

ロボット設計製作論1

ロボットの速度制御実験

林原、王、藤井

ロボットの速度制御実験

実験 1 : 渡されたサンプルプログラムでロボットの最高速度を計測していく

同じ条件で5回以上の実験を行い、速度を求めるための距離・時間データを記録し、速度を計算した上、各実験データと共に平均値と最大誤差、もしくは分散をレポートに記入する、以降の実験同様

実験 2 : 1 mの直線を5秒間で走行するように、モータに加えるPWMのデューティ比(0~1.0)を調整してください。

走行データ(時間、距離)を計測し、実際の走行速度を計算

実験 3 : ロボットの速度をエンコーダをカウントし、車輪の速度を測定するプログラムを作成して、車輪が回転時のエンコーダ値を確認しましょう。

左右の車輪が回転した際のエンコーダ値から計算した速度値を確認し、写真かスクリーンショットでイメージをセーブしてレポートに入れる。

「未ロボ実習用制御ガイドスタートガイド」に参照してください。

11月29日までオンラインコンパイルをできるようにしてください

ロボットの速度制御実験

速度制御レポート：提出期限（2021年12月20日 23:50）

内容： （実験目的，実験手法，実験データと考察が必要）

1. ロボットの最高速度の特性を実験で確認（課題1）
2. ロボットの定速走行特性を実験で確認（課題2）
（走行直進性，距離精度）
3. 車輪エンコーダによる回転速度の計測（課題3）
（今回はカウンタ値の変化を写真記録）
4. 速度制御特性に関する実験（課題4）
P制御の効果，I制御の効果の確認（実験4-1，4-2，4-3）
ゲインチューニングによりPI制御での特性の確認（実験4-4）
5. 軌道制御により図形作成実験
（12月12日の軌道制御の回で詳細説明）

ロボットの速度制御実験

実験 1 : 渡されたサンプルプログラムでロボットの最高速度を計測していく
同じ条件で5回以上の実験を行い、速度を求めるための距離・時間
データを記録し、速度を計算した上、各実験データと共に平均値と
最大誤差、もしくは分散をレポートに記入する、以降の実験同様

以下は左のモータを動くサンプルプログラムに、右のモータを駆動するコマンドを
追加して、ロボットの両輪で全力前進をするプログラムを完成してください。

```
#include "mbed.h"
#include "Motor.h "
#include "adrobo.h"

BusOut led(LED1,LED2,LED3,LED4);
Motor motor_left(MOTOR11,MOTOR12);

int main() {
    float i= 1.0f;
    motor_left.setMaxRatio( 1.0 );    /* 初期設定が最大出力は25%か100%に設定*/
    while(1) {
        motor_left = i;
    }
}
```

ロボットの速度制御実験

実験 2 : 1 mの直線を 5 秒間で走行するように、モータに加えるPWMのデューティ比 (0~1.0) を調整してください。

走行データ（時間、距離）を計測し、実際の走行速度を計算

ロボットの全力走行に関して、非常に速いと感じたと思います。図形を書くためには、もっとゆっくり正確に移動することが必要になります。では、ゆっくり動くためにはどうすればいいでしょうか。

既にみなさん知っているように、PWM制御を使用します。

今回は、CHOP (PWM用の端子) が Lの時にモータに電流が流れるので、図の Lの幅を調整します。ロボット電子回路で方法に関しては、学びましたのでここで実機で確認、実験 1 のプログラムで、制御することができます。

`moter_left`, `moter_right` の値を変え、ロボットの走行特性を実験

ロボットの速度制御実験

実験 3 : ロボットの速度をエンコーダをカウントし、車輪の速度を測定するプログラムを作成して、車輪が回転時のエンコーダ値を確認しましょう。

左右の車輪が回転した際のエンコーダ値から計算した速度値を確認し、写真かスクリーンショットでイメージをセーブしてレポートに入れる。

上記の実験 2 で様々な調整を行ったと思いますが、おそらくは電池やコースの路面など、少しの変化で大きく挙動が変化することに気づいたと思います。

これは、回転数を計測せずに適当に電流を流しているからです。環境からの影響を少なくするためには、減速比を大きくすればいいのですが、緩慢な動きしかできなくなってしまいます。

よって、正確の軌道での走行は、速度制御が不可欠となり、エンコーダによる各車輪の速度計測は必須となります。

ロボットの速度制御実験

実験3： ロボットの速度をエンコーダをカウントし、車輪の速度を測定するプログラムを作成して、車輪が回転時のエンコーダ値を確認しましょう。

割込処理

ロボットを制御するためには、多くの場合、複数の処理を同時に行わなくてはなりません。

今回の場合は、「ロータリエンコーダのパルスのカウントしながら」「結果を表示する」「モータに加える信号を変更する」など

同じ周期で行うのであれば、順番に行えば良いのですが、違う場合は割り込み処理を使います。

これは、あるイベントが起きたときに、今の処理を一時終始して、別の処理を行い、終了後にまた再開するというものです。割込処理は速やかに終了して、本来の処理に戻すことが重要です。

さらに、同じ周期で行う処理は、フィードバック制御の安定性の保証、計測した変位から速度などへの計算に大きな意味を持ちます。

ロボットの速度制御実験

実験3： ロボットの速度をエンコーダをカウントし、車輪の速度を測定するプログラムを作成して、車輪が回転時のエンコーダ値を確認しましょう。

左右の車輪が回転した際のエンコーダ値から計算した速度値を確認し、写真かスクリーンショットでイメージをセーブしてレポートに入れる。

```
QEI qei_left(GPIO1, GPIO2, NC, 48, QEI::X4_ENCODING);
QEI qei_right(GPIO3, GPIO4, NC, 48, QEI::X4_ENCODING);

const double sampling_time = 0.020;
const double move_per_pulse = 0.0005;

int main() {
    in.mode(PullUp);
    while(1) {
        int enc_left = qei_left.getPulses();
        int enc_right = qei_right.getPulses();

        qei_left.reset();
        qei_right.reset();
    }
}
```


ロボットの速度制御実験

実験3： ロボットの速度をエンコーダをカウントし、車輪の速度を測定するプログラムを作成して、車輪が回転時のエンコーダ値を確認しましょう。

左右の車輪が回転した際のエンコーダ値から計算した速度値を確認し、写真かスクリーンショットでイメージをセーブしてレポートに入れる。

つづく

```
double speed_left = move_per_pulse * enc_left / sampling_time;
double speed_right = move_per_pulse * enc_right / sampling_time;

printf("%f %f\n", speed_left, speed_right);

speed_left = speed_right = 0.0;

wait(0.1);
}
}
```

ロボットの速度制御実験

4. 速度制御

通常のロボットの速度制御には、PD制御、PI制御とPID制御などがあります。

Pは 制御の比例項 (Proportional Control Part) ,

Dは 制御の項 (Differential Control Part) ,

Iは 制御の積分項 (Integral Control Part)

とそれぞれ意味をして、フィードバック制御には、もっとも基礎的かつ効果的な制御手法となります。

今回利用する制御アルゴリズムは、エンコーダの性能上の理由で、PI制御則に限定しています。制御則は以下の式で表せます。

$$E = V_d - V, \quad \tau = K_p E - K_I \int E$$

ここで、 V_d は速度の目標値、 V は現在計測された速度とし、速度の誤差は E となります。 K_p は比例制御ゲイン、 K_I は積分制御ゲインと呼びます。計算された制御出力 τ をモータへ出力し、その車輪の速度制御を行います。

自動制御の基本手法

最も基本の制御手法として

フィードバック制御 (feedback control)

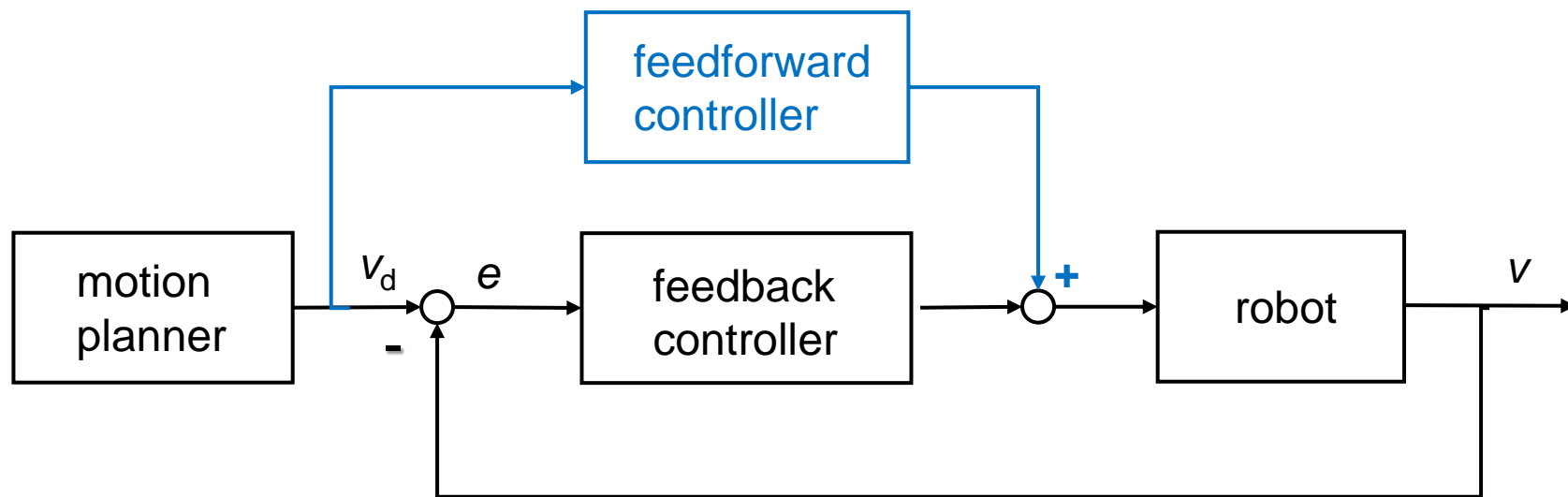
フィードフォワード制御 (feedforward control)
(通常、フィードバック制御と組み合わせる)

自動制御の基本手法

最も基本の制御手法として

フィードバック制御 (feedback control)

フィードフォワード制御 (feedforward control)



ロボットの速度制御実験

4. 速度制御（つづく）

今回利用する制御アルゴリズムは、エンコーダの性能上の理由で、PI制御則に限定しています。制御則は以下の式で表せます。

$$E = V_d - V, \quad \tau = K_p E - K_I \int E$$

ここで、 V_d は速度の目標値、 V は現在計測された速度とし、速度の誤差は E となります。 K_p は比例制御ゲイン、 K_I は積分制御ゲインと呼びます。計算された制御出力 τ をモータへ出力し、その車輪の速度制御を行います。

制御システムには、安定性、応答特性、定常誤差など性質と特性指標があります。

制御システムに影響するのは、各制御要素のゲインの値となります。

理論的に計算することもあります。今回は、実験的に求める方法
ゲイン・チューニングをして、良い速度制御システムを目指します。

ロボットの速度制御実験

【課題4】PI制御のサンプルプログラム（スタートガイドテキストのP46）の比例ゲイン K_p と積分ゲイン K_i をチューニングし、よりよい速度制御を実現しましょう。

この課題に関しては、以下の四つの実験をそれぞれ複数回実行し、データを記録して、**平均値や最大誤差値などを計算・分析し、レポートを作成**しましょう。

実験4-1：制御のIの成分を入れない、P成分のみの実験を試みる

（ $K_i=0.0$ 、 $K_p=0.1$ ）目標速度は20cm毎秒

無負荷の走行と負荷あり（斜面か箱押し）の走行時のロボットの速度を計測する

実験4-2：制御ゲインの調整（チューニング）：目標速度は20cm毎秒

P制御ゲイン K_p を 走行時車輪が発振するまで、徐々にあげ、発振直前の値を探し出す。その値の0.8ぐらいを最適なP制御ゲインにする

そのゲインでの負荷ありの走行時のロボットの速度を計測する

実験4-3：制御のI成分(K_i)を復活し、無負荷と負荷有りの走行の速度を計測し、I成分のない実験の結果を比較し、PI制御のI成分の効果を確認する。

実験4-4：チューニングした制御ゲインでロボットの走行性能（異なる速度）を確認する。

ロボットの速度制御実験

実験4-1：制御のI成分を入れない、P成分のみの実験を試みる

($K_i=0.0$ 、 $K_p=0.1$) 目標速度は20cm毎秒
無負荷の走行と負荷あり（斜面か箱押し）の走行時のロボットの速度を計測する

実験4-2：制御ゲインの調整（チューニング）：目標速度は20cm毎秒

P制御ゲイン K_p を 走行時車輪が発振するまで、徐々にあげ、発振直前の値を探し出す。その値の0.8ぐらいを最適なP制御ゲインにする
そのゲインでの負荷ありの走行時のロボットの速度を計測する

実験4-3：制御のI成分(K_i)を復活し、無負荷と負荷有りの走行の速度を計測し、I成分のない実験の結果と比較し、PI制御のI成分の効果を確認する。

実験4-4：チューニングした制御ゲインでロボットの走行性能（異なる速度）を確認する。

ロボットの速度制御実験

速度制御レポート：提出期限（2021年12月20日 23:50）

内容： （実験目的，実験手法，実験データと考察が必要）

1. ロボットの最高速度の特性を実験で確認（課題1）
2. ロボットの定速走行特性を実験で確認（課題2）
（走行直進性，距離精度）
3. 車輪エンコーダによる回転速度の計測（課題3）
（今回はカウンタ値の変化を写真記録）
4. 速度制御特性に関する実験（課題4）
P制御の効果，I制御の効果の確認（実験4-1，4-2，4-3）
ゲインチューニングによりPI制御での特性の確認（実験4-4）
5. 軌道制御により図形作成実験
（12月12日の軌道制御の回で詳細説明）