

実時間Linux上のデータベースを利用した 可変デジタルPID制御系の基礎検討

山田 健仁^{*1} 杉山 淳一^{*2} 寺西 信^{*3}

Variable Digital PID Controller using Control Parameters Database Management System on Real-Time Linux

Takehito YAMADA ^{*1}, Junichi SUGIYAMA ^{*2} and Makoto TERANISHI ^{*3}

Abstract

We propose a system which observes a distribution of each control parameter to the amount of the features of a step response. The step response in the case of changing each control parameter of a digital PID control system is measured comprehensively, and the response and the parameters are registered into a database. The amount of the features of a step response is calculated from a set of the measured data. The system was examined in a positioning mechanism which used DC motor as a control plant. The characteristics of the tendency to the control parameters distribution was observed with this system.

Key Words: PID control, digital control, optimization, database, ART-Linux

1. まえがき

PID制御では、目標値と出力との偏差の比例(P)、積分(I)、微分(D)の各動作の合成で制御対象への操作量が決定される。このため比例定数、積分動作の時定数、微分動作の時定数の各パラメータ（以下 制御パラメータ）によって制御系の性能や安定性が決定される。これらの制御パラメータの理論的な決定手法は多く提案されている^{1), 2)}。しかし、実際は制御対象とそのモデルとの誤差や制御対象に含まれる非線形性のために、試行錯誤や経験によって最終的な制御パラメータが決定されることが多い。また、このようにして決定された制御パラメータであっても、制御対象の変動や外乱の変化によって、最適な制御パラメータも変動することになる。

本研究では、コンピュータ上にデジタルPID制御則を実装し、多数の異なる制御パラメータによって制御を行い、その制御応答結果と評価結果をリアルタ

イムでデータベース(以下DB)に記録し、その結果から最適な制御パラメータを自動的に決定するシステムの実現を最終的な目標とする。このシステムの実現のための基礎実験として、制御応答結果と制御パラメータの分布を自動的に観測・記録するシステムをART-Linux上に構築した。この観測結果から得られた知見は、制御パラメータの自動最適化のための手法に有効と考えられる。

2. 可変デジタルPID制御システムの構成

図1に本実験に使用したデジタルPID制御実験システムの構成を示す。制御対象は、DCモータによりその傾斜角度が操作できる金属のレール上で、金属球の位置を制御する機構である^{3), 4)}。金属球の位置は、ポテンショメータの原理で検出できるようになっている。この制御機構の運動方程式は式(1)で表せ、金属レールの傾斜角度が小さいとして線形化するとそ

^{*1} 情報電子工学科

^{*2} 豊橋科学技術大学

^{*3} 教育研究支援センター

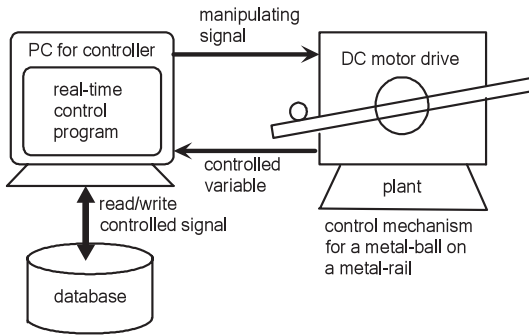


図1 データベースを利用した可変デジタルPID制御システム

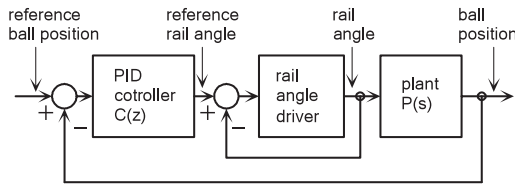


図2 制御系の構成ブロック図

の伝達関数は式(2)と表せ、基本的に2型であることが分かる。

この制御対象に式(3)の直列型のPID制御器を適用する。制御器を実装する際には、コンピュータ制御によるデジタル制御器とするため、式(3)を式(4)の双線形Z変換で離散化した式(5)より導出される制御演算則により実現する。図2に制御系のブロック図を示す。

$$mg \sin \theta = m \frac{d^2 x}{dt^2} \quad (1)$$

$$P(s) = \frac{X(s)}{\Theta(s)} = \frac{g}{s^2} \quad (2)$$

$$C(s) = k \cdot \left(\frac{s + \frac{1}{aT}}{s + \frac{1}{T}} \right) \cdot \left(\frac{s + b}{s} \right) \quad (3)$$

$$s = \frac{2}{T_s} \cdot \frac{z - 1}{z + 1} \quad (4)$$

$$C(z) = \frac{U(z)}{E(z)} = \frac{\alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2}}{\beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2}} \quad (5)$$

$$\alpha_0 = k(4aT + 2T_s + bT_s^2 + 2abTT_s)$$

$$\alpha_1 = k(-8aT + 2bT_s^2)$$

$$\alpha_2 = k(4aT - 2T_s - 2abTT_s + bT_s^2)$$

$$\beta_0 = 4aT + 2aT_s, \quad \beta_1 = -8aT$$

$$\beta_2 = 4aT - 2aT_s, \quad T = \frac{1}{2\pi f_{gc} \sqrt{a}}$$

T_s はサンプリング周期, f_{gc} は開ループのゲイン交差周波数である。デジタル制御演算則実装の際、サンプリング周波数 f_s をゲイン交差周波数 ($f_{gc} = 1$ [Hz]) の30倍以上の33 [Hz] ($T_s = 30$ [ms]) とし、制御系の離散化による影響を抑制するようにした⁵⁾。

図1のシステムにおいて、コンピュータ上に実装するデジタル制御器部分はリアルタイム処理が必要であること、及び、制御応答結果と制御パラメータの分布を自動的に観測・記録するためにデータベースが必要であることから、OSをART-Linux⁶⁾として、データベース管理システムにLinux上で稼働する軽量データベースのSQLite3.3.8⁷⁾を使用した。また、実際の制御システムへの応用を勘案して、ワンチップマイコン上にこの制御システムが実装出来るように、制御中のデータベースはメモリ上に展開するファイルシステムとしている。

本システムで使用するデータベースは、以下の3つのテーブルからなる。なお、SQLite3.3.8は外部キーをサポートしていないため、下表のような構成とした。

- pid_id テーブル:

このテーブルはPID制御のパラメータを記録するテーブルである(表1)。

- pid_act テーブル:

このテーブルは制御中に観測される制御系の入出力動作を記録するテーブルである(表2)。

- pid_result テーブル:

このテーブルは、pid_act テーブルに記録した制御応答

表1 pid_id テーブルの構造

列名	型	説明
id	int	PID制御パラメータのID (主キー)
a	float	微分 (D) 動作のパラメータ
b	float	積分 (I) 動作のパラメータ
k	float	比例 (P) 動作のパラメータ
pos	float	金属球の目標値 (位置)

表2 pid_act テーブルの構造

列名	型	説明
pid_id	int	PID制御パラメータのID (主キー)
count	int	制御の試行回数 (主キー)
time	float	時刻
position	float	金属球の位置
angle	float	ビーム角度
mot_out	float	操作量

表3 pid_result テーブルの構造

列名	型	説明
pid_id	int	PID制御パラメータのID (主キー)
count	int	制御の試行回数 (主キー)
rise_time	float	立ち上がり時間 T_r
overshoot	float	最大行き過ぎ量 A_p
overshoot_time	float	最大行き過ぎ時間 T_p
set_time	float	整定時間 T_s
offset	float	定常偏差 e_f
set_length	float	整定時間の長さ T_{sl}

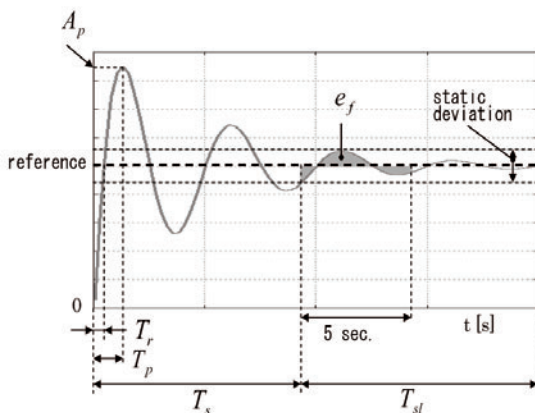


図3 ステップ応答の評価指標

から図3に示す指標に従って抽出した評価指標を記録するテーブルである(表3)。

本実験では以上のテーブルとしたが、これにサンプリング周期 T_s などのパラメータを追加することは容

易にでき、より複雑な条件下で制御システムの状態を観測・記録できる。

3. 実験結果

制御対象の数式モデルと制御仕様から所要の制御帯域が満たせるように各パラメータを設定し、その近傍で比例ゲイン k 、積分パラメータ b 、微分パラメータ a を変化させ、それに対する各ステップ応答から表3のテーブルに示した特徴量をデータベース上に収集した。

図4は、微分パラメータ a を固定して、比例ゲイン k 及び積分パラメータ b を変化させた場合の最大行き過ぎ量 A_p のプロット図である。前述のように制御対象は基本的に2型であるため、この図から分かるように積分パラメータは非常に小さい値で、目標値に偏差なく到達できる。なお、図4、図6、図7では、濃淡の濃い部分が A_p が小さい場合を表している。

図5は、微分パラメータ a を固定して、比例ゲイン k と積分パラメータ b を変化させた場合の定常偏差 e_f のプロット図である。図のように積分パラメータ b が小さい領域で定常偏差が小さい状態が観測できる。また、積分パラメータ b を0に近づけると定常偏差が大きくなる様子も観測できる。濃淡の濃い部分が e_f が小さい場合を表している。

図6は、積分パラメータ b を固定して、比例ゲイン k 及び微分パラメータ a を変化させた場合の最大行き過ぎ量 A_p のプロット図である。微分パラメータ a を大きくするに伴い最大行き過ぎ量 A_p が減少するが、比例ゲイン k を小さくすると最大行き過ぎ量 A_p が増加することが分かる。

図7は、比例ゲインを固定して、積分及び微分パラメータを変化させた場合の最大行き過ぎ量 A_p の分布を求めたプロット図である。

これらの実験結果は、このような単純な制御対象に対しての定性的な説明に合致する。従って、この程度の制御対象に対するPID制御器の最適化は、ステップ応答の結果から容易に導ける。しかし、サンプリング周波数の変化、制御演算時間遅れ、制御対象の高次元化などによりパラメータが増え、PID制御系であっても定性的な解釈は困難となり、最適な制御パラメータを決定することは容易ではなくなる。今回提案したシステムは、そのような場合でもデータベースを拡張することにより、容易に対応できる。

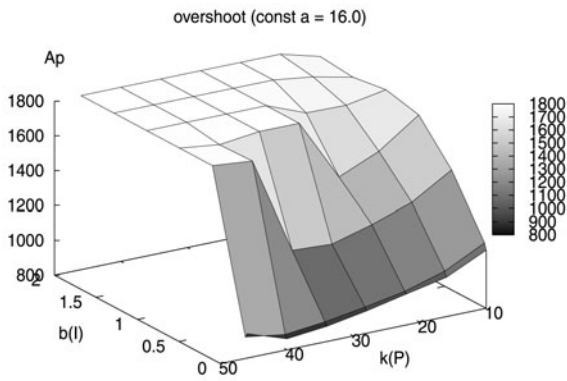


図4 比例ゲイン k , 積分パラメータ b と最大行き過ぎ量 A_p の3次元表示 (目標値:800)

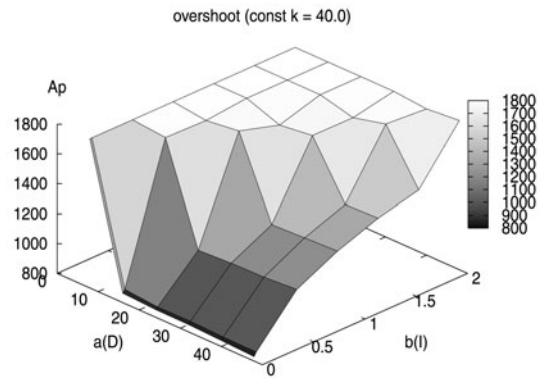


図7 積分パラメータ b , 微分パラメータ a と最大行き過ぎ量 A_p の3次元表示

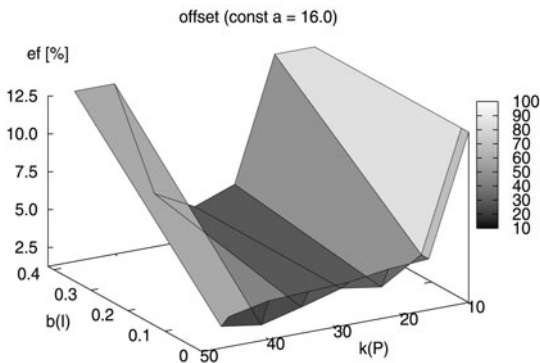


図5 比例ゲイン k , 積分パラメータ b と定常偏差 ef の3次元表示

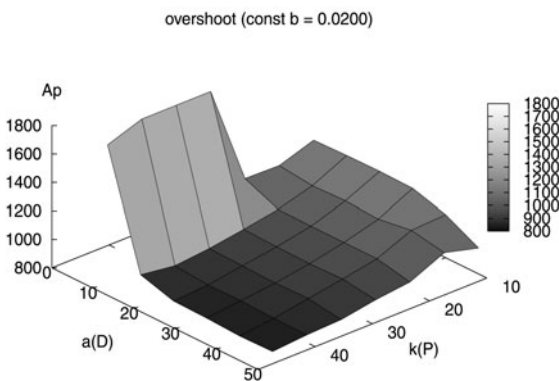


図6 比例ゲイン k , 微分パラメータ a と最大行き過ぎ量 A_p の3次元表示 (目標値: 800)

4. あとがき

ディジタルPID制御系においてステップ応答の特徴パラメータに対する各制御パラメータの分布を観測・記録するシステムを構築した。

制御応答の観測結果から、ステップ応答の設定条件を満たす制御パラメータの分布は、ある領域を有しており、この結果を利用することで、制御応答の特徴パラメータを評価関数とした、PID制御パラメータの最適化が可能と考えられる。

謝辞

本実験システムの開発にあたり、ご協力をいただいた、情報電子工学科第28期生の松田和政君に感謝いたします。

本研究の一部は、中国電力技術研究財団の試験研究助成のもとで行われた。

文献

- 1) A. Visioli: Practical PID Control, Springer-Verlag, pp.35-91, (2006)
- 2) 須田信英: PID制御, 朝倉書店 pp.9-38, (1992)
- 3) 山田健仁, 百田正広: Linux を利用した遠隔制御基礎実験システムの開発, 論文集「高専教育」, 第27号, pp.13-18, (2004)
- 4) 山田健仁, 芳川健: リアルタイムディジタル制御とネットワーク技術の修得を目的とした遠隔制御実験

システムの検討, 徳山高専研究紀要, 第 30 号, pp. 13-20, (2006)

5) 中村政俊, 香田英之, 久良修郭: サーボ系による連続経路制御のサンプル値制御における適正サンプル周波数の決定法, 計測自動制御学会論文集, Vol. 28, No. 5, pp. 649-651, (1992)

6) 石綿陽一: SMP カーネルに基づく ART-Linux の安定化と実時間処理性能の測定, 計測自動制御学会システムインテグレーション部門講演会講演論文集, pp. 417-418, (2002)

7) SQLite home page : <http://www.sqlite.org/>
(2007. 8. 24 受理)

