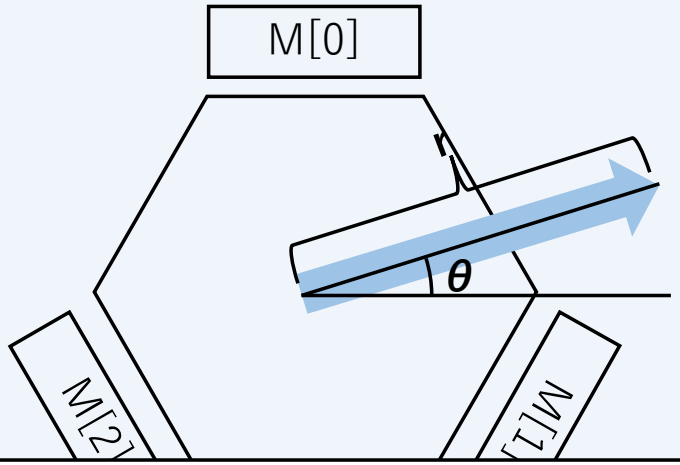
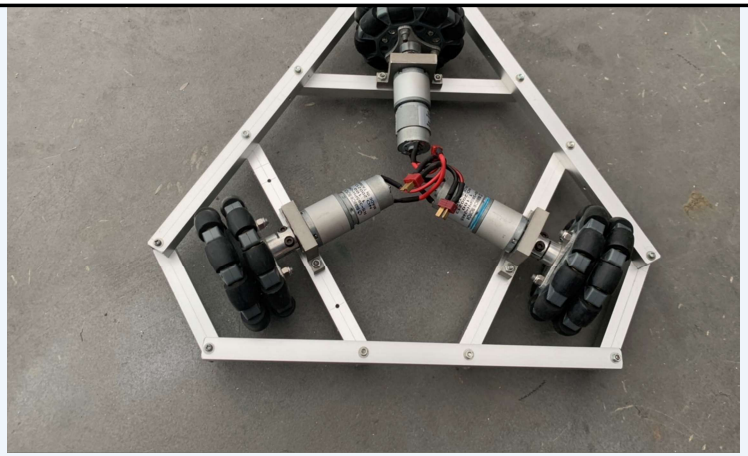


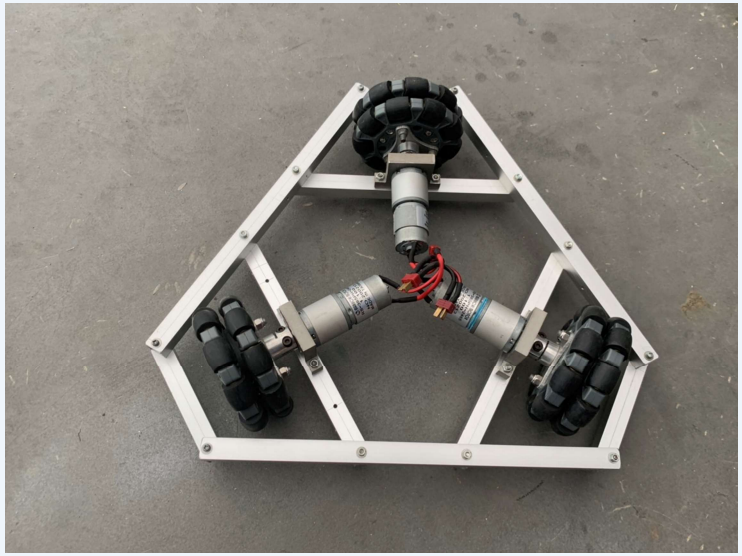
沖縄高専　ロボット製作委員会
制御部門



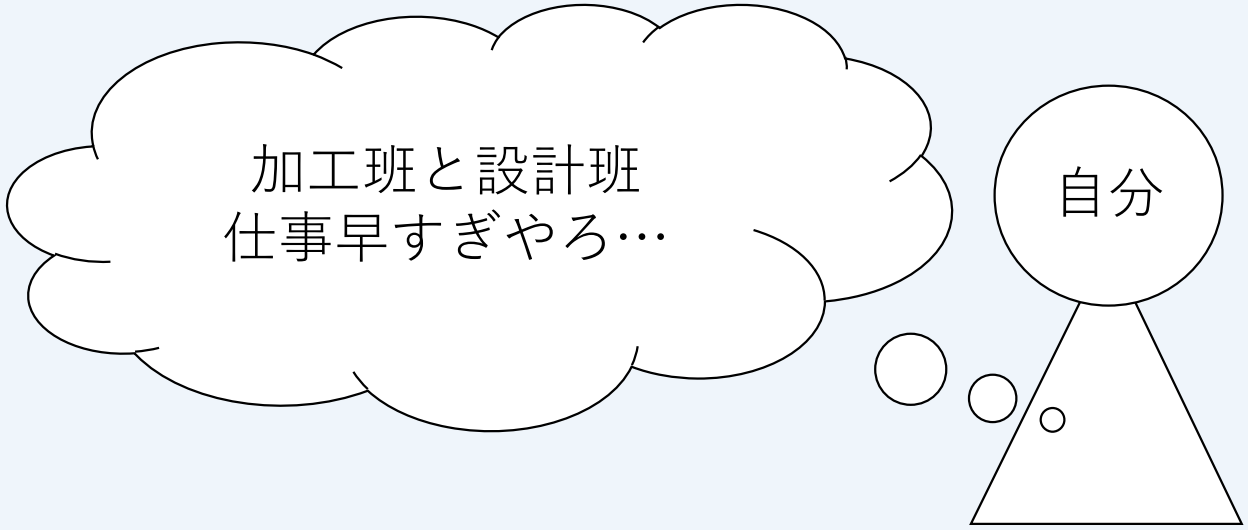
- とりあえず作るぜ！編
- 地獄の調整編

➢とりあえず作るぜ！編

- とりあえず作るぜ！編



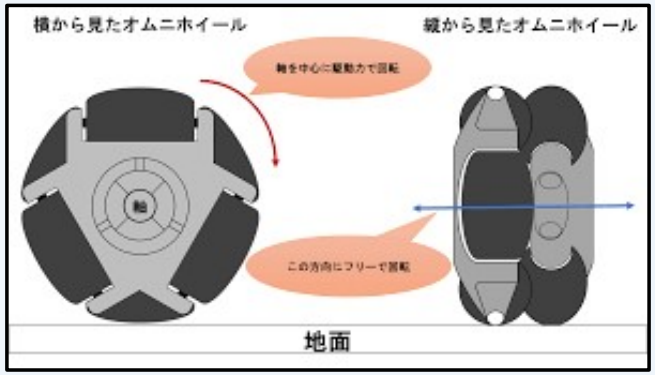
3輪オムニの足回りが
思ったよりも早く
完成してしまいました



➢　とりあえず作るぜ！編

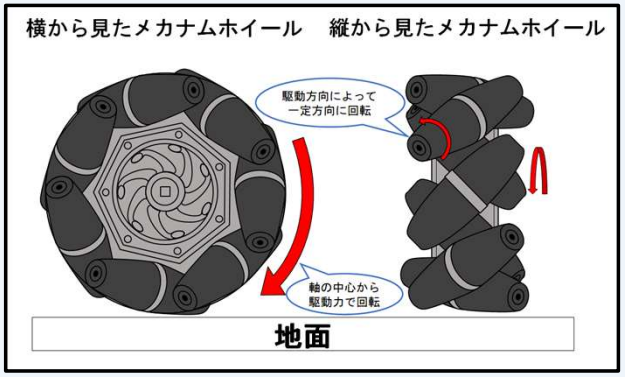
ロボコン豆知識　～足回り～

ロボコンで使われる足回りは主に3種類



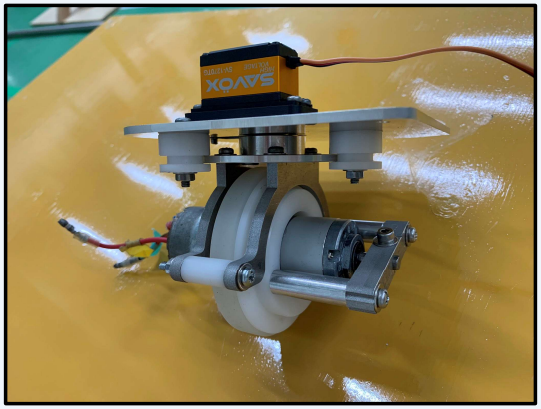
オムニホイール

タイヤに垂直な方向に転がるようにバレルというものがついているので、全方向に移動することができる。



メカナムホイール

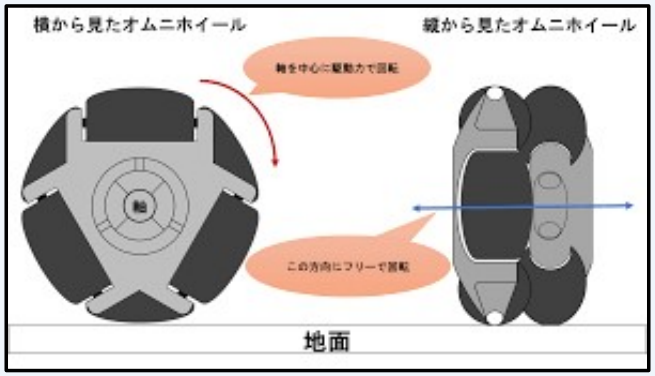
タイヤに対して斜めの方向に転がるようにバレルというものがついているので、全方向に移動することができる。



ステアリング

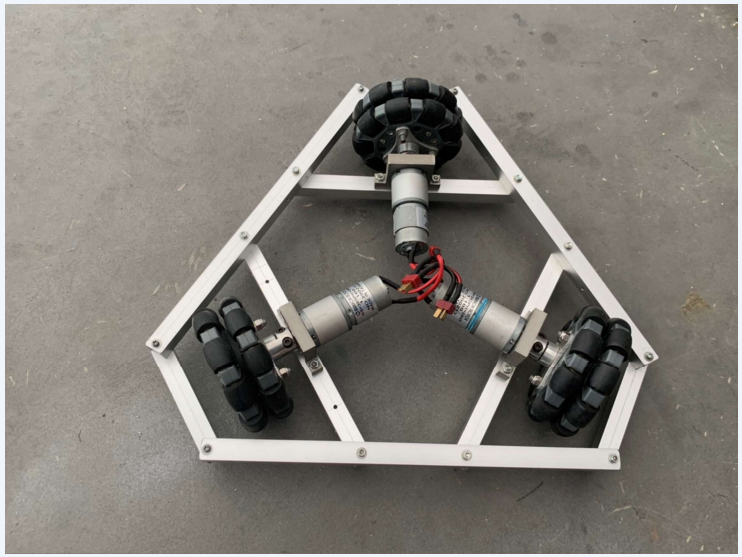
タイヤは普通のタイヤだが、タイヤの向き自体を変えることで、全方向に移動することができる。車のタイヤはこれ。

➢ とりあえず作るぜ！編



オムニホイール

タイヤに垂直な方向に転がるようにバレルというものがついているので、全方向に移動することができる。



「3輪オムニ」というと、オムニホイールが写真のように3つ取り付けられてる状態です

- とりあえず作るぜ！編

```
void Move(float sheta, float r)
{
   //3輪足回りを制御するよ
   float M[3];
   int i;

   M[0] = cos(sheta) * r;
   M[1] = cos(sheta + 4.189) * r;
   M[2] = cos(sheta + 2.094) * r;

   for(i = 0; i < 3; i++){
      if(M[i] < 0){
         MD[i] = 0;
      }
      else{
         MD[i] = 1;
      }
      MP[i] = abs(M[i]);
   }
}
```

とりあえず3輪オムニの
足回りを制御しました

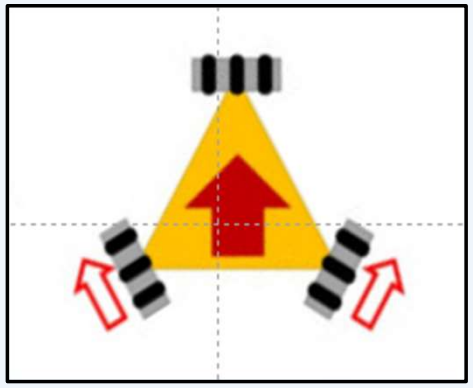
まって、まだ読み飛ばさないで

なんか急にプログラム出されてビビると思うけど、もうちょっと頑張って

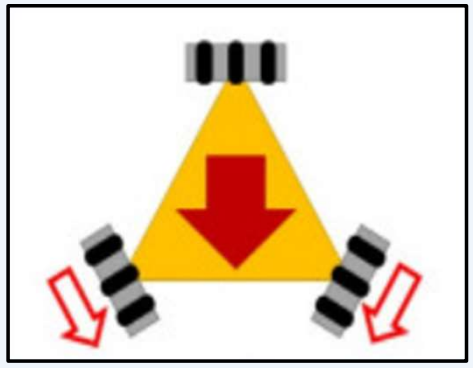
➢　とりあえず作るぜ！編

順番に説明していくと、

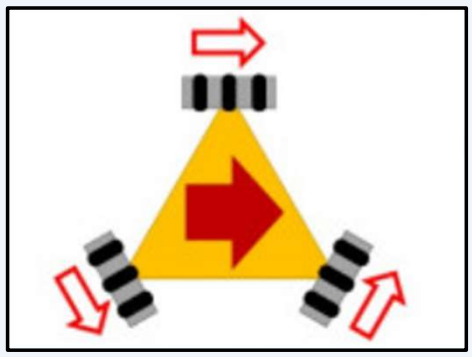
まず前に行きたいときはこう



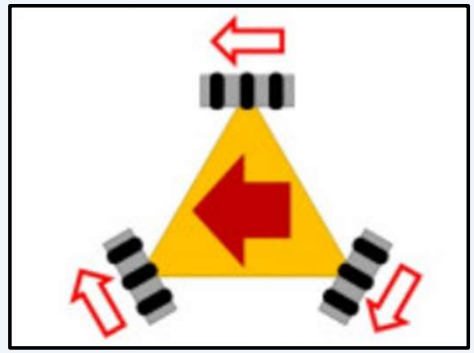
後ろはこう



右



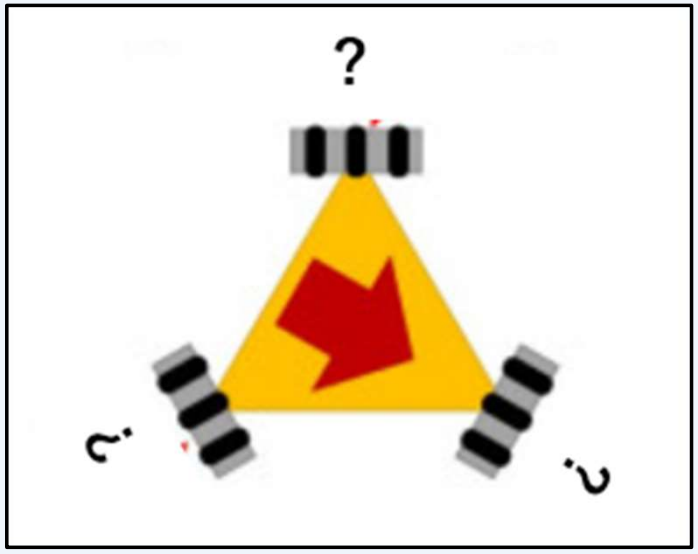
左



こう動かしますよね

- とりあえず作るぜ！編

じゃあ、斜めに行きたいときは？
ってなりますよね



- とりあえず作るぜ！編

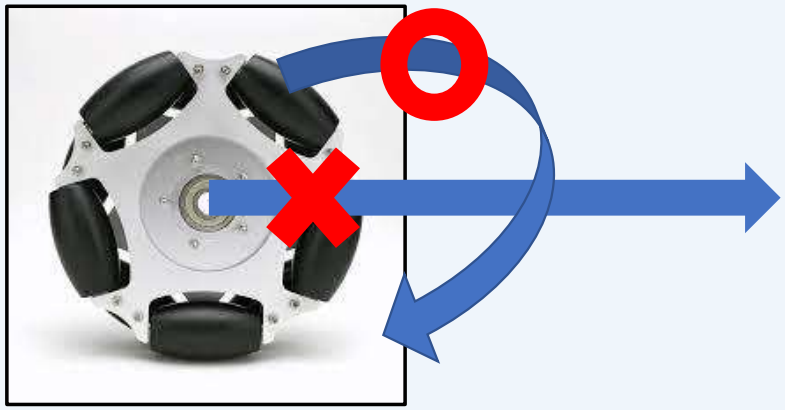
```
void Move(float sheta, float r)
{
    //3輪足回りを制御するよ
    float M[3];
    int i;

    M[0] = cos(sheta) * r;
    M[1] = cos(sheta + 4.189) * r;
    M[2] = cos(sheta + 2.094) * r;

    for(i = 0; i < 3; i++){
        if(M[i] < 0){
            MD[i] = 0;
        }
        else{
            MD[i] = 1;
        }
        MP[i] = abs(M[i]);
    }
}
```

それで、360度どこでも行けるようにしたのがこのプログラムなんです

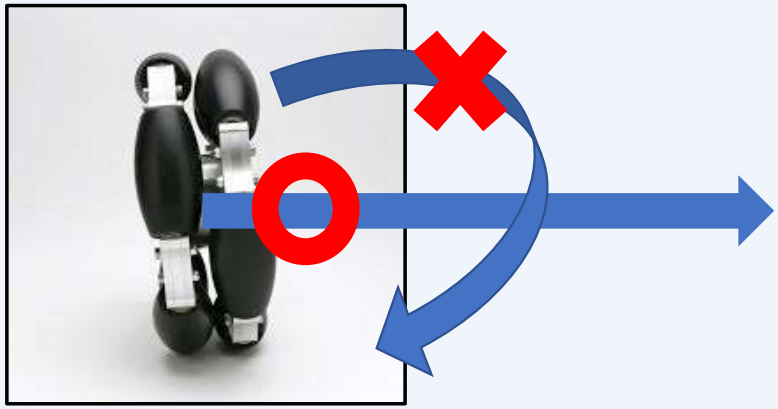
次のスライドから図で説明します

sinとかcosとか出てくるので、「sin…cos…うっ頭が」って人は読み飛ばしてください

➢　とりあえず作るぜ！編

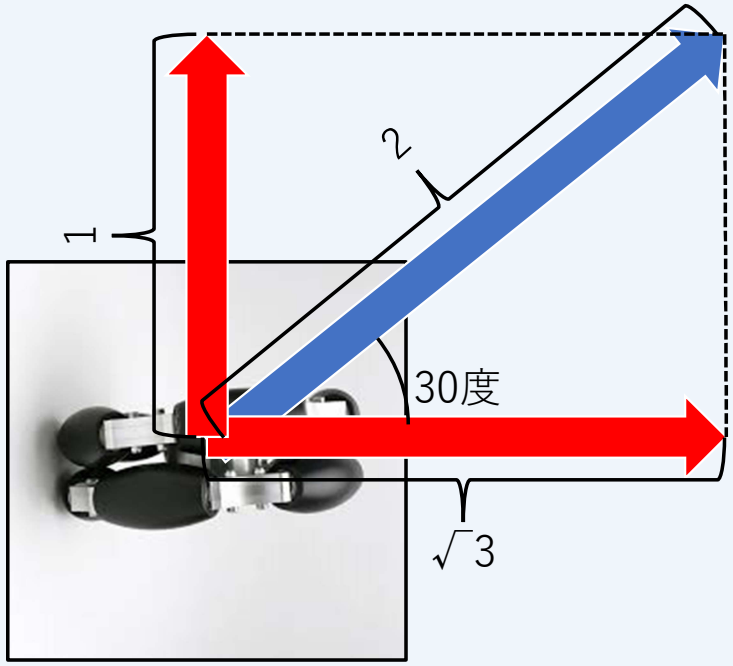


まず、オムニホイールは、縦方向には自らが回転した分しか進めません。引きずっても動かせません。



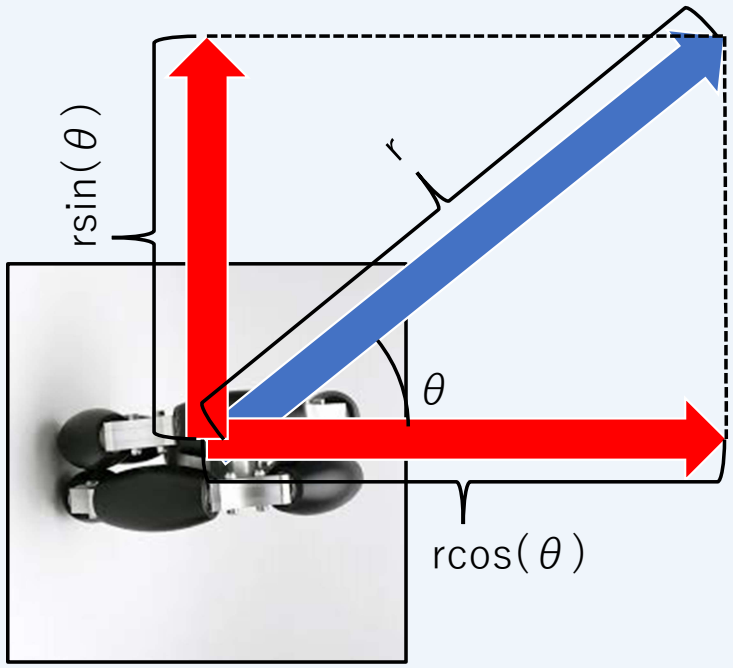
逆に、オムニホイールは、横方向には引きずった分しか進みません。どれだけ回転しようが横には進めないからです。引きずったら動かせるのは、バレルが横方向についているおかげですね。

➢　とりあえず作るぜ！編



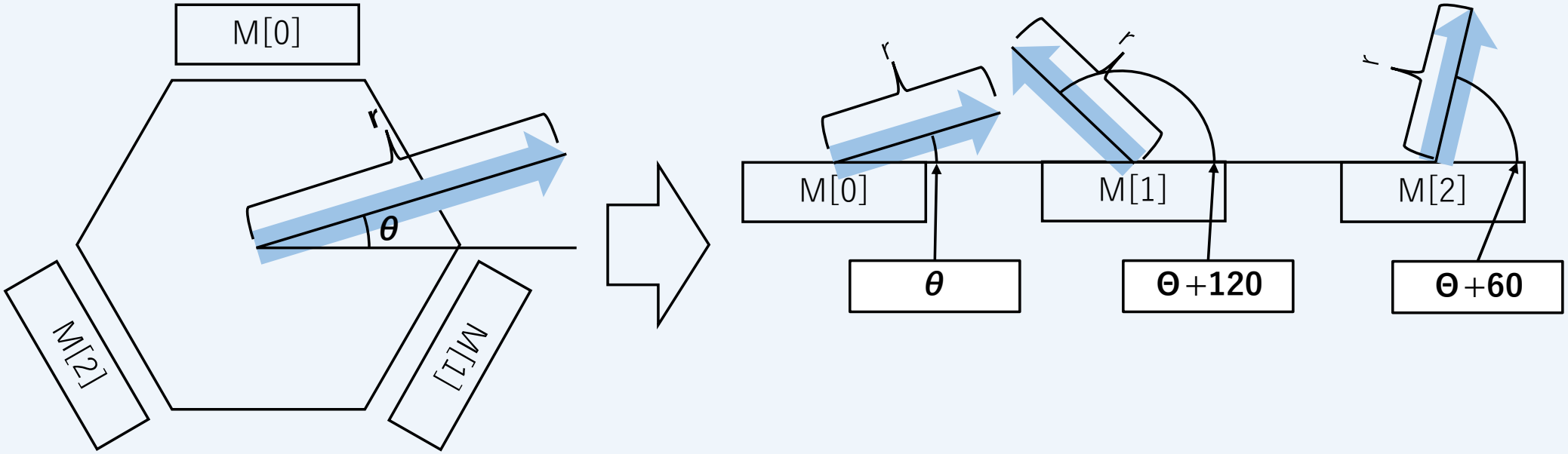
例えば、30度の方向に２のスピードで動きたいときは、縦方向に√3のスピードで回転しながら、横方向に1のスピードで引きずられればいいわけです。

➢　とりあえず作るぜ！編



つまり、θ度の方向にｒのスピードで動きたいときは、縦方向にrcos(θ)のスピードで回転しながら、横方向にrsin(θ)のスピードで引きずられればいいわけですね。

- とりあえず作るぜ！編



ロボットの移動する角度と速さから、各オムニホイールが移動する角度と速さを求めて、

- とりあえず作るぜ！編

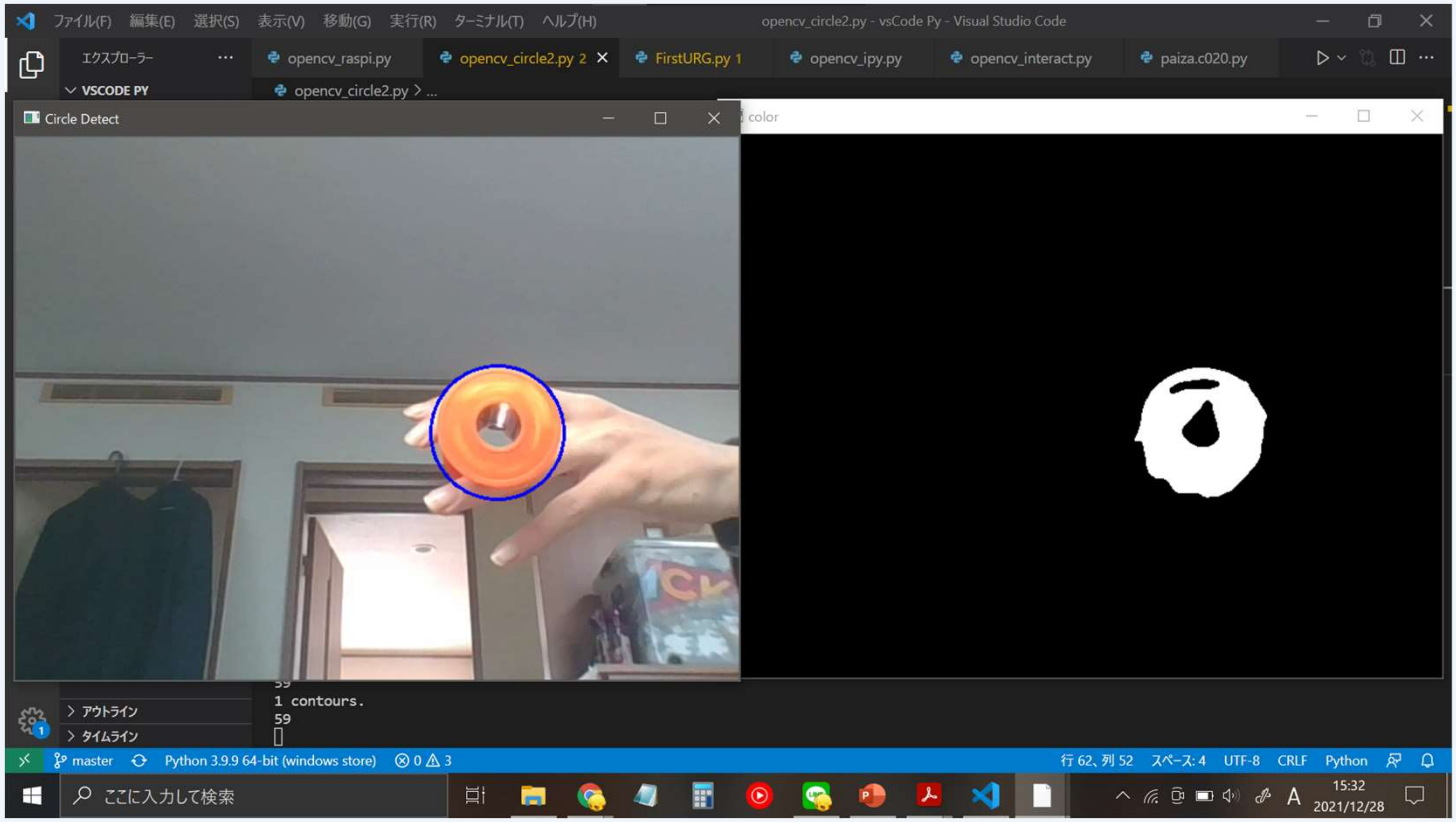
```
void Move(float sheta, float r)
{
   //3輪足回りを制御するよ
   float M[3];
   int i;

   M[0] = cos(sheta) * r;
   M[1] = cos(sheta + 4.189) * r;
   M[2] = cos(sheta + 2.094) * r;

   for(i = 0; i < 3; i++){
      if(M[i] < 0){
         MD[i] = 0;
      }
      else{
         MD[i] = 1;
      }
      MP[i] = abs(M[i]);
   }
}
```

縦方向にどれくらいのスピードで回転すればいいのかを３つのオムニそれぞれで求めると、こうなります
4.189とか2.094とかは、120度とか60度を弧度法に直したやつですね

この3行が大事です。後はまあ無視してください

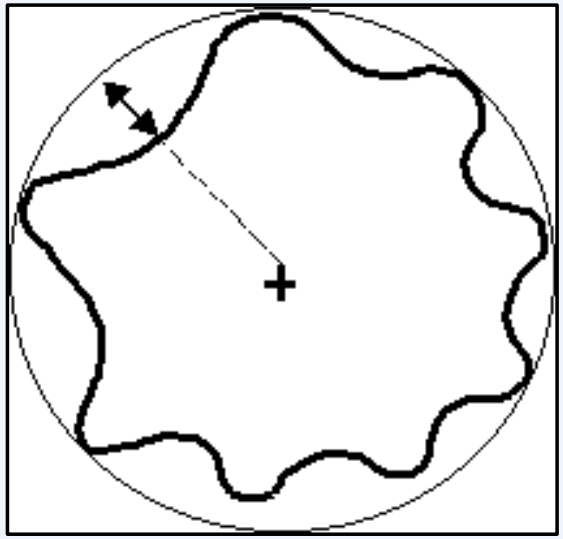
- とりあえず作るぜ！編

足回りが制御できたので、画像処理のプログラムを書きました。
実は画像処理初挑戦



- とりあえず作るぜ！編

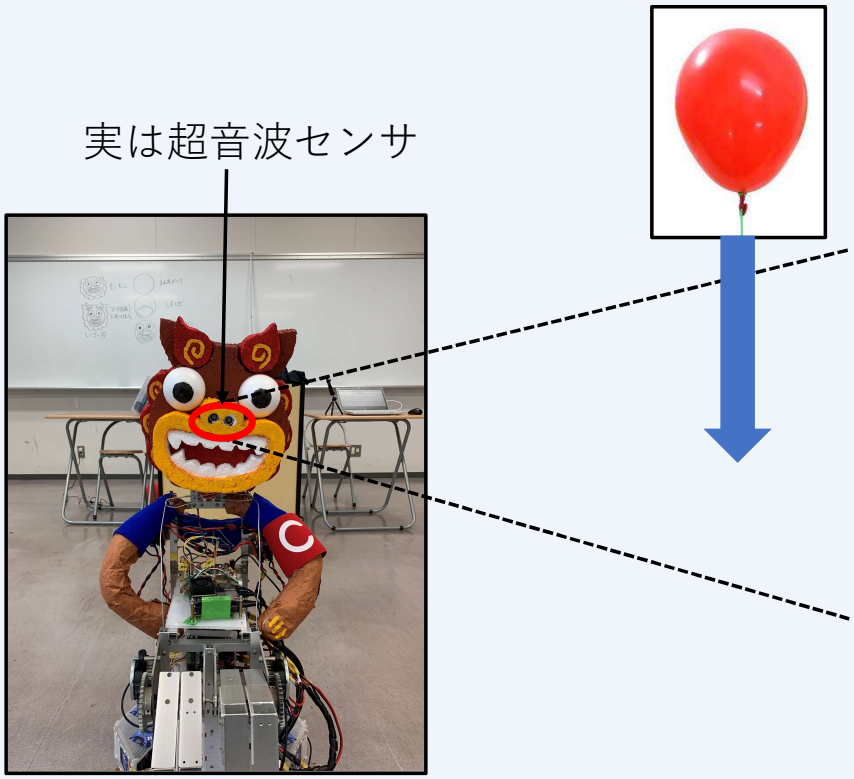
画像処理には、「最小外接円」というアルゴリズムを使いました。今流行りの、AIとかは使っていません(;‘∀’)。コードは以下のサイトをコピp…勉強させていただきました。
https://blog.ashija.net/2017/11/28/post-2549/

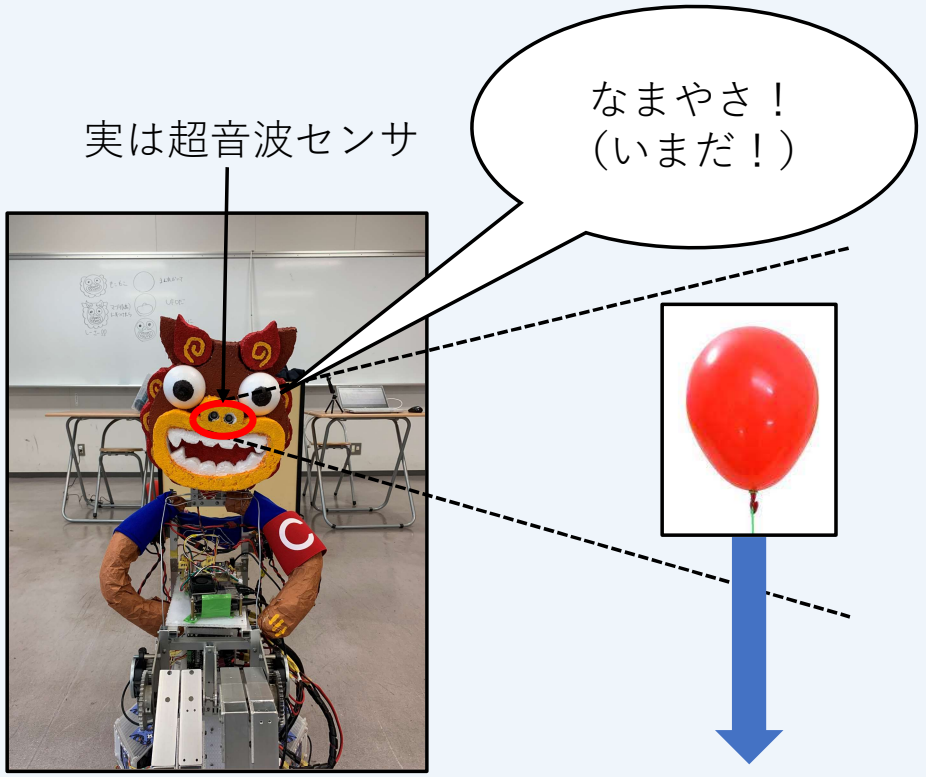


最小外接円とは、ある図形があったら、その図形を囲める一番小さい円を見つけるプログラムのことです。

➢　とりあえず作るぜ！編

最後に、ボールをけるプログラムを書きました。
シンプルに超音波センサで検知しています。





➢地獄の調整編

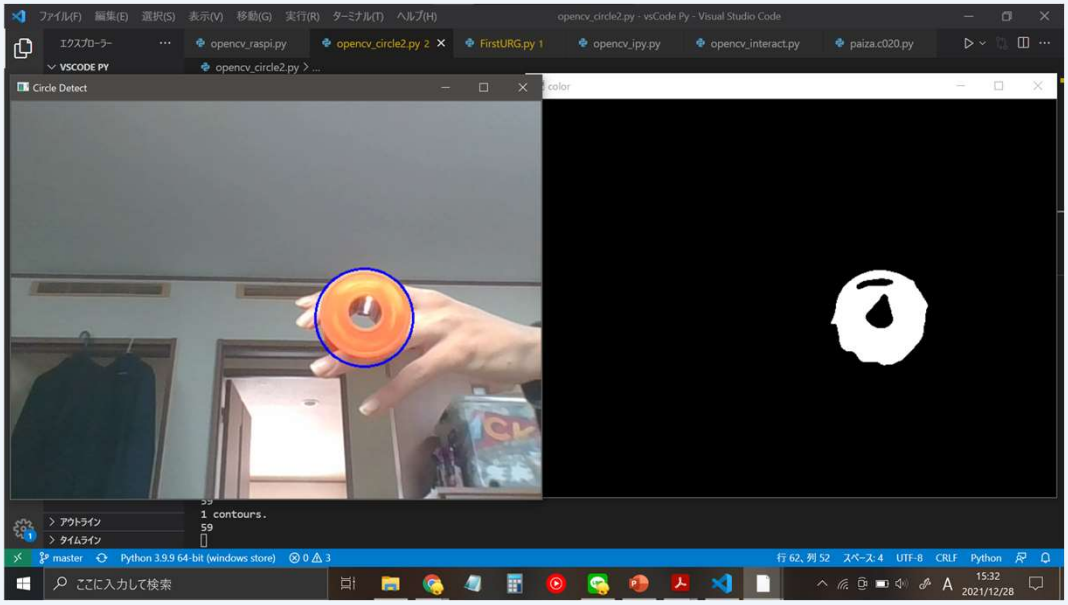
➢ 地獄の調整編

とりあえず一通りのプログラムを書き終えたので、すべてのプログラムを一緒に動かし、パラメータの調整などを行いました。

全部一緒にするとこんな感じ

➢ 地獄の調整編

一番悩まされたのは画像処理の速さですね。最初は1秒に7回ほどしか画像処理ができなかったです。1秒に7回というと早く感じるかもしれませんが、それは逆にいうと1/7秒のラグが生じるということ。リフティングをするには遅すぎるんです。



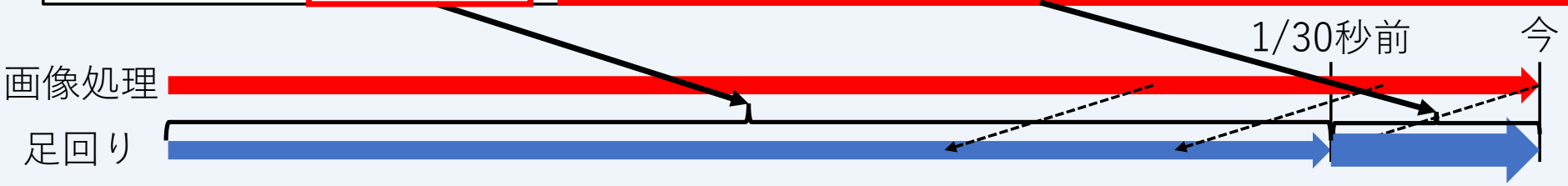
画素を落とすと1秒に30回まで改善しました。

- 地獄の調整編

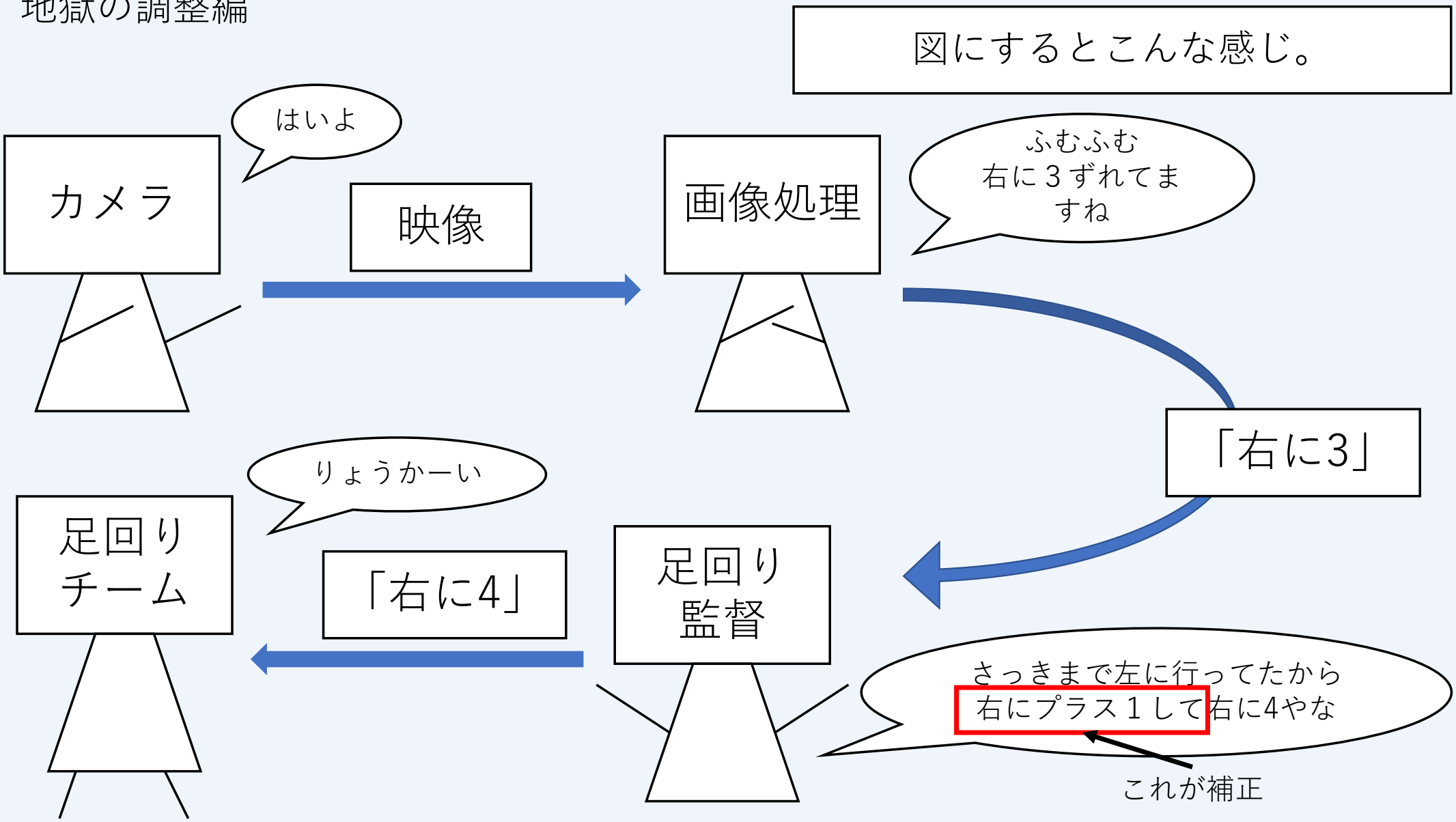
しかし、1/30秒のラグでもまだ遅いんですね。それ以上早くするのはあきらめて、ラグを補うプログラムを書くことにしました。

ロボットがどこに、どれくらいの速さで移動しているかを常に記録しておいて、直近1/30秒分のデータを足し合わせます。それをもとに画像処理の結果に補正をかけることで、ラグを補えます。

```
//ボールの座標にロボットの移動分補正をかけるよ
    move[1] = (float)ball[1] - ((integ_x[0] + integ_x[1] + integ_x[2] + integ_x[3] + integ_x[4]) * 10);
    move[2] = (float)ball[2] + ((integ_y[0] + integ_y[1] + integ_y[2] + integ_y[3] + integ_y[4]) * 10);
```



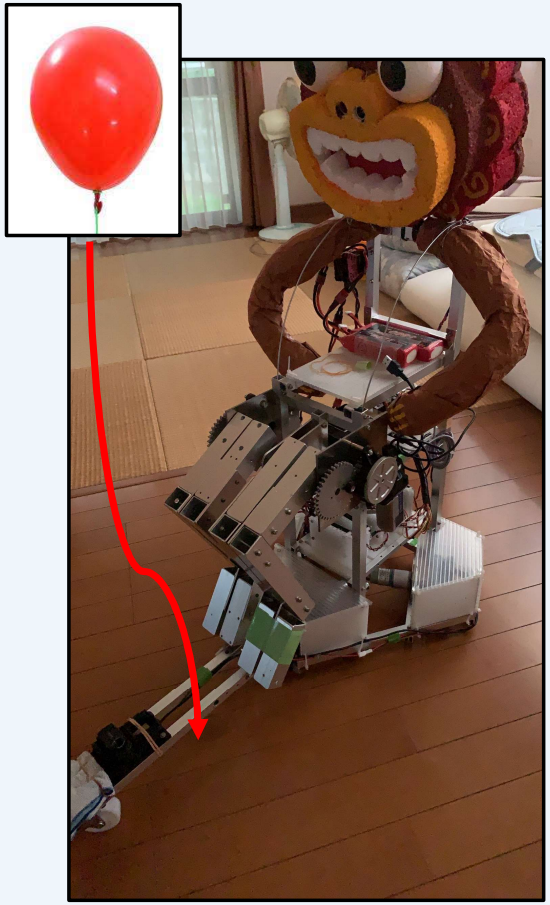
➢ 地獄の調整編


図にするとこんな感じ。
カメラ
はいよ
映像
画像処理
ふむふむ
右に３ずれてま
すね
「右に3」
足回り
チーム
りょうかーい
「右に4」
足回り
監督
さっきまで左に行ってたから
右にプラス１して右に4やな
これが補正

- 地獄の調整編

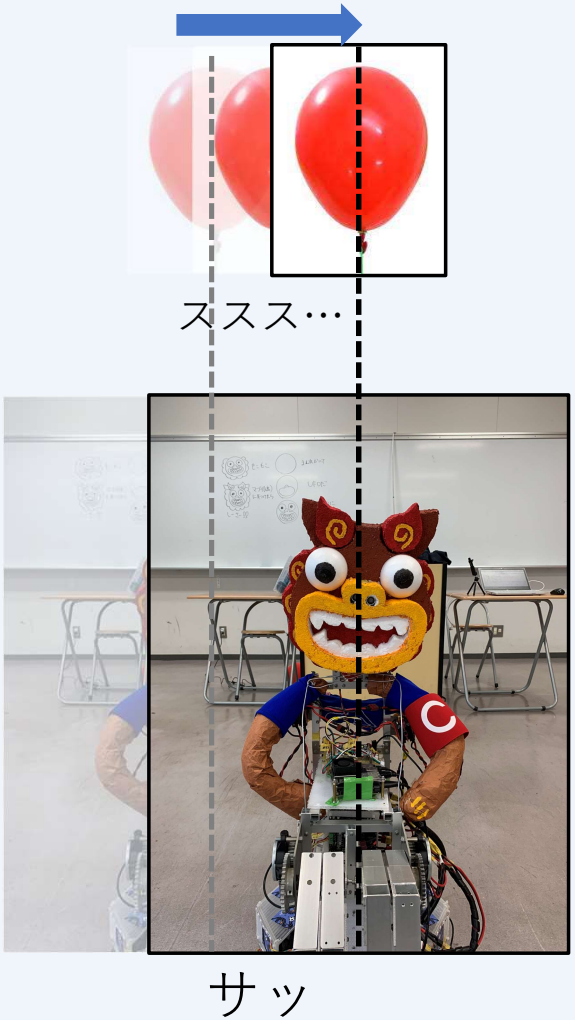
ラグ補正のおかげで何とかそれっぽくなりました。
しかし、４回より伸びない…
↓初4回成功の動画です

➢ 地獄の調整編



リフティングが安定しない原因は、
風船の落ち方だと考えました。
こいつ、ちょうど足元でゆらゆらと
動きやがるんですよね

➢　地獄の調整編



最初に考えてたのは、風船がゆらゆら動いてもそれに対応できるくらいロボットの動きを早くすることでした。

- 地獄の調整編

無理でした。

➢ 地獄の調整編

高く蹴り上げる

低く蹴り上げる

だけどね、気づいちゃうんです。
そもそも風船がゆらゆらしなけりゃ問題ないんです。

具体的には、風船をあえて低く蹴り上げることで、「ゆらゆらする前にける」

あ、この時点で大会一週間前くらいですね

➢ 地獄の調整編

2021/10/17　高専ロボコン　九州沖縄地区大会

- 地獄の調整編

なんと、全国に行けてしまいました
実は、「ゆらゆらする前にける」のが成功して、
最高20回まで記録を伸ばすのに成功してたんです

全国決まっても気は抜けません
むしろ気を引き締めてさらに調整を重ねます

- 地獄の調整編

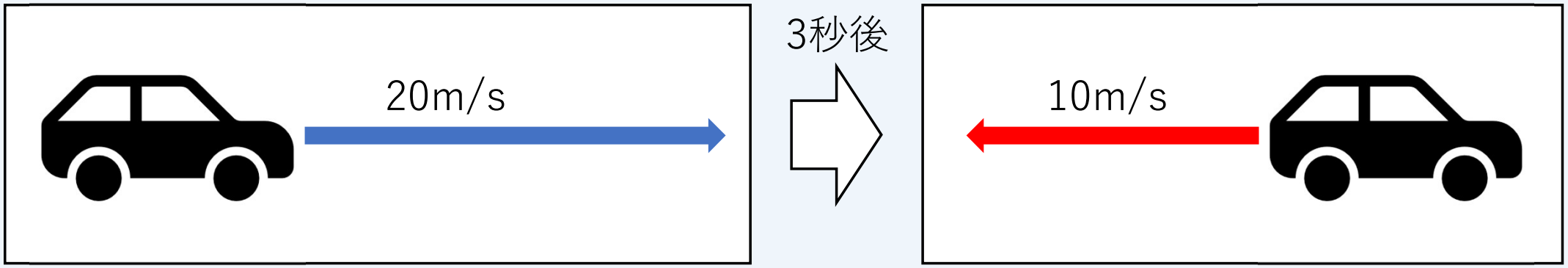
まず、ロボットが傾いたりしないように、
加速度を制御するプログラムを追加しました

```
//加速度制御するよ
//左に傾かないように～
accel = (kyori * cos(kakudo + 0.524)) - (speed * cos(theta + 0.524));
if(accel > maxaccel){
    kyori = ((speed * cos(theta + 0.524)) + maxaccel) / cos(kakudo + 0.524);
}
//右に傾かないように～
accel = (kyori * cos(kakudo + 2.618)) - (speed * cos(theta + 2.618));
if(accel > maxaccel){
    kyori = ((speed * cos(theta + 2.618)) + maxaccel) / cos(kakudo + 2.618);
}
```

➢ 地獄の調整編

加速度というのは、「速度の速度」のことです。

たとえば、20m/sで前に進んでいた車が、3秒後には10m/sで後ろに進んでいたとします。この場合、速度は20m/sから-10m/sに、3秒で30m/s減っています。この時の加速度は、-10m/s^2になります。



- 地獄の調整編

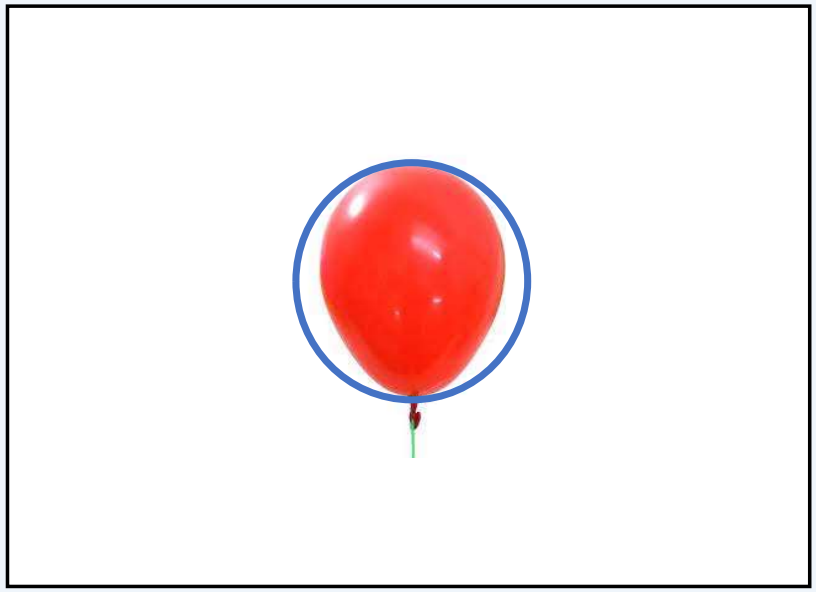
加速度が大きくなると、ロボットがこけます。
なので、加速度が一定以上にならないようにプログラムします。
このプログラムは少し難しいので説明を省きます。

```
//加速度制御するよ
//左に傾かないように～
accel = (kyori * cos(kakudo + 0.524)) - (speed * cos(theta + 0.524));
if(accel > maxaccel){
    kyori = ((speed * cos(theta + 0.524)) + maxaccel) / cos(kakudo + 0.524);
}
//右に傾かないように～
accel = (kyori * cos(kakudo + 2.618)) - (speed * cos(theta + 2.618));
if(accel > maxaccel){
    kyori = ((speed * cos(theta + 2.618)) + maxaccel) / cos(kakudo + 2.618);
}
```

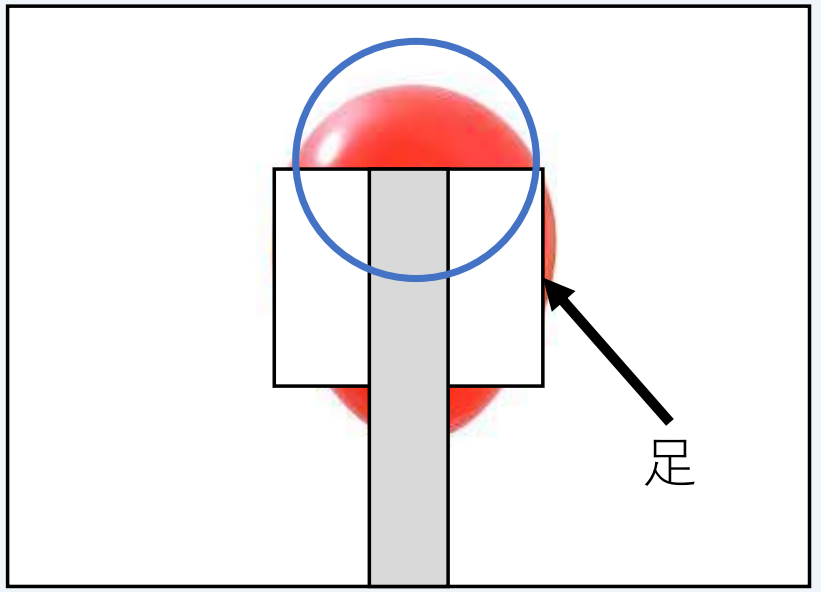
➢ 地獄の調整編

そして、調整している途中に気づいたんですが、どうも風船をけってるときのロボットの動作が安定しないんですよね。
風船をけってるときは足で風船が隠れて、画像処理が乱れていたようです。

けってないとき



けってるとき

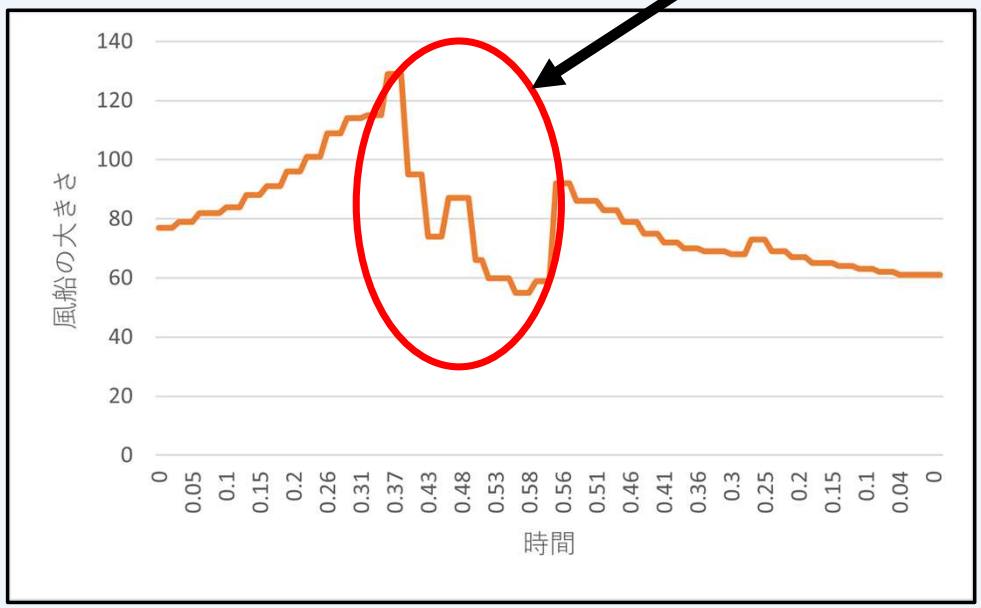


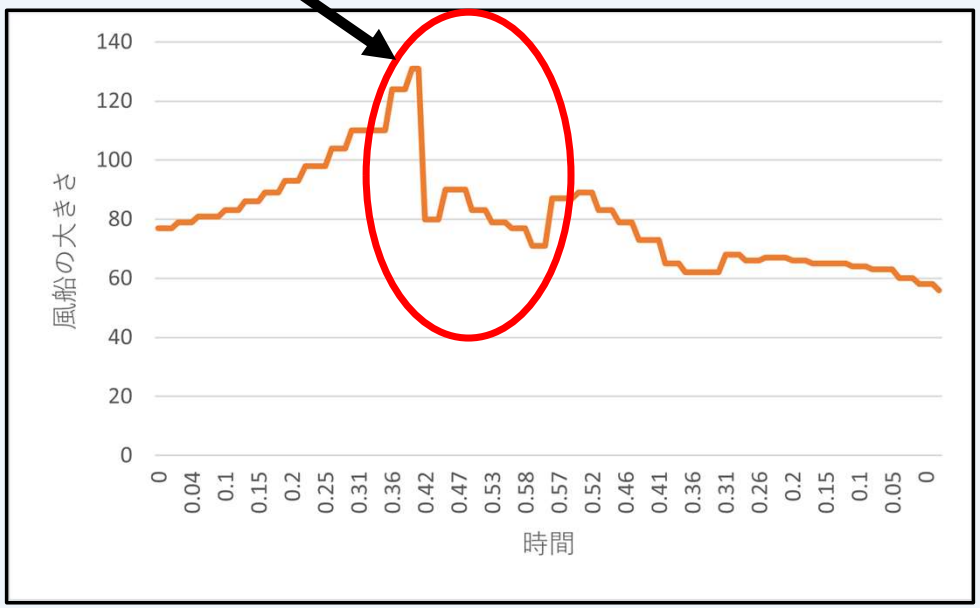
<< 足元からの視点 >>

➢ 地獄の調整編

風船をける前後の風船の大きさのデータをグラフ化して、大体どのタイミングで風船が足に隠れるのかを見ました。

おそらくここらへん





- 地獄の調整編

タイミングが分かったので、あとはそのタイミングでゆっくり移動するようにします。

```
//足でボールが隠れて処理が乱れるからゆっくり移動するよ
legtimer = t_up.read() - t_down.read();
if(legtimer < 0){
    legtimer = 0;
}
if((t_up.read() >= 0.3 && t_up.read() <= 0.35) || (legtimer > 0.3)){
    kyori = kyori * 0.4;
}
```

この3行が大事なので
後は無視していいです

日本語で書くと、
「もし今『蹴ってる』なら、
動く速さを40%にする」
です

- 地獄の調整編

2021/11/28　高専ロボコン　全国大会

国技館ではうまく
調整がいかない中で
最高13回のリフティング

- 地獄の調整編

安川電機特別賞をいただきました
最善を尽くせたかはわからないけど、
ただ一つ言えるのは

まじで楽しかった！

ここまでお付き合いいただきありがとうございました。
これを見てロボコンやりたくなったら、いつでも沖縄高専で待ってます！
ロボコンやったことない人でも大丈夫
（何なら自分も初心者でした）