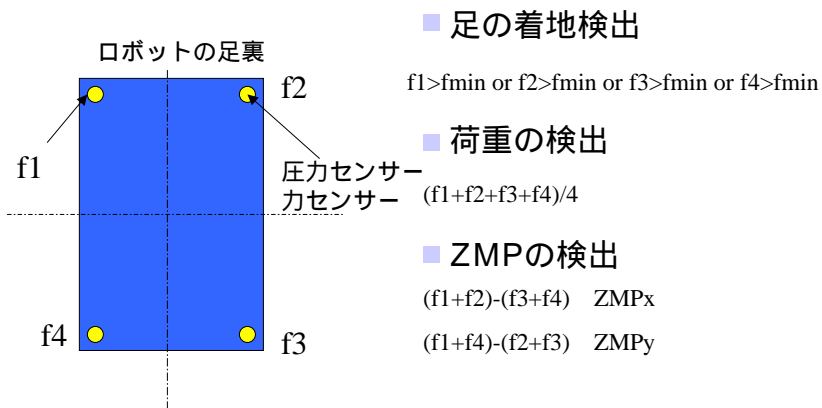


足裏感圧センサー

NRC
西村輝一

足裏感圧センサーの目的



センサーのいろいろ

イナバゴム株式会社



標準製品
 SR-12 …… 外形寸法 11 × 10mm
 SR-12-L …… リード線付き
 SR-12-LT …… リード線、コネクタ付き



Hokuriku Electric Industry

SMC



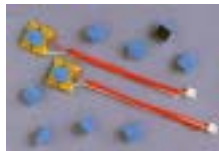
オムロン株式会社



・外形寸法 : 8 × 8 × 2.9 mm
 ・検出荷重 : 0 ~ 10 N
 ・耐荷重 : 20 N

感圧センサーの原理

イナバゴム株式会社



標準製品
 SP-12 …… 外形寸法 11 × 10mm
 オーダー品
 SP-12-L …… リード線付き
 SP-12-LT …… リード線、コネクタ付き

電気的特性：
 最大許容電圧 3.0 (V)
 使用推奨電圧 3 ~ 6 (V)
 最大許容電流 2.0 (mA)
 使用推奨電流 5 (mA)
 無負荷抵抗値 500 K (オーム) 以上
 最大負荷抵抗値 500 (オーム)
 最大許容負荷 4.0 (kgf)

<http://www.inaba-rubber.co.jp/katarogu/inast/inast.html>

イナストマーは、絶縁性のゴム材料と、導電性粒子の2つの材料から成る。絶縁性のゴム材料中に導電性粒子がほぼ均等に分散された状態で成形されたものである。そして、無加圧時は導電性粒子は互いに接触しておらず（導電経路を形成していない状態）体積抵抗、表面抵抗ともに、10の7乗以上の非常に高い電機抵抗値を示している。これに加圧すると導電粒子が次第に接触し始め、導電経路が形成され、3次的に導電経路が増えてF-R特性は滑らかに変化していくことになる。

感圧センサーの取り付け

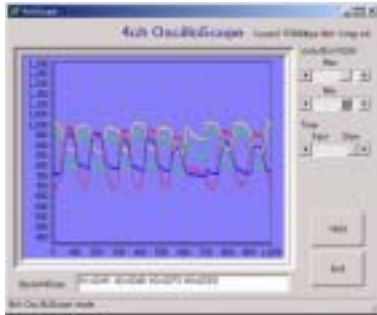
試験回路



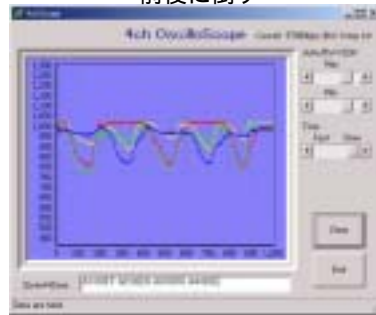
NR1の足裏



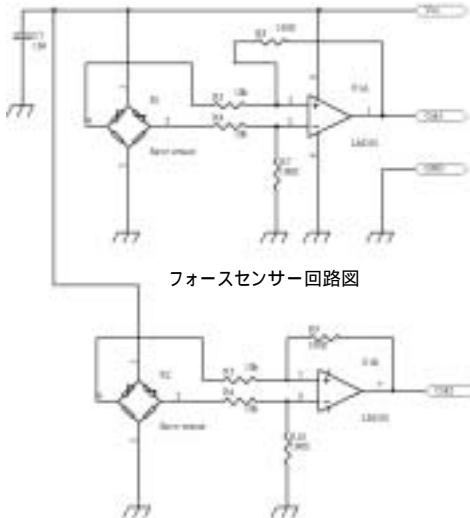
左右に倒す



前後に倒す



フォースセンサーボード



フォースセンサー回路図

Hokuriku Electric Industry

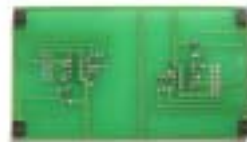
Max 5N



フォースセンサー



センサーボード部品面



センサーボードセンサー面

AD変換

SH-7045のAD変換の例

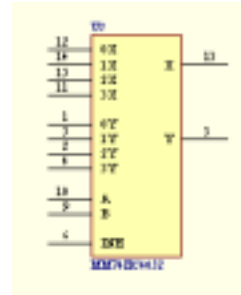
端子名	機能	入出力	備 考
アナログ電源	V _{CC}	入力	アナログ動作電源
アナロググランド	V _{SS}	入力	アナログ動作のグランドおよびAD変換の基準電圧
シフトレジスタの電源	V _{DD}	入力	AD変換用の基準電圧 (0V, 0.5V, 1.0V, 1.5V, 2.0V, 2.5V)
AD0	アナログ入力0	入力	アナログ入力チャンネル0
	アナログ入力1	入力	アナログ入力チャンネル1
	アナログ入力2	入力	アナログ入力チャンネル2
	アナログ入力3	入力	アナログ入力チャンネル3
AD01	アナログ入力4	入力	アナログ入力チャンネル4
	アナログ入力5	入力	アナログ入力チャンネル5
	アナログ入力6	入力	アナログ入力チャンネル6
	アナログ入力7	入力	アナログ入力チャンネル7
AD変換トリガ入力	ADTRIG	入力	AD変換開始トリガの外部信号入力

ADの初期化

```
void ad_int(void)
{
    AD_CSR0.BYTE = 0x13; // 4CHのスキャンモード
    AD_CSR0.BIT.ADST = 1; // AD0変換開始!
    AD_CSR1.BYTE = 0x13; // 4CHのスキャンモード
    AD_CSR1.BIT.ADST = 1; // AD1変換開始!
}
```

マルチプレクサの活用

AD入力が少ない場合はマルチプレクサを活用する。

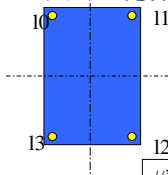


ADのデータ取り込み

```
int get_ad0(void)
{
    return (int)(0x3ff & (AD_DRA0.WORD >> 6));
}
```

着地の検出

ロボットの足裏



足裏センサーにより

- 足が離れているか?
- 足の何処かが接地しているか?
- 全てが接地しているか?
- など様々な情報が得られる。

//着地チェック

```
if (I0>50||I1>50||I2>50||I3>50){l_on=1;}else{l_on=0;}
if (r0>50||r1>50||r2>50||r3>50){r_on=1;}else{r_on=0;}
//体重ののった事をチェック
if (I0>250&&I1>250&&I2>250&&I3>250){l_weight=1;}else{l_weight=0;}
if (r0>250&&r1>250&&r2>250&&r3>250){r_weight=1;}else{r_weight=0;}
//全てが接地しているかどうか
if (I0>50&&I1>50&&I2>50&&I3>50){l_w_on=1;}else{l_w_on=0;}
if (r0>50&&r1>50&&r2>50&&r3>50){r_w_on=1;}else{r_w_on=0;}
```

ZMPの検出

ロボットの足裏

```
//ZMPの計算
lx_zmp_t=l0-l1+l3-l2;
ly_zmp_t=l0+l1-l2-l3;
rx_zmp_t=r0-r1+r3-r2;
ry_zmp_t=r0+r1-r2-r3;
//平均化処理
k=10; //平均回数
lx_zmp=(lx_zmp*(k-1)+lx_zmp_t)/k;
ly_zmp=(ly_zmp*(k-1)+ly_zmp_t)/k;
rx_zmp=(rx_zmp*(k-1)+rx_zmp_t)/k;
ry_zmp=(ry_zmp*(k-1)+ry_zmp_t)/k;
```

フィルターを入れたり平均化処理を行うと、ノイズや振動に強い。

制御の方法

制御1: ZMPにより床反力制御を行う。

制御2: さらにZMPが足裏より外れた場合は上体の制御を行う。

PID制御の実際

```

void L_l_cont(void){
    int zmp_x,zmp_y;

    比例制御    微分制御    積分制御
    zmp_x=lx_zmp/20+(lx_zmp-p_lx_zmp)*3;
    zmp_y=ly_zmp/20+(ly_zmp-p_ly_zmp)*3;
    if(zmp_x>1000)zmp_x=1000;
    if(zmp_x<-1000)zmp_x=-1000;    制御量の最大最小を制限しておく。
    if(zmp_y>1000)zmp_y=1000;
    if(zmp_y<-1000)zmp_y=-1000;
    Leg.L.servo[5] = Leg_c.L.servo[5]+zmp_x/5;    足首の軸に出力
    Leg.L.servo[4] = Leg_c.L.servo[4]-zmp_y/5;
    p_lx_zmp=lx_zmp;    一つ前のデータを保存しておく。
    p_ly_zmp=ly_zmp;
}
    
```

トルクセンサー



試作中

軸トルクの検出により、ZMPを求める。

足裏センサーに比較して足裏が浮いた場合も検出が可能。

さらにトルク制御が可能。

足裏センサー制御のポイント

- 足裏センサー制御はロボットの剛性によって大きく影響される。平面に設置された感圧センサーでは少し倒れるとセンサーが浮き、制御が困難で、柔軟性が必要。
- 足首だけの制御はあまり意味はなく、上体とともに制御を行う必要がある。
- ZMPのみではなく足裏から得られる様々な情報を活用すると良い。