デッドレコニング

demura.net

デッドレコニング（dead reckoning）

- 内界センサにより自己位置を推定する
  - エンコーダ
  - ジャイロ

運動学1

x, y, θはロボット重心の位置，姿勢

$$\dot{x} = 0.5\, r\, (\omega_r + \omega_l) \sin\theta$$

$$\dot{y} = 0.5\, r\, (\omega_r + \omega_l) \cos\theta$$

$$\dot{\theta} = 0.5\, r\, (\omega_r - \omega_l) / d$$

0.5はロボット重心の速度だから
左右輪の平均値

ここで，
r: 車輪の半径
ωr, ωl: 車輪の角速度　右，左
d: 中心から車輪の間隔

上の式を時間で積分すればよい

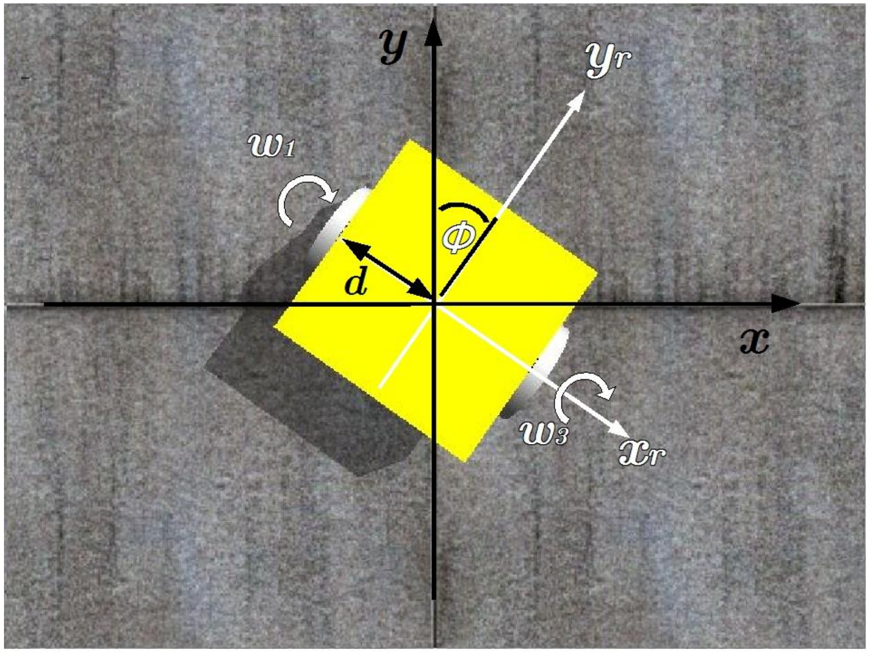
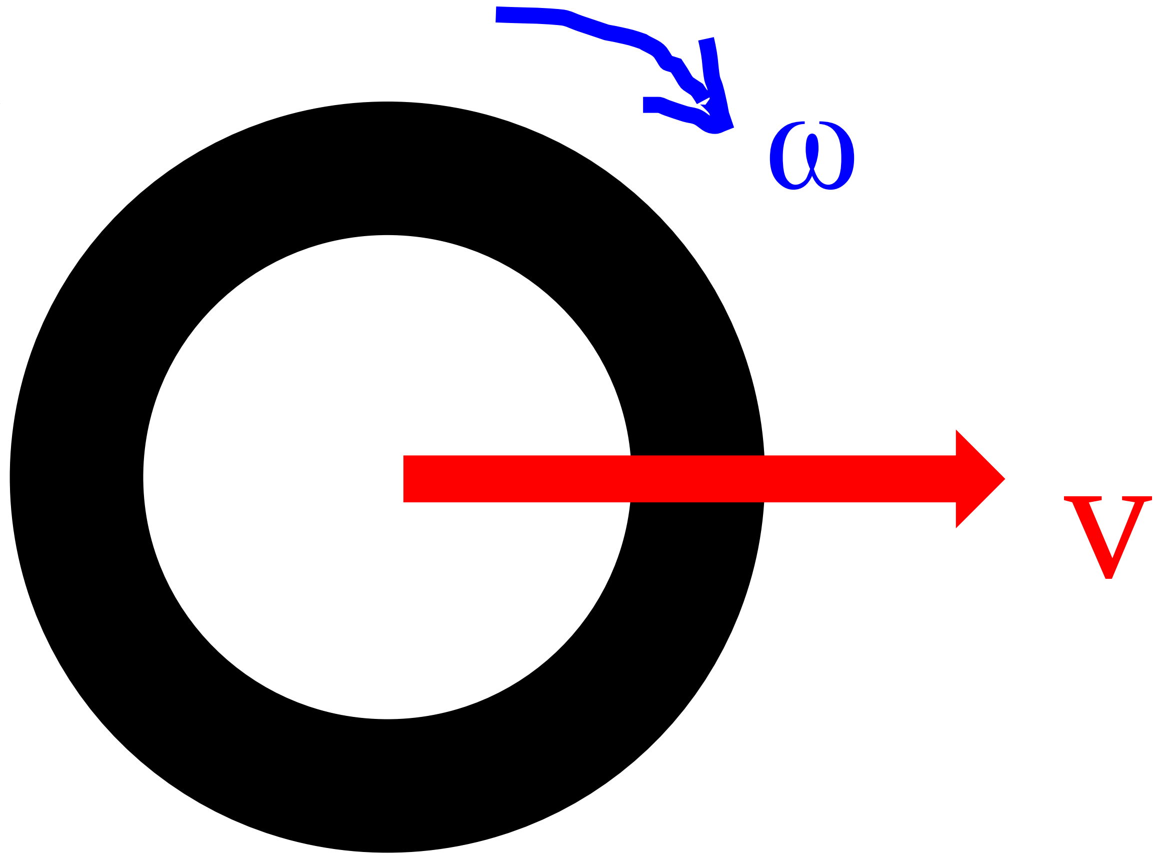


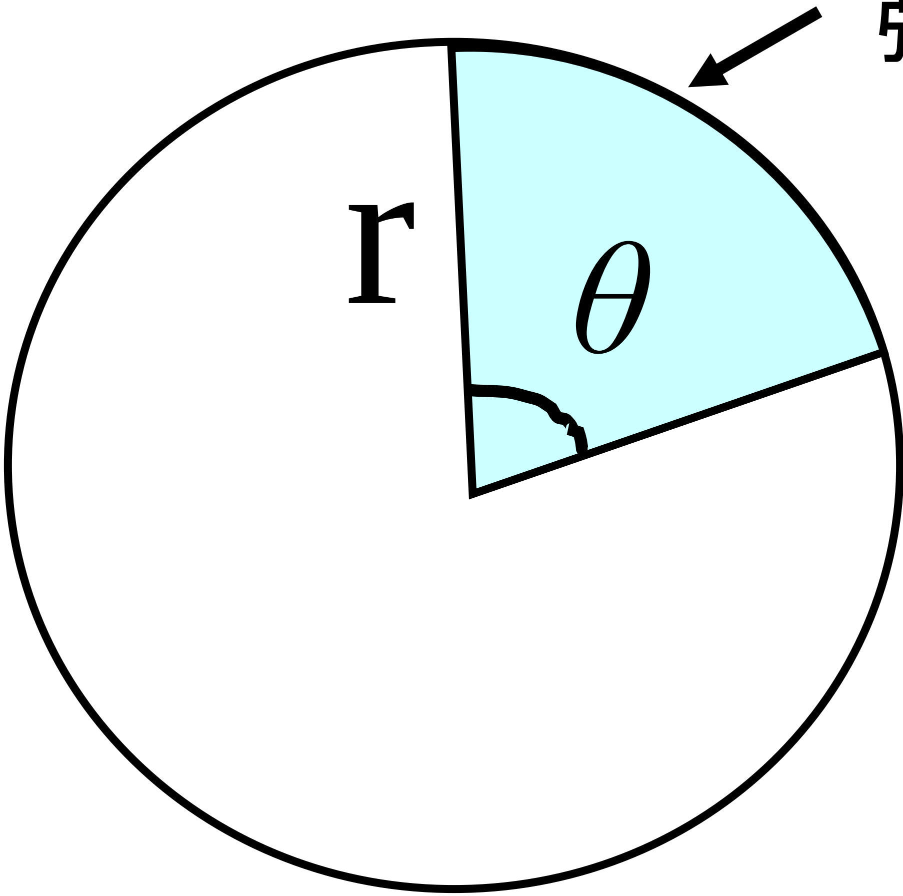
図4.11 差動駆動型ロボットの運動学
教科書P126から転載

なぜ、車輪の速度vはrω

r: 車輪の半径
ω: 回転速度
（角速度）



円周長は$2\pi r$



弧の長さ $l = r\theta$

両辺を時間で微分

$$v = r\omega$$

運動学２

$$x = x_0 + 0.5 \int_0^t r\,(\omega_r + \omega_l)\sin\theta\,dt$$

$$y = y_0 + 0.5 \int_0^t r\,(\omega_r + \omega_l)\cos\theta\,dt$$

$$\theta = \theta_0 + 0.5\, r/d \int_0^t (\omega_r - \omega_l)\,dt$$

ここで,
r: タイヤの半径
ωr, ωl: タイヤの角速度　右，左
d:中心からタイヤの間隔

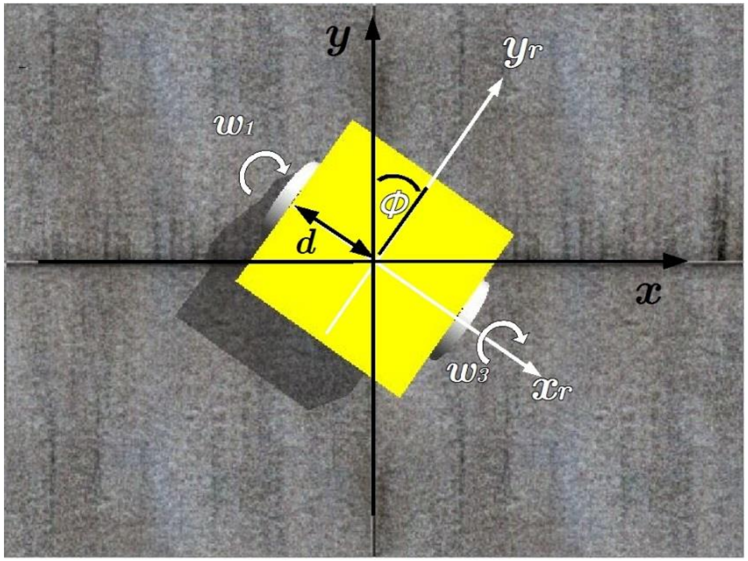


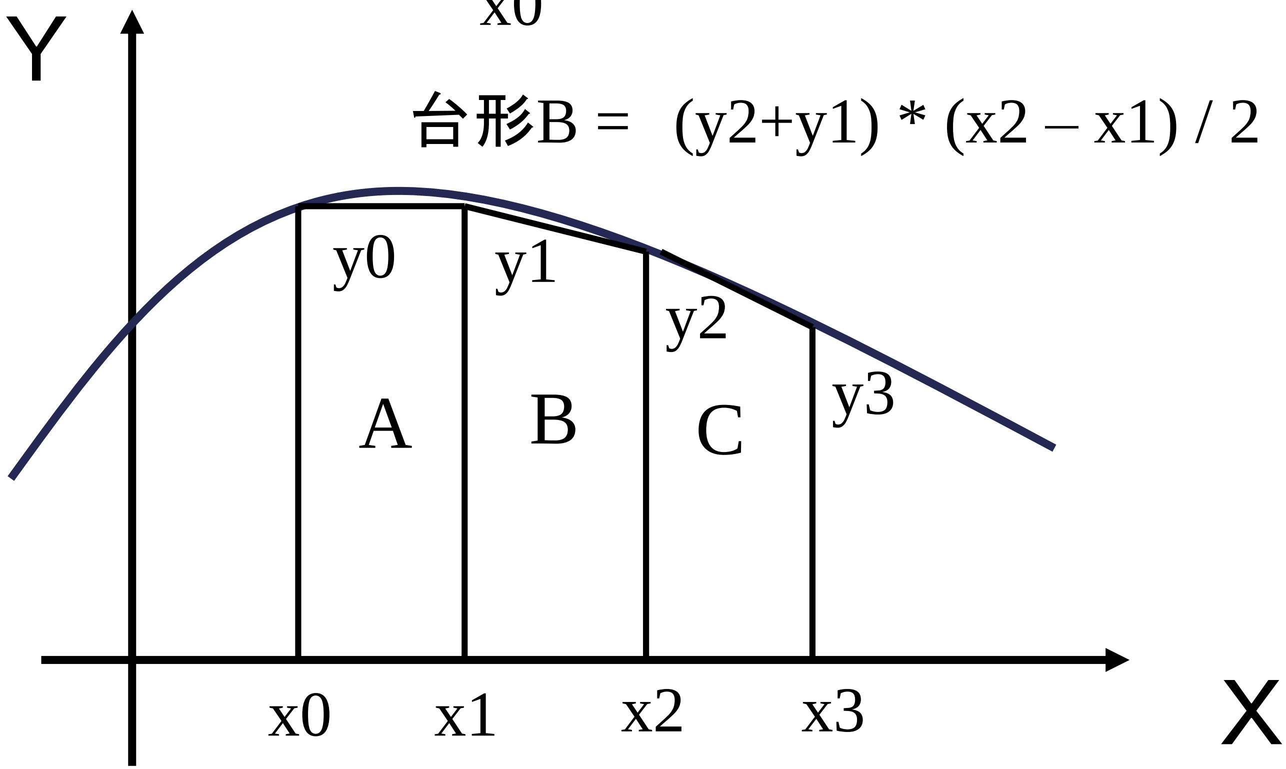
図4.11 差動駆動型ロボットの運動学
教科書P126から転載

数値積分（ロボットプログラミングＩ）

- 台形公式

$$\int_{x0}^{x3} f(x)\,dx \quad = 台形A + B + C$$

台形B = (y2+y1) * (x2 – x1) / 2



積分は微小区間の足し算である.

数値積分するには

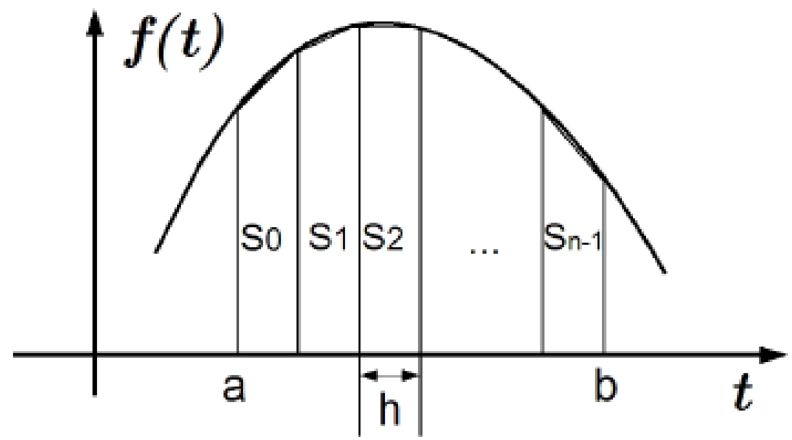


図 4.12　台形近似による積分

数式をコンピュータで解く学問は数値計算とよばれていて，ロボットをプログラミングするうえで必要な知識です．数値積分には台形法，シンプソン法，ガウス型数値積分法などありますが，ここでは一番簡単な台形法を説明します．台形法は図 4.12 のように，積分を小さな区間の足し算で近似します．

簡単! 実践！ロボットシミュレーション　P127　から転載

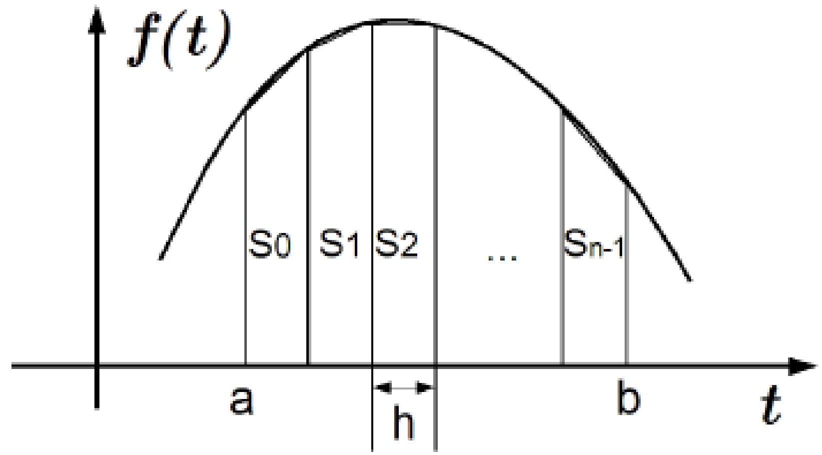


図 4.12　台形近似による積分

$$\int_a^b f(t)dt \fallingdotseq S_0 + S_1 + \cdots + S_{n-1} \fallingdotseq \sum_0^{n-1} S_i \tag{4.32}$$

ここで，S_i は小さな区間での 台形 i の面積 = 高さ × (上底 + 下底)/2 です．

$$S_i = \frac{h\{f(t_i) + f(t_{i+1})\}}{2} \tag{4.33}$$

ここで，h は台形の高さ $\frac{(b-a)}{n}$，$f(t_i)$ は上底，$f(t_{i+1})$ は下底です．また，h のことを積分の**ステップサイズ**といいます．式 (4.32) を整理すると次式になります．

$$\int_a^b f(t)dt \fallingdotseq \frac{1}{2}h\sum_{i=0}^{n-1}\{f(h \times i + a) + f(h \times (i+1) + a)\} \tag{4.34}$$

簡単! 実践！ロボットシミュレーション　P127　から転載

$$\int_a^b f(t)dt \fallingdotseq \frac{1}{2}h\sum_{i=0}^{n-1}\{f(h\times i+a)+f(h\times(i+1)+a)\} \tag{4.34}$$

これをプログラムすると以下のように表せます.

```
double a = 積分の始点, b = 積分の終点, n = 区間の分割数;
double S = 0;                                  // S は積分の値
double h = (b - a)/n;                          // h は積分のステップサイズ

for (int i = 0; i < n-1; i++) {
  S += h * ( f(h*i+a) + f(h*(i+1)+a) ) / 2;   // f() は積分する関数
}
```

簡単! 実践!ロボットシミュレーション P127 から転載

end