

位置制御に適した制御器設計の研究

電気科 3年 久保田 優太
指導教諭 高田 直人

1. はじめに

自律移動ロボットにおける直進や旋回動作は、目標値と現在値との比較により、訂正動作を繰り返しながら目標値に近づくように制御することが一般である。このような位置制御系の制御器には、 $PI-D$ (微分先行型比例積分)制御や $I-PD$ (比例微分先行型積分)制御動作が使われてきた。しかし、 P (比例動作)、 I (積分動作)、 D (微分動作)のそれぞれのゲインを最適に選ばないと、応答特性にオーバシュート(行き過ぎ量)が発生することが課題であった。自律移動ロボットの位置制御系におけるオーバシュートは、行き過ぎ量を修正するために修正動作を繰り返すことになり、競技ロボットにおいては大きなタイムロスにつながる。この課題を解決するために本研究では、位置制御に適した制御器について研究を行った。

2. 制御対象と制御器設計

2.1 制御対象

制御対象は図 1 に示すような自律移動ロボット ($W250mm \times D250mm \times H190mm$, 重量 $1848g$) とした。駆動輪は 4 輪独立駆動方式のメカナムホイール, ヨー軸(水平)方向センサとして慣性センサ $BNO055$ を使用した。また, 前後・左右方向の移動距離はロータリエンコーダ (マイクロテックラボラトリ社製 $MES(4)-17-360P$, 360 パルス/回転) を 4 逡倍して 1440 パルス/回転の分解能で使用した。

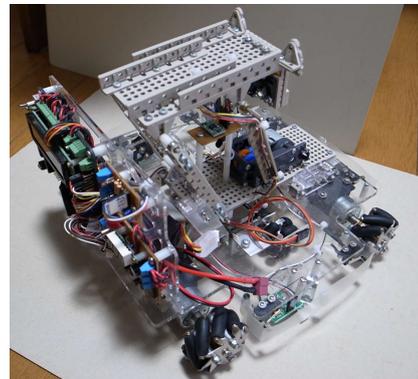
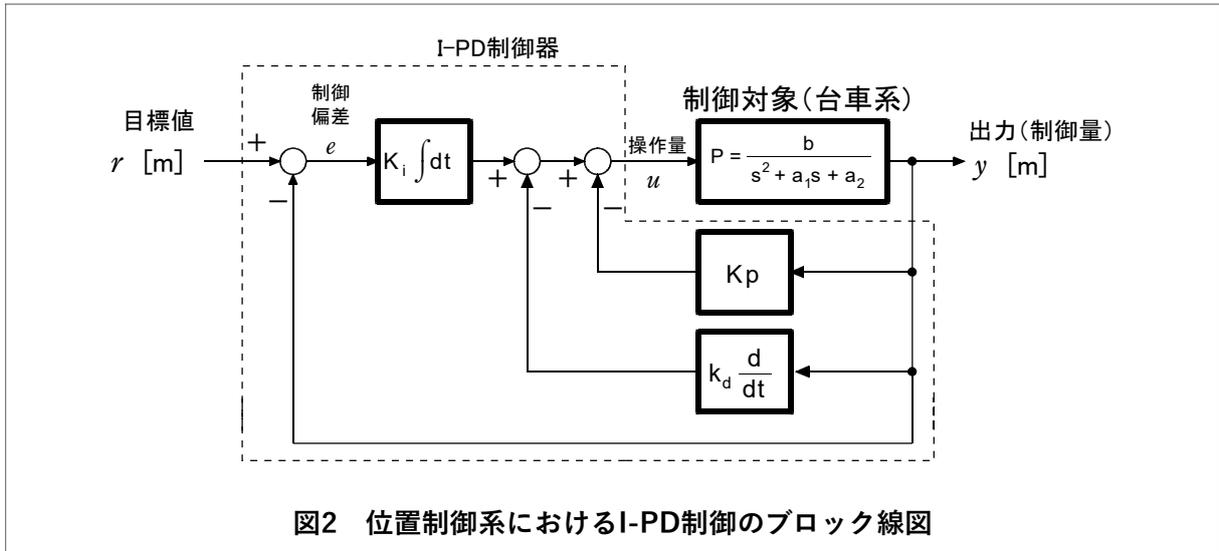


図1 制御対象となる自律移動ロボット

2.2 制御器設計

制御器には $I-PD$ (比例・微分先行型積分)制御を採用した。一般的な PID 制御では, 台車が初期位置状態で目標値を新たに設定した場合, 大きな操作量が発生し, 駆動モータや台車系に悪影響を及ぼすことがある。それに対して $I-PD$ 制御は, 初期操作量が抑制されるため, オーバシュートが発生しにくいという特徴がある。位置制御系における $I-PD$ 制御のブロック線図を図 2 に示す。この制御系は, 車体の前後動作と旋回動作のどちらにも適用することができる。図 2 に示すブロック線図において, 制御対象(台車系)に対する操作量 u は式(1)になる。



$$u = k_i \int (r - y) dt - k_p y - k_d \frac{d}{dt} y \quad (1)$$

また、制御対象の台車モデルを $P = \frac{b}{s^2 + a_1 s + a_2}$ としたとき、目標値 $r(m)$ から制御量 $y(m)$ までの伝達関数 P_{cl} は式(2)となる。

$$P_{cl} = \frac{b k_i}{s^3 + (a_1 + b k_d) s^2 + (a_2 + b k_p) s + b k_i} \quad (2)$$

I-PD 制御器の比例ゲイン k_p 、微分ゲイン k_d 、積分ゲイン k_i の設計には、式(2)より台

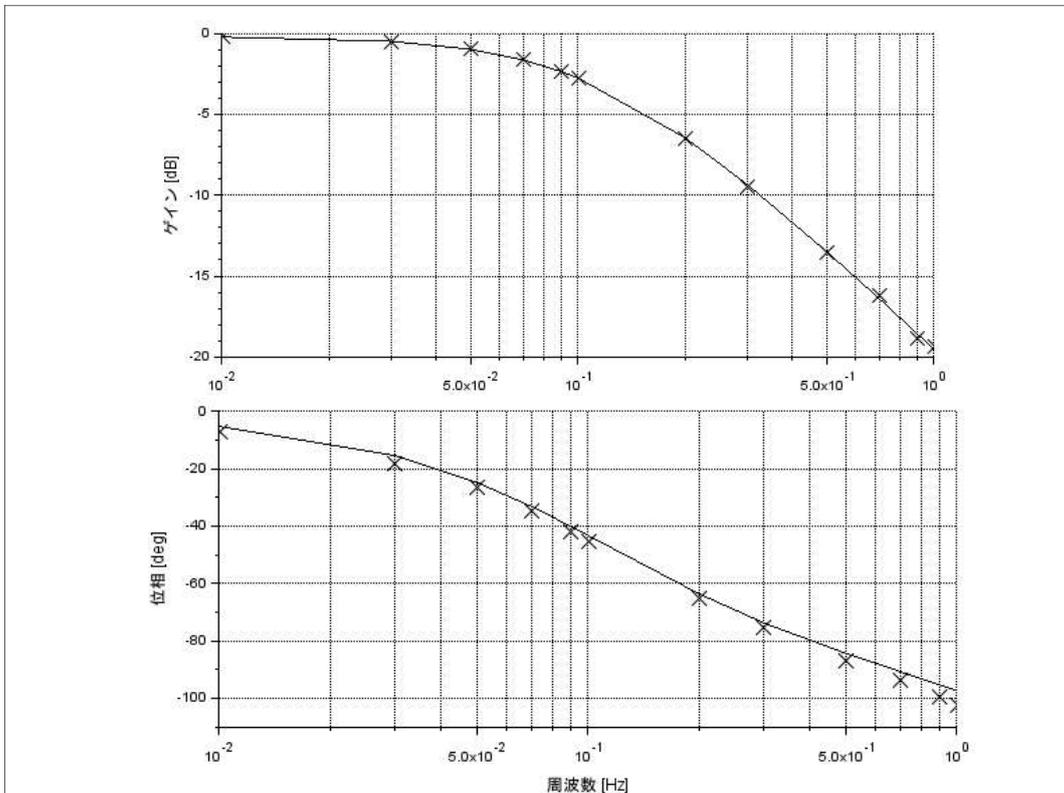


図3 台車系の前後方向における周波数応答 (振幅0.5m, Kp=10)

車系のパラメータ a_1 , a_2 , b が必要である。そこで台車系のパラメータを求めるために、周波数応答法によってパラメータ同定を行った。同定実験は、比例制御系を構成した台車系において、入力信号の振幅を $0.5m$ 、比例ゲイン $kp=10$ とし、 $0.01Hz$ から $1.0Hz$ までの正弦波信号を目標値として与えた。入力信号に対する制御対象(台車)応答は、ロータリエンコーダを使って制御量を読取り、振幅比と位相差から図 3 のようなボード線図が得られた。

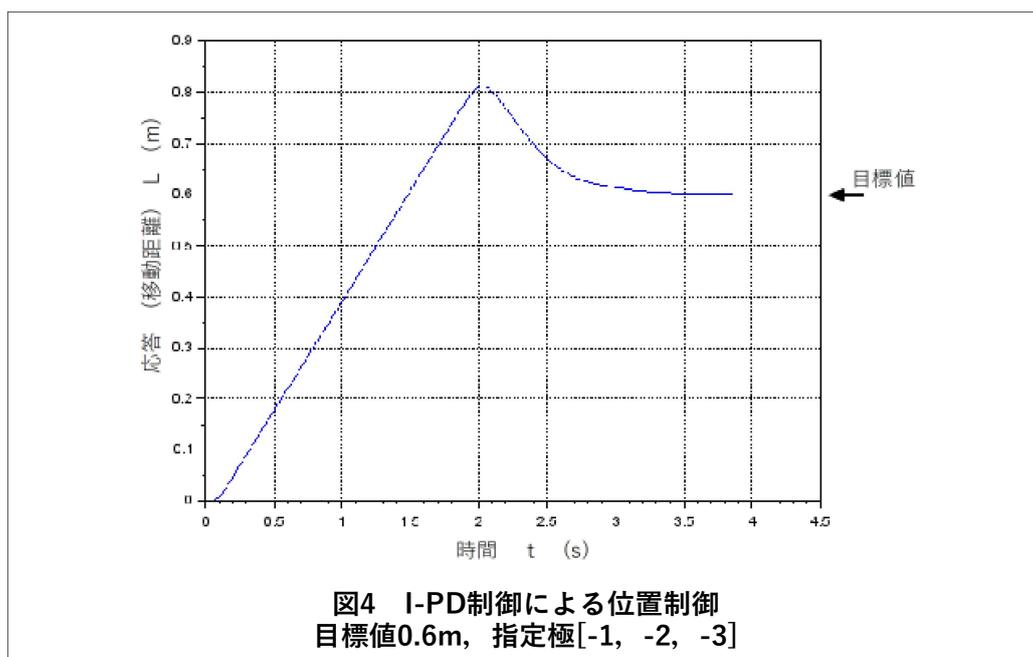
このボード線図より、台車系モデルを近似的に $P = \frac{b}{s^2 + a_1s + a_2}$ とした場合、実験値に最も適合するパラメータ a_1, a_2, b を数値探索によって求め $a_1 = 26.20, a_2 = 0.3720, b = 1.748$ が得られた。比例ゲイン kp 、微分ゲイン kd 、積分ゲイン ki は式(2)より、分母の特性多項式の極を $\lambda_1, \lambda_2, \lambda_3$ とすると、式(3)によって設計することができる。

$$k_p = \frac{\lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_1 \lambda_3 - a_2}{b} \quad k_d = - \frac{(a_1 + \lambda_1 + \lambda_2 + \lambda_3)}{b} \quad k_i = - \frac{\lambda_1 \lambda_2 \lambda_3}{b} \quad (3)$$

3. 制御実験結果

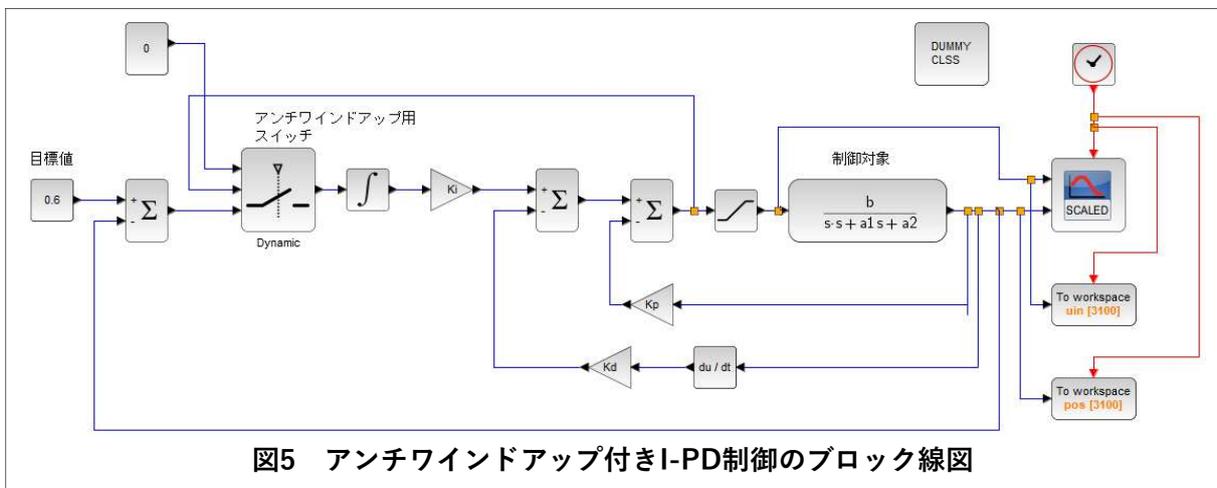
3.1 I-PD制御

I-PD 制御による制御器のゲイン設計は式(3)で求めた。台車が目標値の追従を完了する整定時間を約 1 秒とし、指定極を $\lambda_1 = -2, \lambda_2 = -3, \lambda_3 = -4$ とすると、比例ゲイン $kp = 14.665$ 、積分ゲイン $ki = 13.734$ 、微分ゲイン $kd = -9.845$ となった。この値を式(1)に代入することによって得られる操作量 u によって駆動輪のモータを制御することにより、目標値に対する応答を調べた。図 4 はその実験結果である。目標値 $0.6m$ に対して最大 $0.8m$ の行き過ぎ量が発生している。指定極を変更して行き過ぎ量が発生しない比例・積分・微分ゲインに変更しても目標値への収束性が悪く、整定時間が長くなる傾向が見られた。



3.2 アンチwindアップ付きI-PD制御

I-PD 制御における位置制御で行き過ぎ量が発生してしまう原因は、積分器の動作にある。積分器は目標値と現在値との差である制御偏差（以後、たんに偏差と称す）を積分するため、特に偏差が大きいと積分器の値が増加し続けて操作量が上限に到達する、いわゆる飽和現象が起こる。さらに飽和が発生しても積分器は増大し続けるため、行き過ぎ量が発生する。そこで、操作量が飽和したら積分器の動作を停止するアンチwindアップという対策を試みた。積分器の動作を停止するとは、操作量 u が飽和したら、積分器への入力をゼロに置き替えることである。こうすることにより、操作量が飽和した場合、積分器の増加を抑制でき、積分出力が過度に成長することを回避できる。図 5 にアンチwindアップ付き *I-PD* 制御のシミュレーションのためのブロック線図を示す。



応答の指定極を $\lambda_1 = -4$, $\lambda_2 = -6$, $\lambda_3 = -8$ としたとき、比例ゲイン $k_p = 59.301$, 積分ゲイン $k_i = 109.871$, 微分ゲイン $k_d = -4.695$ となった。図 5 に示すブロック線図によるシミュレーションと実験結果を図 6 に示す。図 6 の上図が制御量である。目標値 $0.6m$ に対して、台車の移動距離がシミュレーションと実験値が良好に一致していることがわかる。また、図 6 の下図が操作量に関するシミュレーションと実験結果の比較である。操作量は $6.0V$ を上限に設定してあるため、実験値ではほぼ $6V$ に制約されていることがわかる。この期間は積分器への偏差入力に停止しており、積分器が過大な値に成長してしまうことを防いでいる。目標値と現在値の偏差が小さくなると操作量も小さくなり、最終的に目標値に整定できている。このことから、操作量の飽和時には積分器の動作を停止するアンチwindアップが極めて重要な働きをすることが確認できた。

リスト 1 は前後方向に移動するための制御演算処理である。前後方向の移動距離はエンコーダカウンタのパルス数から知ることができる。この値と目標値との偏差を求め、アンチwindアップ付き *I-PD* 制御演算処理をおこなっている。この関数は、タイマ割込を使って $10ms$ ごとに制御演算を行い、操作量 u を求めている。さらに、操作量 u は $[V]$ を単位とする数値であるため、この値からモータに与える *PWM* 制御のデューティ比 ($0 \sim 100\%$) に変換している。

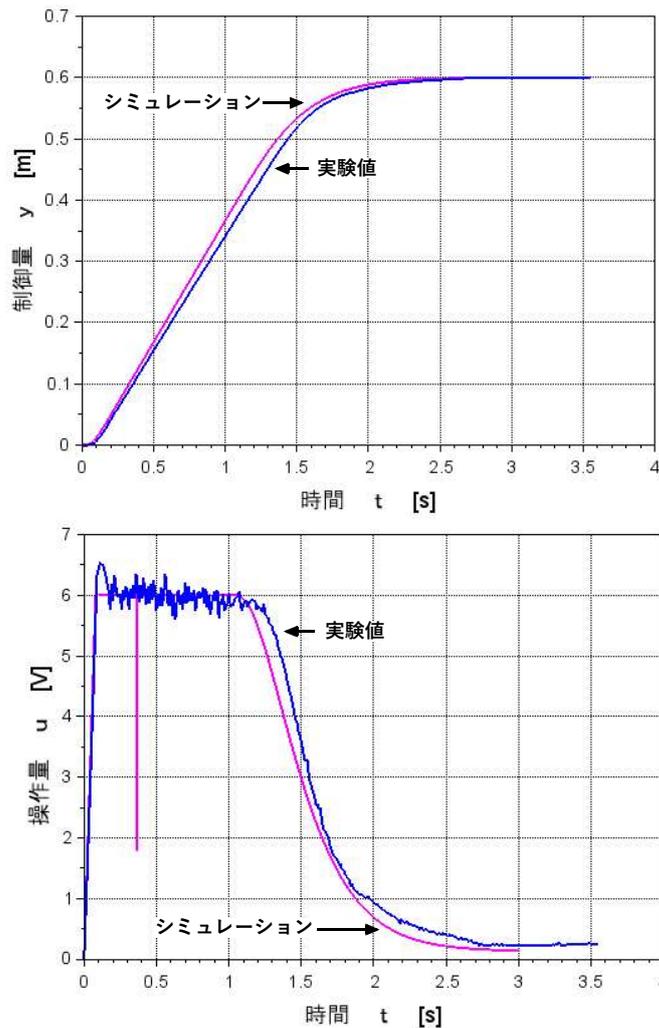


図6 アンチwindアップ付きI-PD制御による制御実験結果

リスト1 アンチwindアップ付きI-PD制御によるフィードバック制御関数（前後動作）

```

void enc_move_u(void) {
    // 直進走行制御
    if(f.enc == 1){
        // 直進走行制御フラグがセットされていれば制御演算
        get_enc_dis(); // エンコーダによる移動距離の取得
        er_m = xy_cmd - dyy; // 前後方向の偏差
        dxy = dyy; // 前後方向の現在値
        if( fabs(uxy) > UMAX ) { } // ★★アンチwindアップ（積分停止）
        else dis_int += er_m * STIME; // 積分
        uxy = Kim * dis_int - Kpm * dxy - Kdm * (dxy - dxy_old)*INVST; // I-PD制御演算
        dxy_old = dxy;
        if(fabs(er_m) < 0.003) f.enc = 0; // 偏差の収束判定 (3mm以内なら制御終了)
        if(uxy >= 0){
            set_motor_dir1(F,F); set_motor_dir2(F,F); // メカナムホイールの前進回転
        }
        else { set_motor_dir1(R,R); set_motor_dir2(R,R); // メカナムホイールの後退回転
              uxy = - uxy;
            }
        dtm = uxy * V2DT; // 操作量からデューティ比に変換
        if(dtm > 90) dtm = 90; // デューティの上限リミッタ
        set_pwm_duty1(dtm, dtm); set_pwm_duty2(dtm, dtm);
    }
    else {
        set_motor_dir1(B,B); set_motor_dir2(B,B);
    }
}

```

4 おわりに

一般的な I - PD 制御と、操作量が飽和した際に積分器の動作を停止するアンチwindアップ付き I - PD 制御についてシミュレーションと実験をとおして比較を行った。この結果、アンチwindアップ付き I - PD 制御ではシミュレーションと実験値が良好な一致が見られ、操作量が飽和した際に積分器の動作を停止することが極めて効果的であることがわかった。また、理論と実験結果との整合性もよく、台車系のパラメータ同定結果も正しい値が得られていることが確認できた。この研究成果は、本校のロボット研究製作部において、自律移動ロボットや課題研究などにおいて、制御系設計に活用していく予定である。

最後に、研究助成にあたり尽力いただいた関係者の皆様に感謝を申し上げます。

参考文献

- [1] 高校数学でマスターする 現代制御とデジタル制御, 小坂 学著, コロナ社(2015)