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r_R}{2} & \frac{r_L}{2} \\ \frac{r_R}{T} & -\frac{r_L}{T} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (5)$$

- 3) オドメトリの計算
オドメトリの計算式である式 3 を離散時間でも計算できるように漸化式に変形し、k 番目のときの自己位置推定の結果を $P_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T$ で表現すると、

$$P_{k+1} = P_k + \begin{bmatrix} v_k \cos(\theta_k) \\ v_k \sin(\theta_k) \\ \omega_k \end{bmatrix} \Delta t \quad (6)$$

1 ~ 3 の計算を繰り返すことによって、時々刻々のオドメトリデータを計算することができる。SH ボード上からはエンコーダ値が 5ms 毎に送られており、パソコン上で計算しています。

1.4 実際の使用について

YP-Spur でオドメトリにアクセスするには `Spur_get_pos_**(double *x, double *y, double *th)` 関数 (**は座標系を表す 2 文字) を使用することでアクセスすることができます。また、YP-Spur の実装では、SH ボード上からエンコーダ値は 5ms でデータが送信されていますので、これより短い間隔でオドメトリを取得しても同じ値が返ってくるだけとなります。

2. 座標系

2.1 座標系とは

座標系 (coordinate system) とは、原点の位置、座標軸の向き、長さの単位を決めたものであって、点の位置を座標で表すようにしたシステムです。例えば座標系には以下のようなものがあります。

- 直交座標系
- 斜交座標系
- 極座標系
- 一般化座標系

山彦は 2 次元平面を移動することを考えているため、YP-Spur では 2 次元直交座標系をサポートしています。2 次元平面の X 軸、それに垂直な平面上の軸を Y 軸、さらに、X 軸に対するロボットの姿勢を θ とし、ロボットの位置・姿勢を (x, y, θ) の 3 つのパラメータで表現します。また、長さの単位は SI 単位系となっており、座標系の原点はユーザが定義します。

3. YP-Spur で提供される 3 つの座標系

YP-Spur では 3 つの座標系を使用することができます。各座標系の関係は、図 2 に示します。

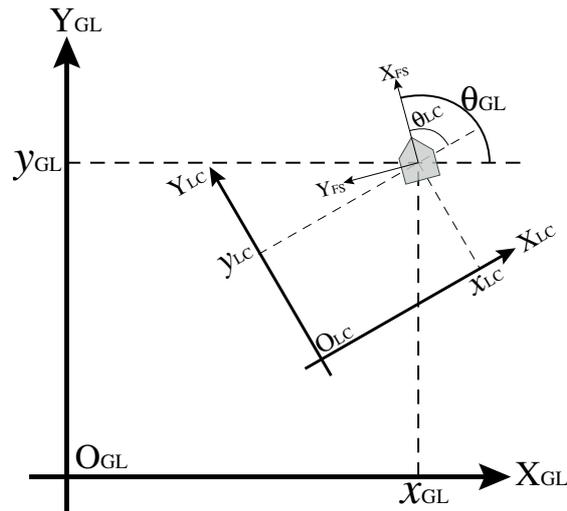


Fig. 2 Interrelation among coordinates of YP-Spur

FS(Front Side coordinate system) 座標系 :

ロボット座標系。ロボットに張り付けられ、ロボットと共に移動する座標系です。

GL(Global coordinate system) 座標系 :

世界座標系。一度決めたらなるべく張り替えてはいけません。屋外走行などでは GPS 座標系などを設定します。

LC(LoCal coordinate system) 座標系 :

ローカル座標系。GL 上の座標系です。プログラマによって随時設定、変更できる座標系です。一般的に位置修正などで座標系を変更したいときはこちらを変更します。

3.1 Spur_set_pos と Spur_adjust_pos

YP-Spur で座標系を設定する為の関数として、*Spur_set_pos_**(double x, double y, double th);* と *Spur_adjust_pos_**(double x, double y, double th);* の 2 つ (以下 set と adjust) があります。

set 関数を使用すると、line コマンドなどで与えた命令も同時に座標変換されるためロボットの動き方は変化しませんが、adjust 関数を使用すると、自己位置のみ修正するため、修正時に、ロボット動きが変化します。

とりあえずは set 関数はロボットの動きが変化しない、adjust 関数はロボットの動きが変化すると覚えておけば大丈夫です。

4. その他

以下に挙げる参考文献は坪内先生が書いた本で、他大学院などで教科書として使用されています。(ちなみに坪内先生はこの本で表彰されました。) 知能ロボット研究室では指定されていませんが、なぜかみんな持っている教科書です。座標系やオドメトリの話も今回話した内容よりも詳しく載っているので、読んでみると良いと思います。ロボットを作る際の参考となるので、研究室のメンバーなら買う価値はあると思います。

参考文献

- [1] 米田 完, 坪内 孝司, 大隈 久: "はじめてのロボット創造設計", 講談社, 2001.
- [2] 米田 完, 大隈 久, 坪内 孝司: "ここが知りたいロボット創造設計", 講談社, 2005.
- [3] 坪内 孝司, 大隈 久, 米田 完: "これならできるロボット創造設計", 講談社, 2007.