

---

# 歩行アルゴリズム

---

ソフトウェアセッション	11:10-12:00
1)CPUの選定	森永
2)立ち上がりテクニック	森永
3)歩行アルゴリズム	西村

NRC  
西村輝一

---

ROBO-ONE Technical Conference

---

# 歩行のいろいろ

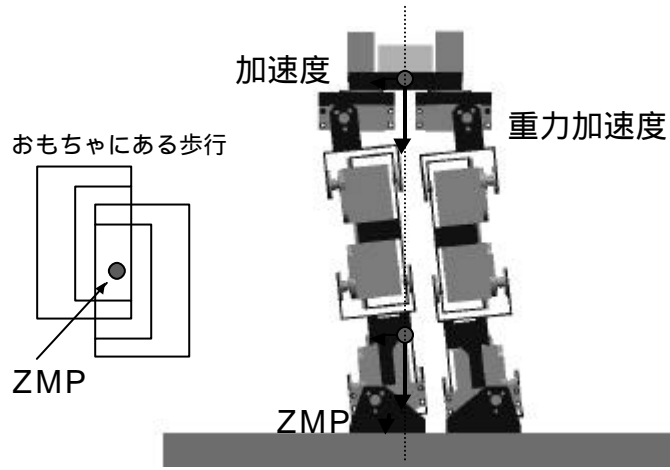
---



---

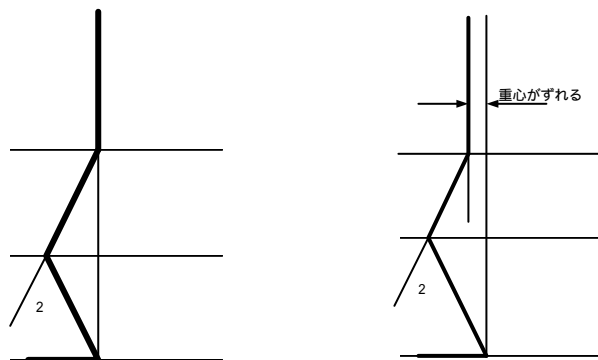
ROBO-ONE Technical Conference

## 静歩行と動歩行



## ロボットのデザイン

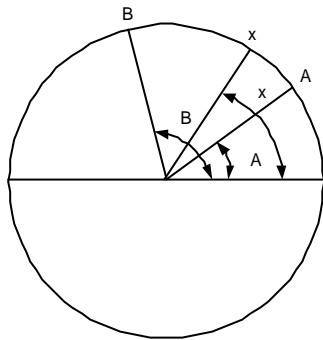
足の長さなどロボットのデザインによって計算時間や制御方法が異なる。



## 補間による動作

図はサーボモータの初期値がAにあって、その時のサーボ角度が Aとする。この位置からBの位置まで一定の角速度で移動する場合にt時間後の位置をxとし、そのサーボ角を xとし、AからBへの移動時間をhとすればt時間後の位置xの角度 xは以下の式で求まる。

$$x = A + (B - A) * t / h$$

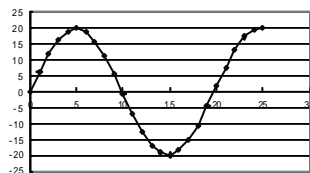
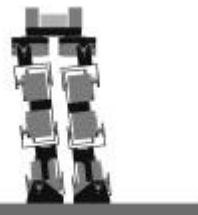
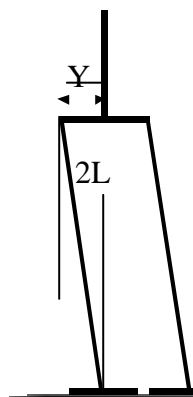


```
// iは軸番号 intervalは移動時間,diは時刻
int line ( int i, int interval, int di){
    return (s_leg[i]+(float)((e_leg[i]-s_leg[i])*di/interval));
}
```

## 重心移動の各関節角度の計算

重心移動の場合、Ymmだけ腰の部分を移動したときの は以下の式となる。足の長さを2Lとすると

$$a = \text{asin}(Y / 2L)$$



重心移動パターン

### 角度計算プログラム

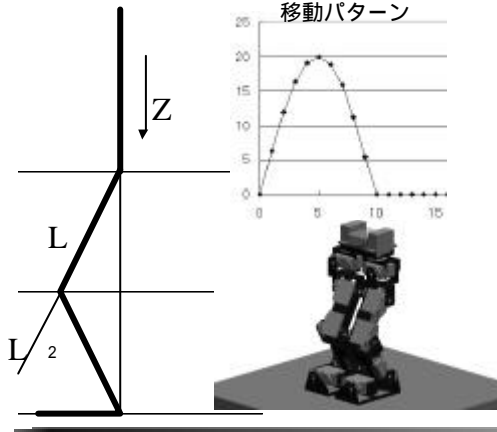
```
void y_cnt (int y){
    float a, thr;
    a=y/(2*L1);
    thr=asin(a);
    a+=thr*KKK+rln[1];
    c_rl[1]=mmchk(a);
    a+=thr*KKK+rln[5];
    c_rl[5]=mmchk(a);
    a+=thr*KKK+l1n[1];
    c_l[1]=mmchk(a);
    a+=thr*KKK+l1n[5];
    c_l[5]=mmchk(a);
}
```

## 屈伸の各関節角度の計算

屈伸の場合は

$$= \text{acos}((2L - Z) / 2L)$$

となる。

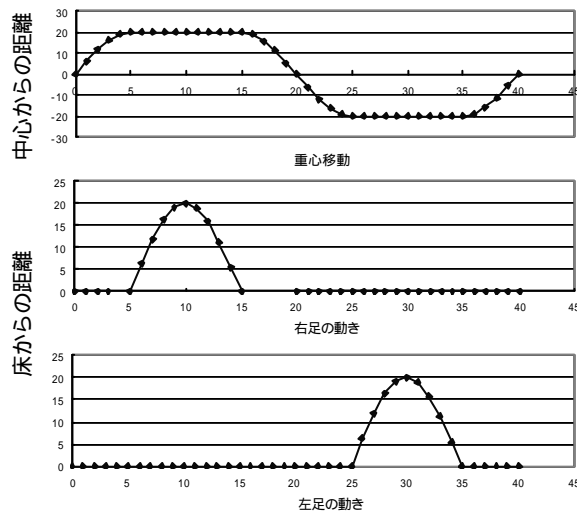


角度計算プログラム

```
//XZの距離で足をコントロール
//-----
void wr_xz(int x,int z){
float a,th0,th1,tha,thb,thc;
a=(2*L1-z)/(2*L1);
th1=acos(a);
a=x/(2*L1-z);
th0=atan(a);
tha=th0+th1;
thb=2*th1;
thc=th1-th0;
a=-tha*(KKK+150)+rln[2];
c_rl[2]=mmchk(a);
a=-thb*KKK+rln[3];
c_rl[3]=mmchk(a);
a=+thc*(KKK+100)+rln[4];
c_rl[4]=mmchk(a);
}
```

ROBO-ONE Technical Conference

## 足ふみパターンの生成



ROBO-ONE Technical Conference

## 歩行の各関節角度の計算

この場合も屈伸の場合と同じに考える。線分ab上で屈伸し、 $\theta_0$ を補正してやれば良い。

hの長さは

$$h = \sqrt{x^2 - (2L - z)^2}$$

となり $L_1 = L_2$ の時

$$\theta_1 = \arccos(h/2L)$$

$$\theta_0 = \arctan(x/(2L - z))$$

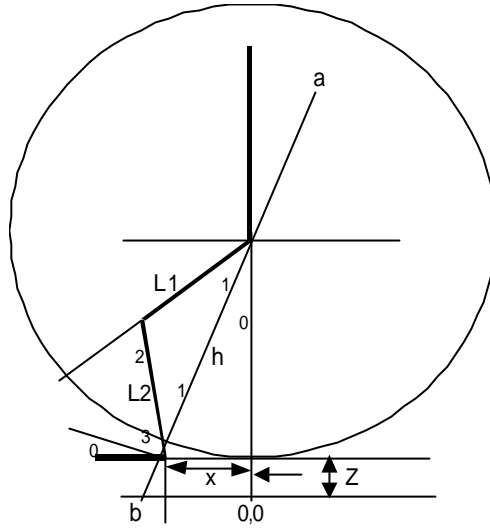
関節の角度を上から順に

a, b, cとすると

$$a = \theta_0 + \theta_1$$

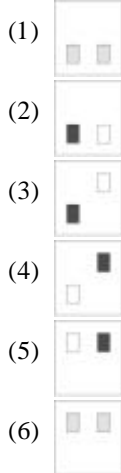
$$b = 2 * \theta_1$$

$$c = \theta_1 - \theta_0$$



## 歩行パターンの生成

(1) 一歩歩行する為の手順



(1) まず両足の中央に重心をおき(セット状態)、足を曲げる。

(2) 重心を左足に乗せる。

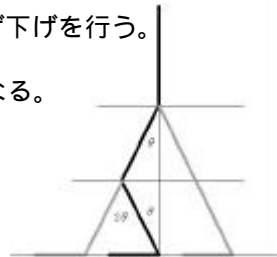
(3) 右足を前に出しながら上げ下げする。

(4) 重心を前に出す(左右の足を後ろに移動)と同時に右に移す。

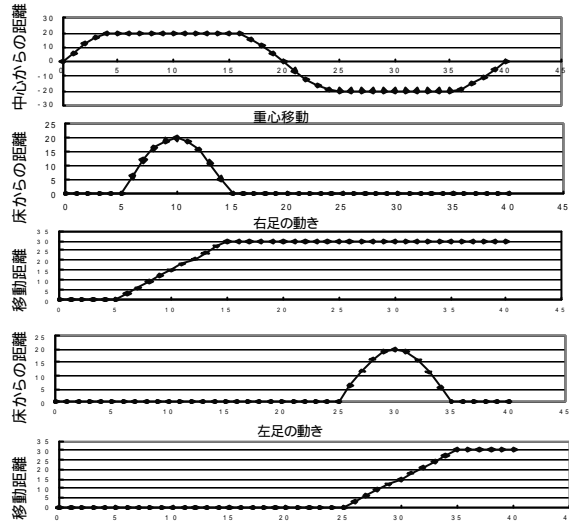
(5) 左足を前に引きつつ足の上げ下げを行う。

(6) 重心を両足の中央に移す。

(7) 足を伸ばせばもとの状態になる。



## 歩行パターンの生成



ROBO-ONE Technical Conference

## プログラミングの実際

```
void f_right(int ko,int hh,int fh,int tim){
    float tt,i;
    tx5=0;txx=0;
    while(txx<=tim){
        tt=(float)(txx*3.14/(tim));
        i=sin(tt);
        i=i*fh; //足を上げる高さ
        r_xz(hh*txx,(int)i+ko);
    }
}
```

歩行パターンの計算

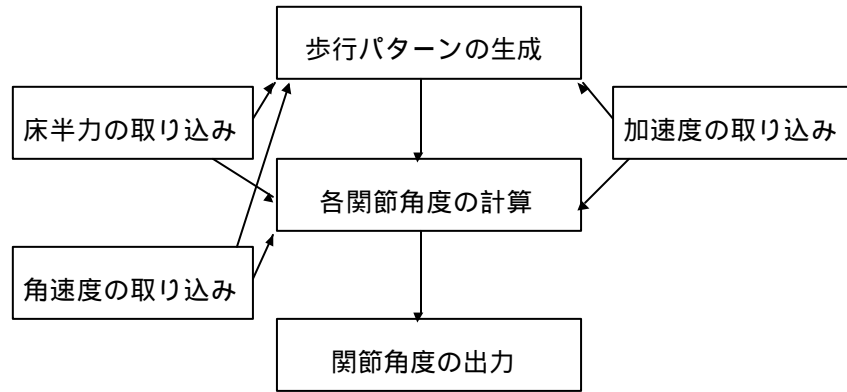
各関節の計算

```
void r_xz(int x,int z){
    float a,th0,th1,tha,thb,thc;
    a=(2*L1-z)/(2*L1);
    th1=acos(a);
    a=x/(2*L1-z);
    th0=atan(a);
    tha=th0+th1;
    thb=2*th1;
    thc=th1-th0;
    a+=tha*(KKK+150)+rln[2];
    c_rl[2]=mmchk(a);
    a+=thb*KKK+rln[3];
    c_rl[3]=mmchk(a);
    a=-thc*(KKK+100)+rln[4];
    c_rl[4]=mmchk(a);
}
```

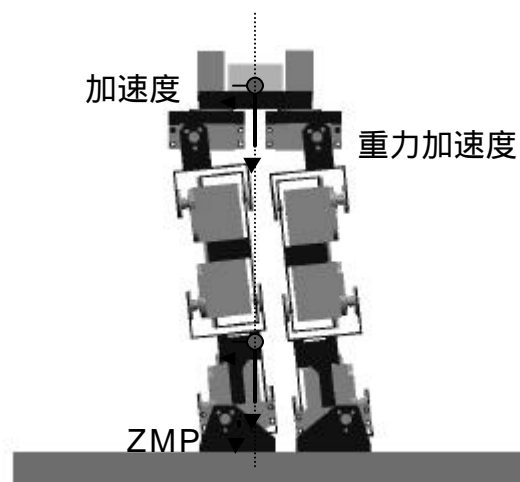
関節角度の出力

ROBO-ONE Technical Conference

## 歩行アルゴリズム



## 動歩行へのステップ



## 静歩行のポイント

---

静歩行の場合の歩行パターン生成方法について述べたが以下の点に注意してロボット製作が必要である。

- 精度の良いサーボ
- 精度の良い加工
- 低重心の構造(静歩行のみ)
- 加速度の低減
- 振動の防止