

序説

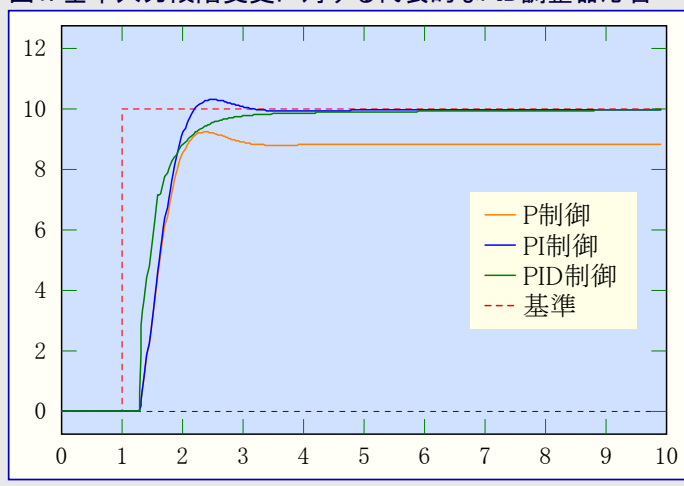
この応用記述は離散 比例/積分/微分(PID:Proportional-Integral-Differential)制御器の簡単な実装を記述します。

基準の値や状態に変更するためにシステムの出力を制御する必要がある応用での動作時、制御演算法の実装が必要になるかもしれません。このような応用の例は電動機制御や、温度、圧力、流速、速度、力及びその他変数の制御があります。PID制御器はその変数が他のいくつかの処理変数の操作によって影響を及ぼされ得る限り、何れかの測定可能な変数を制御するのに使うことができます。

多くの制御対策は時を越えて使われて来ましたが、PID制御器はそれが簡単で良好な性能のために“工業標準”になりました。

PID制御器とその関係についての更なる情報に関して、読者は他の資料、例えばK. J. Astrom & T. Hagglund著“*PID Controllers*”(1995)を調べるべきです。

図1. 基準入力段階変更に対する代表的なPID調整器応答



特徴

- 簡単な離散PID制御器演算法
- 全てのAtmel® AVR®デバイスで支援

目次

序説	1
特徴	1
1. PID制御器	3
1.1. 比例項	3
1.2. 積分項	3
1.3. 微分項	4
1.4. 比例、積分、微分の項の連携	4
1.5. パラメータ調整	4
1.6. 離散PID制御器	5
1.6.1. 演算法の背景	5
2. 実装	5
2.1. 積分項結び	6
3. 更なる開発	6
4. 参考文献	6
5. 改訂履歴	6

1. PID制御器

下図でPID制御器を持つシステムの図式が示されます。PID制御器は測定した処理値 Y を基準設定値 Y_0 と比較します。そして新しい処理入力 u を計算するために差または誤差 e が処理されます。この入力 u は測定した処理値を要求した設定値に戻す調整を試みますでしょう。

PID制御器のような閉路制御機構の他の手段が開路制御器です。開路制御(還元なし)は多くの場合に満足せず、システム特性のために度々達成できません。システム出力からの還元を追加することによって性能が改善できます。

簡単な制御法と違い、PID制御器は信号の変更速度と履歴を基にして処理入力をあえて操作します。これはより大きな精度と安定な制御法を与えます。

基本的な考えは、制御器が感知器によってシステム状態を読むことです。そして誤差値を生成するために要求された基準から測定値を減算します。誤差は、比例項によって現在を扱い、積分項を使って過去から回復し、微分項によって未来を予測する、3つの方法で管理されるでしょう。

右図はPID制御器の図式を示し、ここで K_p は比例項の利得定数、 T_i と T_d は各々積分項と微分項の時間定数を意味します。

図1-1. PID制御器を持つ閉路システム

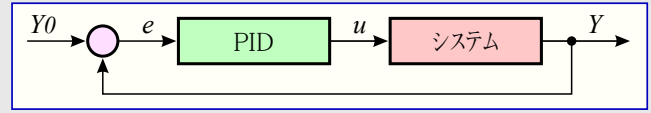
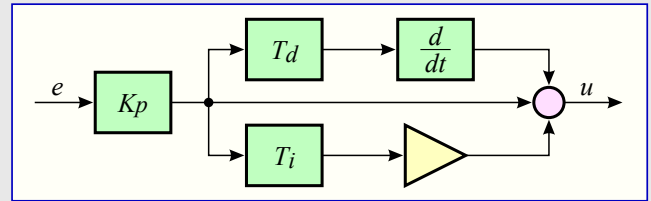


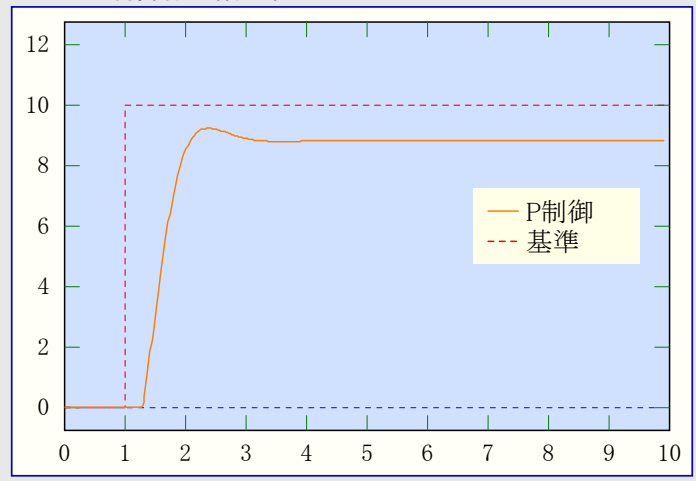
図1-2. PID制御器図式



1.1. 比例項

比例項(P)は誤差に比例したシステム制御入力を与えます。システム制御入力が0でシステム処理値が要求された値に等しい時を除き、P制御だけを使うことは全ての場所で定常的な誤差を与えます。右図ではシステム処理値での定常的な誤差が要求された値(基準)で変更された後で出現します。大きすぎるP項の使用は不安定なシステムにさせます。

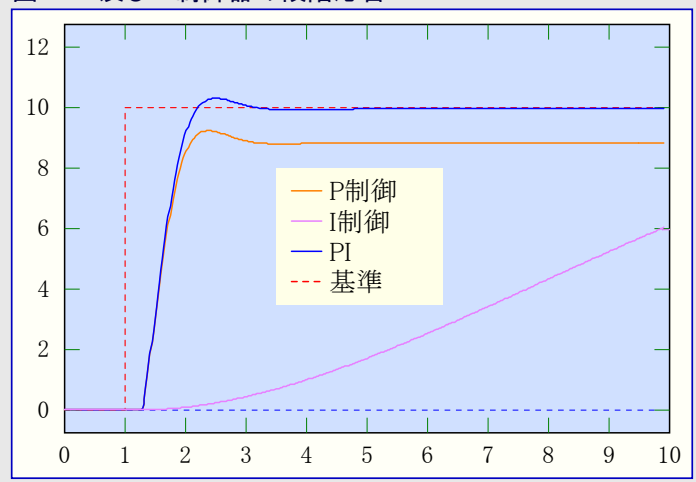
図1-3. P制御器段階応答



1.2. 積分項

積分項(I)はシステム制御入力に以前の誤差の総和からの付加を与えます。誤差の総計はシステム処理値が要求された値と等しくなるまで継続され、これは基準が安定な時に定常的な誤差に帰着しません。I項の最も平凡な使用は一般的にP項を伴い、PI制御器と呼ばれます。I項だけの使用は遅い応答と度々発振するシステムを与えます。右図はI及びPI制御器の段階応答を示します。PI制御器で見られるように、応答は定常的な誤差を持たず、I制御器は非常に緩慢です。

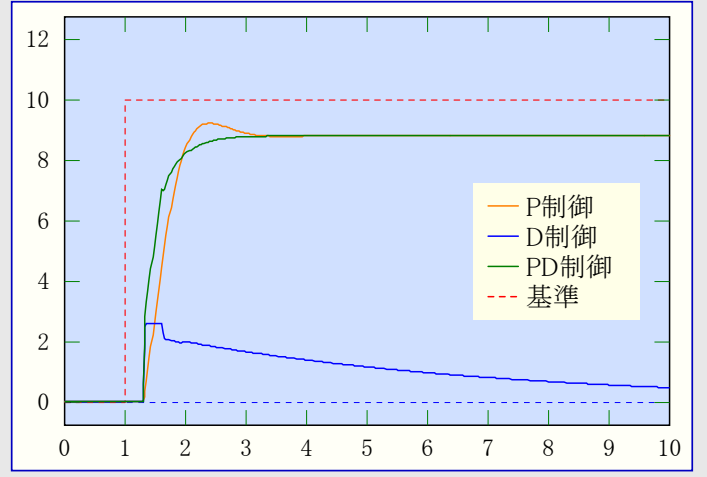
図1-4. I及びPI制御器の段階応答



1.3. 微分項

微分項(D)はシステム制御入力に誤差の変化速度からの付加を与えます。誤差の急な変化はシステム制御入力に付加を与えます。これはシステム状態または基準値の急な変化への応答を改善します。D項は代表的にPまたはPIと共にPDまたはPID制御器として使われます。大きすぎるD項は通常、不安定なシステムにさせます。右図はD及びPD制御器応答を示します。PD制御器の応答はP制御器よりも速いシステム処理値の上昇にさせます。D項が本質的に誤差信号での高域通過濾波器(ハイパスフィルタ)として働き、故にシステムの不安定性を容易に誘引し、雑音にもっと敏感にさせることに注意してください。

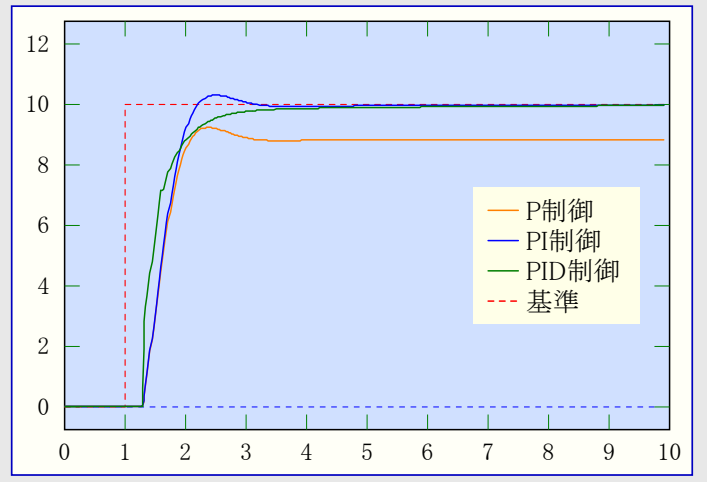
図1-5. D及びPD制御器の段階応答



1.4. 比例、積分、微分の項の連携

PID制御器として全ての項を共に使うことが通常、最良の性能を与えます。右図はP、PI、PID制御器を比べます。PIは定常的な誤差を取り去ることによってPを改善し、PIDは高速応答と行き過ぎ(オーバーシュート)をなくすことによってPIを改善します。

図1-6. P,PI及びPID制御器の段階応答



1.5. パラメータ調整

必要とするPIDパラメータを探す最善の方法はシステムの数式モードからで、そこでパラメータは要求された応答を得るために計算できます。度々システムの詳細な数式内容が利用できず、PIDパラメータの実験的な調節が行われなければなりません。PID制御器用の項を探すのは挑戦的な作業になり得ます。システム特性と各項の動作の仕方についての良い知識が重要です。処理変更または設定点変更での最適な動きは手近の応用に依存します。いくつかの処理は設定点から変わり易い処理の行き過ぎ(オーバーシュート)を許してはなりません。他の処理は設定点到達でエネルギー消費を最小にしなければなりません。一般的に安定性が最大の必要条件です。処理はどんな組み合わせや設定点に関しても発振してはなりません。更に加えて、安定化の効果は或る制限時間内に現れなければなりません。

PID閉路を調整する多くの方法が存在します。方法の選択は調整に関してオフラインで行えるか否かに大きく依存します。ジグラー、ニコルス(Ziegler-Nichols)法が良く知られたオンライン調整計画です。この方法の最初の段階はIとDの利得を0に設定し、出力で安定且つ持続する発振が(可能な限り近く)得られるまで増加します。そして臨海利得 K_c と発振周期 P_c が記録され、P,I,D値は下表を使うことによって補正されます。

PID制御器の性能を最適化するためにパラメータの更なる調整が度々必要です。

読者はPID制御器があまり良く動作しない、または与えられたシステム状態の小さな領域の周辺でだけ動くシステムがあることに注意すべきです。非直線なシステムがこのようになり得ますが、一般的に問題はシステムが不安定で、入力の効果がシステム状態に依存する時のPID制御で度々起きます。

表1-1. ジグラー、ニコルス パラメータ

制御器種別	K_p	T_i	T_d
P	$0.50 \times K_c$		
PD	$0.65 \times K_c$		$0.12 \times P_c$
PI	$0.45 \times K_c$	$0.85 \times P_c$	
PID	$0.65 \times K_c$	$0.50 \times P_c$	$0.12 \times P_c$

1.6. 離散PID制御器

離散PID制御器は誤差を読み、計算し、そして採取周期 T で与えられた時間間隔で制御入力を出力します。採取時間はシステムの最短時間定数より短くあるべきです。

1.6.1. 演算法の背景

簡単な制御法と違い、PID制御器は信号の変更速度と履歴を基にして処理入力をあえて操作します。これはより大きな精度と安定な制御法を与えます。

図1-2はPID制御器の図式を示し、ここで K_p は比例項の利得定数、 T_i と T_d は各々積分項と微分項の時間定数を意味します。

このシステムの伝達関数は以下です。

$$\frac{u}{e}(s) = H(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

これは時間領域で e に関する u を与えます。

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\sigma) d\sigma + T_d \frac{de(t)}{dt} \right)$$

離散形式を得るために以下を使って積分項と微分項を近似します。

$$\int_0^t e(\sigma) d\sigma \doteq T \sum_0^t e(k) \quad \frac{de(t)}{dt} \doteq \frac{e(n) - e(n-1)}{T} \quad t = nT$$

ここで n は時間 t での離散段階です。

これは制御器に以下を与えます。

$$u(n) = K_p e(n) + K_i \sum_0^t e(k) + K_d (e(n) - e(n-1))$$

ここで K_i と K_d は以下です。

$$K_i = \frac{K_p T}{T_i} \quad K_d = \frac{K_p T_d}{T}$$

要求された処理値での変更が制御入力で何れかの望まない急な変化にするのを避けるため、制御器は微分項を処理値だけを基準にすることによって改善されます。

$$u(n) = K_p e(n) + K_i \sum_{k=0}^n e(k) + K_d (y(n) - y(n-1))$$

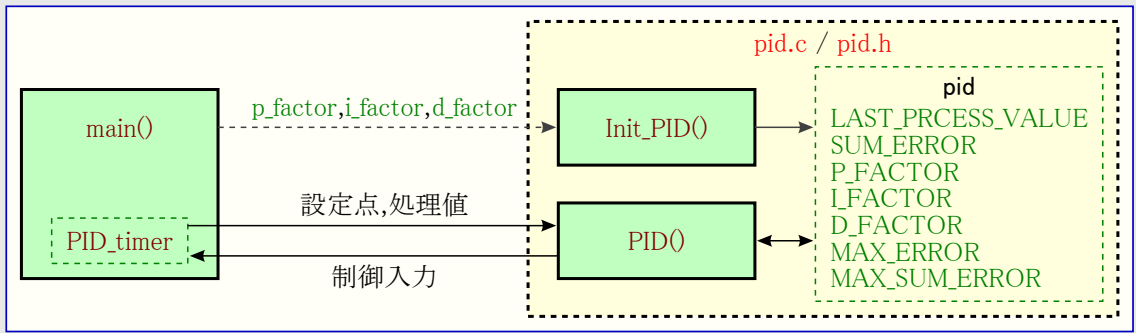
2. 実装

例のコードは”Atmel START(開始)”用には書かれています。これはAtmel Studio 7とIAR™ IDEの両方に対してAtmel STARTの”BROWSE EXAMPLES(例閲覧)”からダウンロードすることができます。

ダウンロードした.atzipファイルをダブルクリックしてください。プロジェクトがAtmel Studio 7にインポートされます。

IARでプロジェクトをインポートするには、”Atmel START in IAR (IARでのAtmel開始)”を参照し、’Atmel Start Output in External Tools (外部ツールでのAtmel開始出力)⇒IAR’を選んでください。

図2-1. 実演応用の構成図



上図では実演応用の単純化された構成図が示されます。

PID制御器はその状態とパラメータを格納するのに構造体を使います。この構造体はmain()で初期化され、それへのポインタだけがInit_PID()関数とPID()関数に渡されます。

PID()関数は時間間隔 T 毎に呼び出されなければならず、これは時間間隔経過時にPID_timerフラグを設定するタイマによって行われます。PID_timerフラグが設定されると、主ルーチンは要求された処理値(設定点)とシステム処理値を読み、PID()関数を呼び出して制御入力の結果を出力します。

精度を増すためにp_factor,i_factor,d_factorは128倍に拡大されます。PID演算法の結果は後で1/128することで尺度を戻されます。この128の値はコンパイラでの最適化を許すのに用いられています。

$$P\text{係数} = 128 \times K_p$$

更に加えて、I係数とD係数の効果は採取時間 T に依存します。

$$I\text{係数} = 128 \times K_p \times \frac{T}{T_i}$$

$$D\text{係数} = 128 \times K_p \times \frac{T_d}{T}$$

2.1. 積分項結び

処理入力 u がかなり高い値に達したとき、それはいくつかの方法で制限されます。PID制御器の内部的な数値範囲、制御器の出力範囲、増幅器の束縛、処理それ自身のどれか……。これは基準設定点と測定された処理値でかなり大きな差がある場合に起き、代表的にはシステムが可能な扱い能力よりも大きな妨害や負荷を処理が持つためです。

制御器が積分項を使う場合、この状況は問題になり得ます。積分項は最後の状況を長く総和に含み、より大きな妨害や負荷が隠されると、PID制御器は積分総和が通常に戻るまで処理入力を過補償するでしょう。

この問題は多くの方法で避けられます。本実装ではMAX_I_TERMより大きくなるのをそれに許さないことによって最大積分総和が制限されます。MAX_I_TERMの正しい量はシステムと使う採取時間に依存します。

3. 更なる開発

ここで示されたPID制御器は単純化した例です。この制御器は快適に動作するでしょうが、或る応用で制御器をもっと強固に(暴走/溢れ)までもする必要があるかもしれません。積分項で飽和修正を追加し、比例項をシステム処理値だけに基礎付けることが必要で有り得ます。

I係数とD係数の計算では採取時間 T が式の一部です。使う採取時間 T が1秒より大きいかまたはずっと小さければ、I係数とD係数どちらかに関する精度が貧弱になるでしょう。積分項と微分項に関する精度を保つように尺度調整をしてPID演算法を書き直すことを考慮してください。

4. 参考文献

"PID Controllers: Theory, Design, and Tuning" K. J. Astrom & T. Hagglund著 国際計測制御学会 1995年発行

5. 改訂履歴

文書改訂	日付	注釈
2558A	2006年5月	初版文書公開
2558B	2016年8月	新雛形といくつかの微細な変更
2558C	2016年9月	「特徴」項を更新 Atmel START関連を追加することによって「実装」章を更新

Atmel®, Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®, AVR®, tinyAVR®, megaAVR®とその他は米国及び他の国に於けるAtmel Corporationの登録商標または商標です。他の用語と製品名は一般的に他の商標です。

お断り: 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイト¹に位置する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

安全重視、軍用、車載応用のお断り: Atmel製品はAtmelが提供する特別に書かれた承諾を除き、そのような製品の機能不全が著しく人に危害を加えたり死に至らしめることがかなり予期されるどんな応用("安全重視応用")に対しても設計されず、またそれらとの接続にも使用されません。安全重視応用は限定なしで、生命維持装置とシステム、核施設と武器システムの操作用の装置やシステムを含みます。Atmelによって軍用等級として特に明確に示される以外、Atmel製品は軍用や航空宇宙の応用や環境のために設計も意図もされていません。Atmelによって車載等級として特に明確に示される以外、Atmel製品は車載応用での使用のために設計も意図もされていません。

© HERO 2021.

本応用記述はAtmelのAVR221応用記述(Rev.2558C-09/2016)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。