

4. PID による温度制御

電気電子工学科 X 年 X 組 番号 氏名

2025 年 12 月 4 日

1 目的

本実験では、ワンボードマイコンを用いたヒーターの温度制御系を対象として、PID 制御の実装およびパラメータ調整を行った。各制御要素（比例・積分・微分）が応答特性に与える影響を定量的に把握するとともに、ステップ応答法による最適な PID パラメータの設計手法を習得することを目的とした。

2 原理

PID 制御は、フィードバック制御において最も一般的な手法の一つであり、偏差に対する比例 (Proportional), 積分 (Integral), 微分 (Derivative) の 3 要素を用いて操作量を決定する。

2.1 PID 制御の伝達関数

制御対象への操作量 $u(t)$ は、目標値と出力値の偏差 $e(t)$ を用いて式 (1) のように表される。

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (1)$$

これをラプラス変換し、伝達関数 $K_C(s)$ として表すと式 (2) となる。

$$K_C(s) = K_P + \frac{K_I}{s} + K_D s = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (2)$$

ここで、 K_P は比例ゲイン、 K_I は積分ゲイン、 K_D は微分ゲインである。また、 $T_I (= K_P / K_I)$ は積分時間、 $T_D (= K_D / K_P)$ は微分時間と定義される。この制御系のブロック図を図 1 および図 2 に示す。

各パラメータが制御系に及ぼす影響は以下の通りである^[1]。

比例動作 (K_P) 偏差に比例した操作量を出力する。 K_P の増加に伴い応答速度は向上し定常偏差は減少するが、過大なゲインは振動的な挙動を招く。

積分動作 (K_I または T_I) 偏差の積分値に応じた操作量を出力する。低周波成分のゲインが無限大となるため、定常偏差を完全に除去できる。

微分動作 (K_D または T_D) 偏差の変化率に応じた操作量を出力する。未来の偏差を予測して補償するため、速応性の向上と振動抑制（ダンピング）に寄与する。

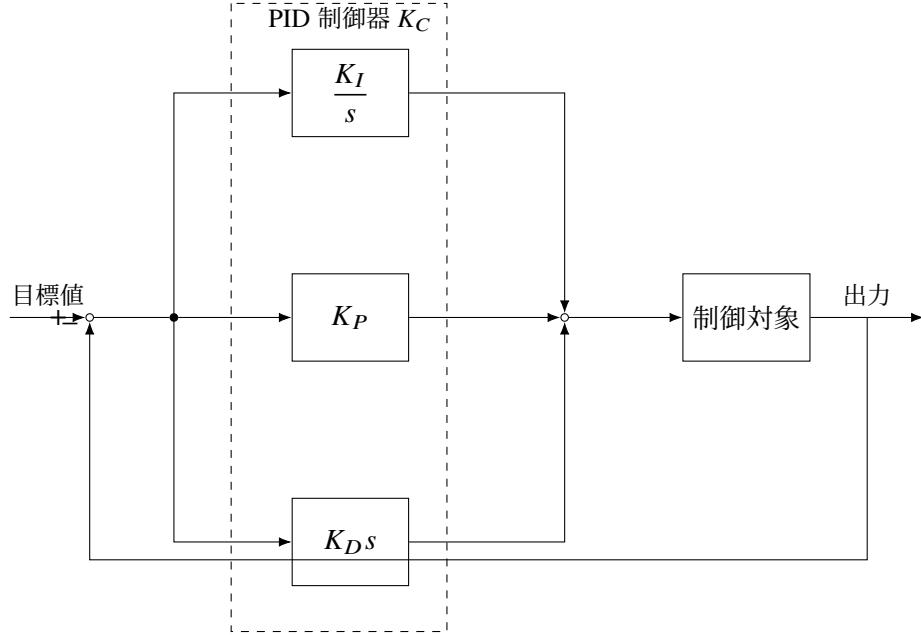


図 1 PID 制御のブロック図（並列型）

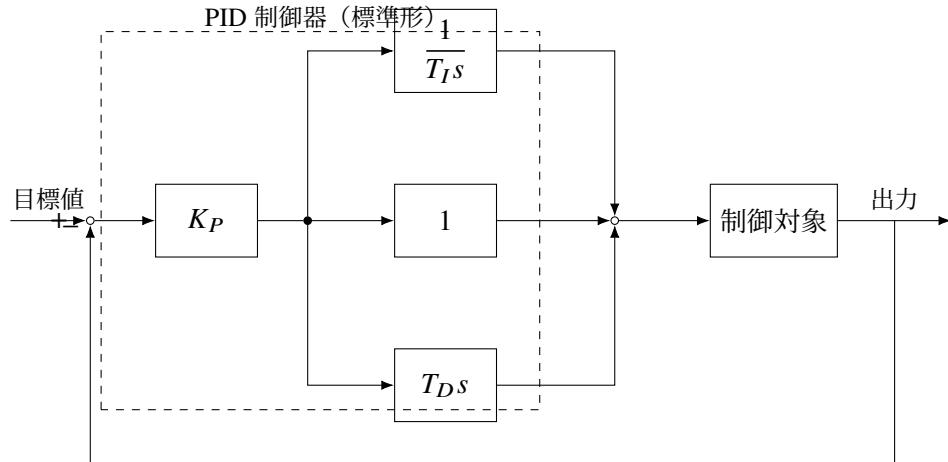


図 2 PID 制御のブロック図（標準型）

2.2 パラメータ調整法（ステップ応答法）

制御対象が開ループにおいて安定であり、式(3)のような1次遅れ系とむだ時間要素で近似できる場合、ステップ応答法によるパラメータ調整が可能である。

$$G_P(s) = \frac{K}{1 + sT} e^{-Ls} \quad (3)$$

図3に示すステップ応答波形から、むだ時間 L および時定数 T （または変曲点の接線から求めた傾き R ）を読み取り、表1に基づき各パラメータを設定する。

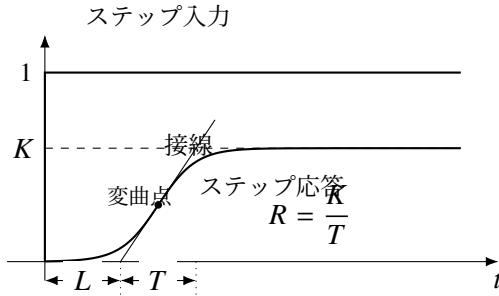


図3 ステップ応答法におけるパラメータの読み取り

表1 ステップ応答法によるPIDパラメータ調整則

パラメータ	K_P	T_I	T_D
P制御	$\frac{1}{RL}$	—	—
PI制御	$\frac{0.9}{RL}$	$3.3L$	—
PID制御	$\frac{1.2}{RL}$	$2L$	$0.5L$

3 実験方法

3.1 使用機器

本実験で使用した主要機器を表2に示す。

表2 使用機器一覧

品名	型番	備考
温度制御実習装置	KENTAC3522S	ヒーター、温度センサ内蔵
制御用PC	—	Windows OS, PidMonitorインストール済

3.2 実験手順

ヒーターの温度を制御量とし、目標温度を100°Cに設定して以下の実験を行った。

3.2.1 ON・OFF制御

制御用ソフトウェア(PidMonitor)のAutoモードを解除し、手動によるON・OFF制御を行った。MV(操作量)を50.0%に設定し、PV(現在温度)が100°Cに到達するまで加熱した。100°Cを超えた時点でMVを0.0%とし、下回った時点で再度50.0%に戻す操作を繰り返し、その際の温度挙動(ハンチング)を確認した。

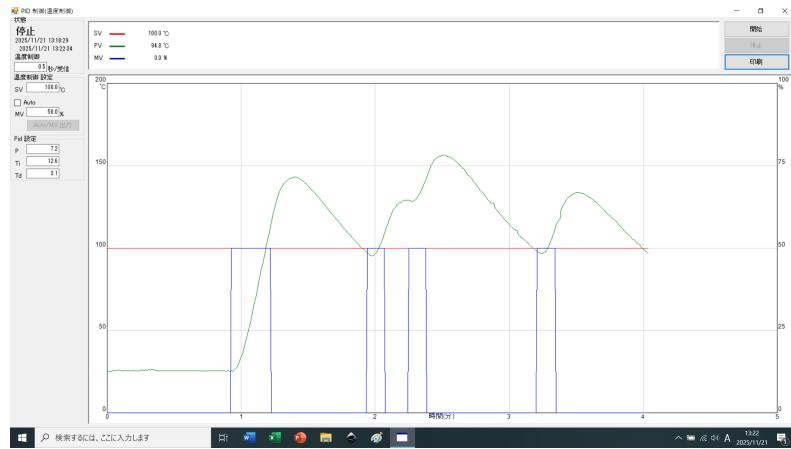
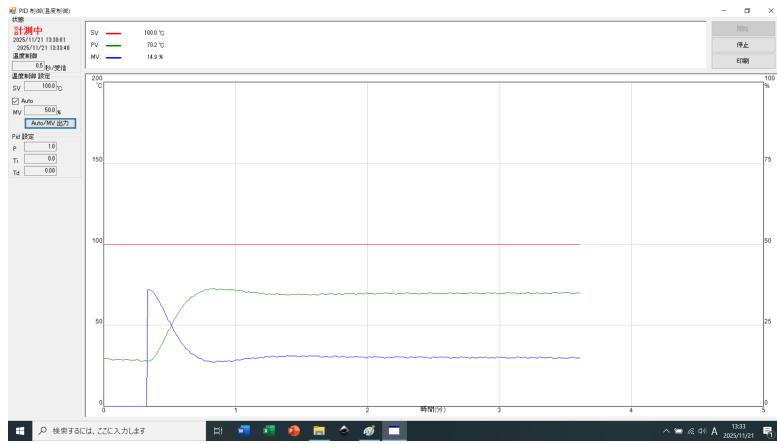


図 4 手動 ON/OFF 制御時の温度挙動

3.2.2 P 制御

積分時間 $T_I = 0$ (実験装置仕様), 微分時間 $T_D = 0$ とし, 比例制御のみを行った. 比例ゲイン K_P を 1, 5, および 100 に設定した場合の温度変化を測定した. 以下にそれぞれの応答波形を示す.



(a) $K_P = 1$



(b) $K_P = 5$

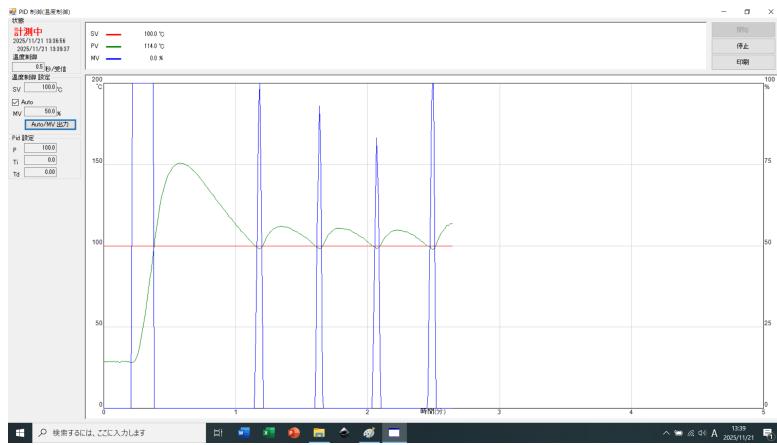


図 5 P 制御の比較 (K_P の違い)

その後、3 分以内に目標温度 $\pm 10^{\circ}\text{C}$ の範囲に収束する最適な $K_P = 5$ を試行錯誤により決定した。

3.2.3 PI 制御

K_P を 5 に固定し, $T_D = 0$ として PI 制御を行った. 積分時間 T_I を 1, 48, および 100 に設定した場合の応答を確認した. 以下に代表的な波形を示す.

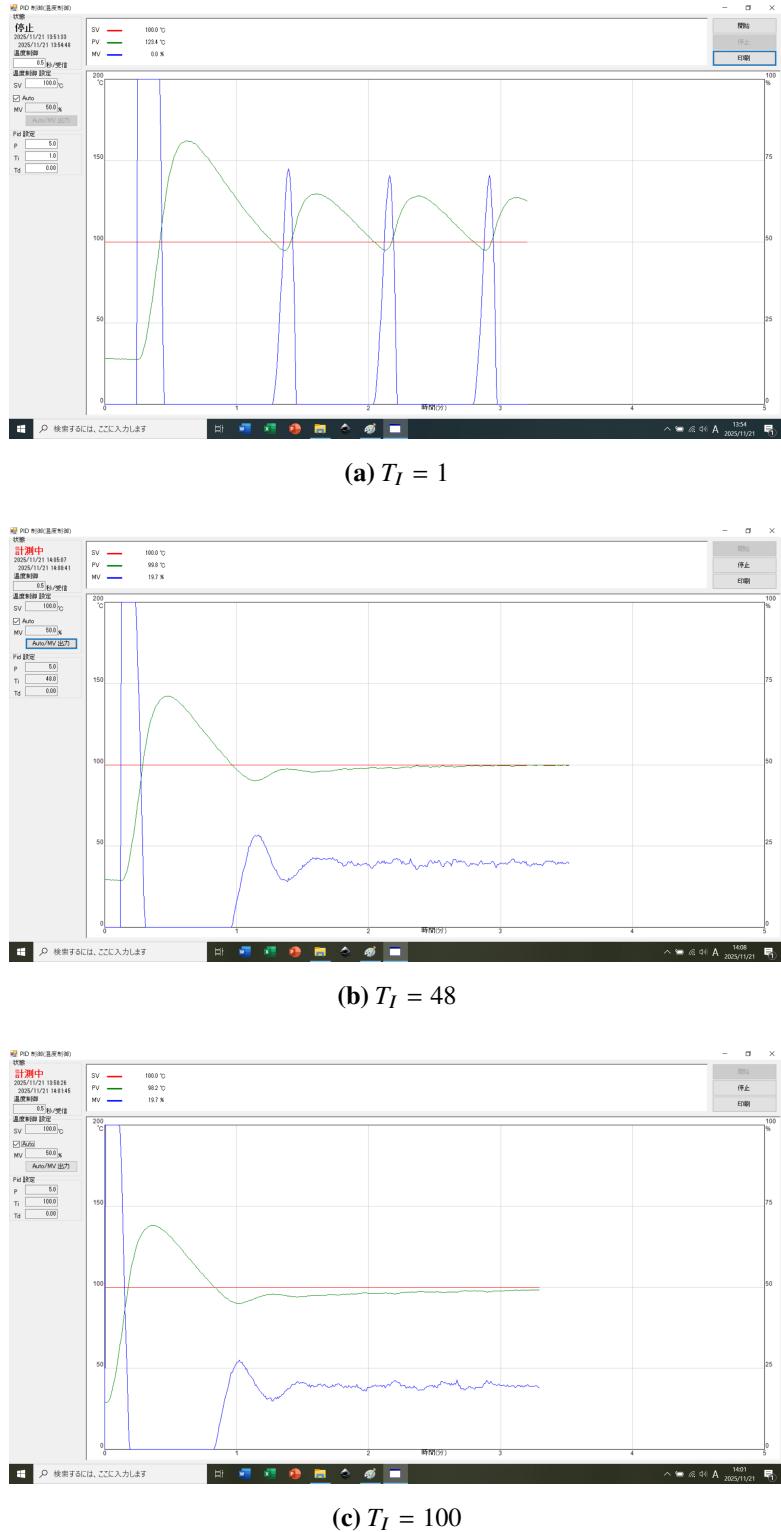


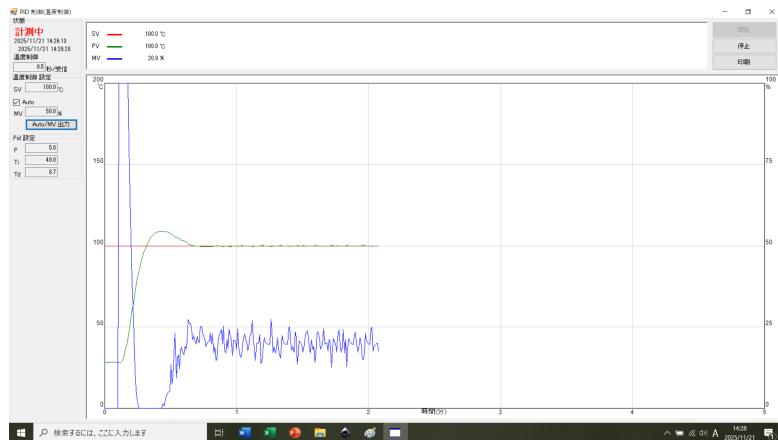
図 6 PI 制御における積分時間 T_I の影響 ($K_P = 5$ 固定)

3.2.4 PID 制御

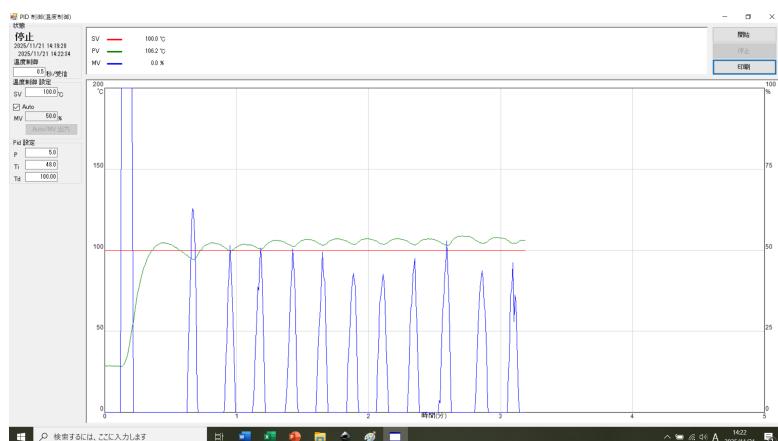
K_P および T_I をそれぞれ 5 および 48 に固定し、PID 制御を行った。微分時間 T_D を 1, 8.7, および 100 に設定した場合の応答を確認した。以下に代表的な波形を示す。



(a) $T_D = 1$



(b) $T_D = 8.7$



(c) $T_D = 100$

図 7 PID 制御における微分時間 T_D の影響 ($K_P = 5, T_I = 48$ 固定)

その後、2分以内に目標温度 $\pm 0.5^{\circ}\text{C}$ の範囲に収束する最適な $T_D = 8.7$ を決定した。

3.2.5 ステップ応答法による同定

ステップ応答法を用いて制御対象のパラメータ同定を行った。

1. ヒーター温度が室温で安定している状態から、MVを10%～20%の範囲でステップ状に入力した。
2. 温度が定常状態に達した際の入力変化幅 $U [\%]$ および温度変化幅 $Y [\%]$ を式(5), (4)により求めた。
3. 定常ゲイン K および、図3に基づく L, T, R を算出した。

$$U = (\text{最終入力} - \text{初期入力}) \quad (4)$$

$$Y = \frac{(\text{最終温度} - \text{初期温度})}{200} \times 100 \quad (5)$$

$$K = \frac{Y}{U} \quad (6)$$

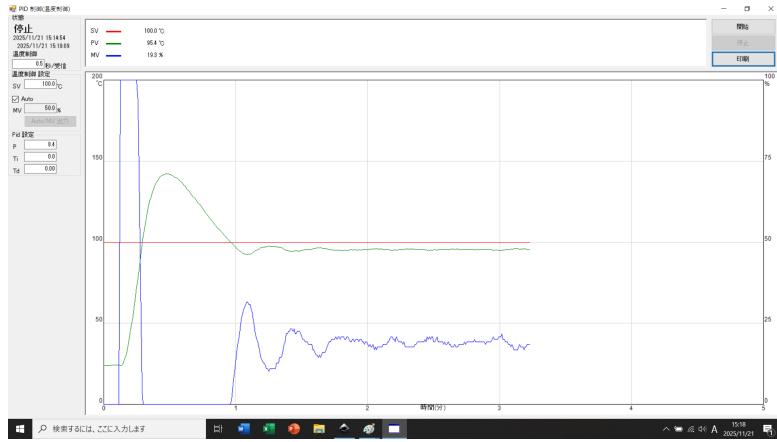
$$R = \frac{K}{T} \quad (7)$$

得られたパラメータを基に、表1を用いて K_P, T_I, T_D を算出し、これを用いた制御実験を行った。以下にステップ応答法で算出したパラメータを示す。

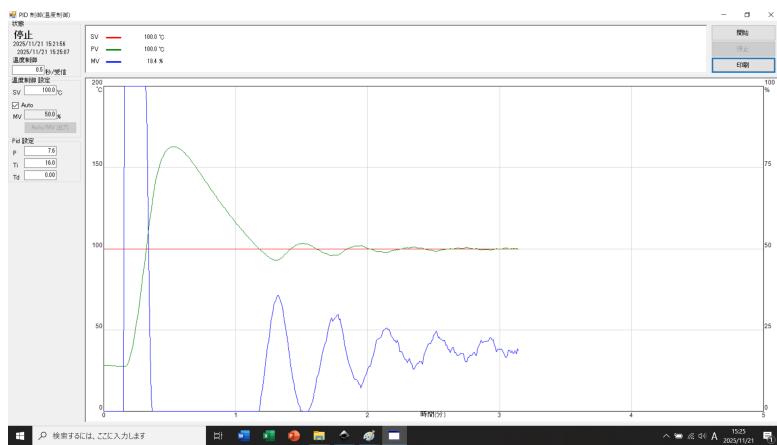
表3 ステップ応答法で算出したPIDパラメータ

制御タイプ	K_P	T_I	T_D
P制御	8.4	—	—
PI制御	7.6	16	—
PID制御	10.1	9.7	2.42

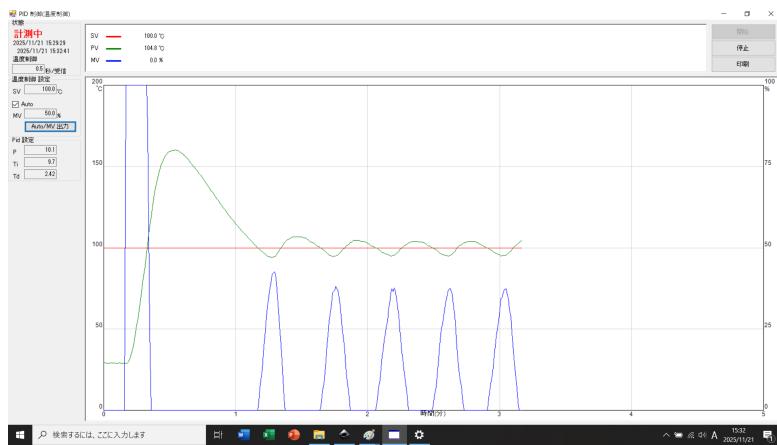
これらのパラメータを用いた制御実験の結果を図に示す。



(a) P 制御 ($K_P = 8.4$)



(b) PI 制御 ($K_P = 7.6, T_I = 16$)



(c) PID 制御 ($K_P = 10.1, T_I = 9.7, T_D = 2.42$)

図 8 ステップ応答法で算出したパラメータを用いた制御実験結果

4 結果および考察

4.1 各制御手法における応答特性の比較

本実験で観測された P 制御, PI 制御, PID 制御の応答波形を図 5, 図 6, 図 7 に示す。P 制御では, ゲイン K_P を大きくすることで立ち上がり時間が短縮されたが, 残留偏差 (オフセット) が確認された。図 5 を観察すると, $K_P = 1$ は立ち上がりが緩やかで目標到達に時間を要する一方, $K_P = 100$ では立ち上がりは著しく速いが振動や過渡応答の大きさ (オーバーシュート) が増大している。これは原理上, 積分要素を持たないため定常偏差を除去しきれないと, 高ゲインによる系のダンピング不足が重なった結果である。PI 制御導入後は, 積分項により定常偏差が解消され, 目標温度付近における定常誤差が小さくなった (図 6)。 T_I が小さい ($T_I = 1$) 場合は積分作用が強く働き, 偏差の補償が速い一方で, 過渡的にオーバーシュートが大きくなる傾向が見られた。 $T_I = 48$ は収束速度と過渡応答のバランスに優れており, 実験上の要件 (3 分以内に $\pm 0.5^{\circ}\text{C}$) を満たす最適解となった。PID 制御では, 微分作用により過渡応答のオーバーシュートと振動が抑えられ, 系の安定化に寄与した (図 7)。ただし, 過度な微分 ($T_D = 100$) は測定ノイズを增幅させるため, 操作量の変動が増え、むしろ収束性を損なう場合があった。今回の試行錯誤の結果, $T_D = 8.7$ はノイズ耐性と応答性の双方において良好な折衷点であり, 最終採用値となった。

4.2 パラメータ設定値による挙動の変化

K_P, T_I, T_D の各値を極端に変化させた際 (1 および 100) の挙動について考察する。本節では実験で観測した各パラメータの影響を定性的にまとめる。

- 比例ゲイン K_P の影響: K_P を増すと立ち上がりは速くなるが, あまり大きくすると系の位相余裕が低下し振動や発散を起こす。図 5 より $K_P = 100$ は著しい振動を伴い, 実用面では推奨できない。
- 積分時間 T_I の影響: T_I が短い (積分作用が強い) と定常偏差は早く解消されるが, 積分作用のリセット (anti-windup) を伴わない場合は操作量の過渡変動が大きくなる。一方 T_I が長いと偏差の解消は緩慢になり定常偏差が残る場合がある。実験では $T_I = 48$ が良好であった。
- 微分時間 T_D の影響: T_D を増加させると応答の振動が抑えられるが, 測定ノイズも増幅されるため操作量に高周波ノイズが現れることがある (図 7)。 T_D が過大な場合は制御則の効きが不安定化する可能性があるため注意が必要である。

これらの挙動は, 実験系が理想的な 1 次遅れ+遅延のモデルより非線形性やサチュレーション (MV の 0 – 100%)、センサ精度の限界、環境熱損失などの影響を受けるためである。そのため理論的に導出した最適値と実装上の最適値に差が生じることがある。

4.3 ステップ応答法による設計値の妥当性

ステップ応答法により算出したパラメータを用いた制御結果は, 試行錯誤法で得られた結果と比較して積極的である傾向が確認された。表 4 に示すように, 実験での最終採用値は設計値よりも小さな比例ゲインと短い積分時間, および小さめの微分時間が採用されている。これにより立ち上がり時間は長くなり定常偏差は大きくなつたが, 低ゲインや小さな微分時間はオーバーシュートや測定ノイズ

の影響を減少させるため、適切な安定性を確保することが重要である。表4に、ステップ応答法で算出した設計値（理論）と、試行錯誤で最終採用した実験値（実践）を示す。

表4 ステップ応答法での設計値と試行錯誤での最終採用値の比較

パラメータ	設計値（ステップ応答法より）	実験での最終採用値
K_P	10.1	5
T_I	9.7	48
T_D	2.42	8.7

比較の結果、ステップ応答法で算出した値は実機での挙動に対して保守的であった。主な原因は以下の通りである： - モデル近似の誤差：実機は理想的1次遅れ+遅延よりも非線形であり、ヒーターの熱伝達・放熱特性やセンサ遅延が影響する。 - 入力の量子化・飽和：MV制約(0–100%)およびヒーターの非線形な出力特性により、理論ゲインがそのまま適用できないことがある。 - 測定ノイズとノイズ処理：実験データにノイズが含まれており、微分項の効果が理論通りに反映されない場合がある（微分はノイズを增幅する）。

したがって、ステップ応答法は初期値生成に有用であるものの、最終的な運用値は実機を用いた微調整（試行錯誤）で決定することが望ましい。特に、本装置のようにアクチュエータの非線形性やセンサノイズが大きい場合は、試験条件を限定して複数回の実験を行い、平均的な応答を基に調整することが重要である。

5 報告事項

以下の制御対象 $G(s)$ に対し、ステップ応答法（表1）を用いてPIDパラメータを設計した結果を示す。

$$G(s) = \frac{5}{2s + 4} e^{-3s} \quad (8)$$

まず、式を標準形 $K/(1+sT) \cdot e^{-Ls}$ に変形すると、

$$G(s) = \frac{1.25}{1 + 0.5s} e^{-3s} \quad (9)$$

となる。これより $K = 1.25, T = 0.5, L = 3$ が得られる。したがって、 $R = K/T = 2.5$ となり、PID制御の各パラメータは以下のように算出される。

$$\begin{aligned} K_P &= \frac{1.2}{RL} = \frac{1.2}{2.5 \times 3} = 0.16 \\ T_I &= 2L = 6.0 \\ T_D &= 0.5L = 1.5 \end{aligned}$$

参考文献

- [1] 豊橋技術科学大学・高等専門学校制御工学教育連携プロジェクト 編:『制御工学』, 実教出版, pp.146-147, 2012.