富山高等専門学校　専攻科

平成 25 年度

特 別 研 究 報 告

研究題目　LED を使った高安定光源の開発

指導教員　由井　四海

提出者　制御情報システム工学　専攻

宮田　大暉

平成 26 年 3 月 10 日　提出

目次

1. 研究目的

材料やガスなどを分析するときに分光法が多く利用されている。分光法とは対象物に光を照射して、透過光あるいは反射光を測定することで、その性質などを特定する方法である。分光法による分析を行うとき、対象物に照射する光の強度とスペクトルが安定していなければ、対象物の性質を特定することができなくなる。そのため、光の強度とスペクトルが安定した高安定光源が必要である。高安定光源には白熱電球やハロゲン電球が用いられているが、これらの電球は電流によりフィラメントを加熱して光を放射しているため、フィラメントが蒸発して小さくなり、光の強度が低下してしまう。また、これらの電球は400nmの青色領域から700nm以上の近赤外領域まで広く光の強度が分布する発光スペクトルをもち、この中で赤外線の強度が強く、この熱を効率よく逃がすために、光源装置に十分な放熱対策を施す必要がある。これに対して発光ダイオード（LED）はフィラメントを加熱して光を放射する方法ではなく、電気的な方法で電子を励起した後に、電子と正孔が再結合することで発光しているので、白熱電球やハロゲン電球より経年変化が小さく、光強度の低下が遅くなる。そのため、発光ダイオード(LED)は長時間使用する高安定光源として有効である。

本研究はLEDを使用して、強度およびスペクトルの波長変化率が2%以下になる高安定光源を開発することを目的とする。

2. 研究内容

2.1. LED の特性

図 2.1 に LED の強度と温度の関係を示す。これより LED は温度によって強度が変化することがわかり、強度を安定化するためには LED の温度を一定にする必要がある。図 2.2 に LED の強度と電流の関係を示す。これより LED は電流によって強度が変化することがわかり、強度を安定化するためには LED に流れる電流も一定にする必要がある。本研究ではピーク波長が 625nm の赤色パワーLED である OptoSupply Limited 製 OSR5XME1C1E を使用した。

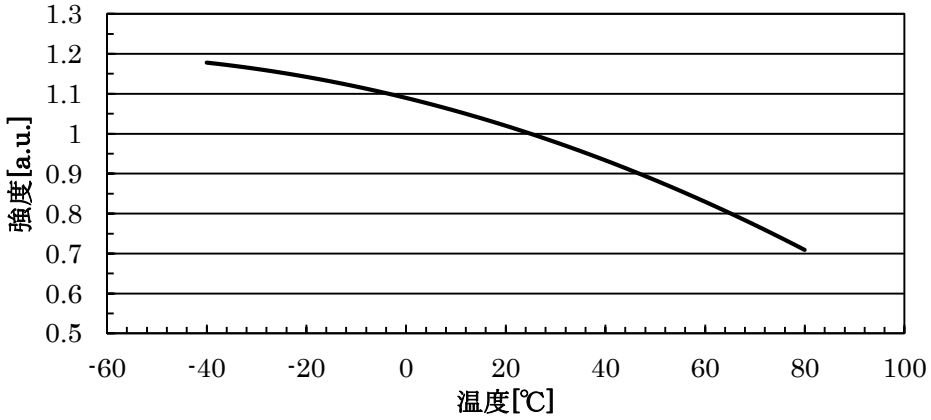

図 2.1　LED の温度-強度特性

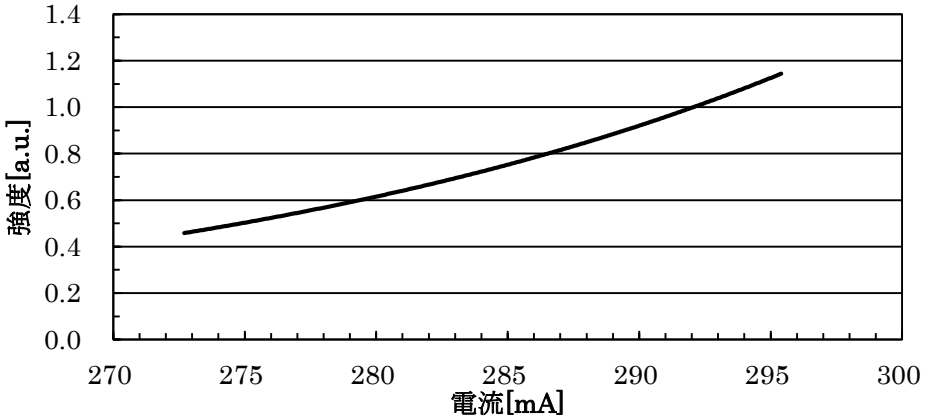

図 2.2　LED の電流-強度特性

2.2. 実験装置の構成

図 2.3 に実験装置の構成を示す。実験装置は大きく、光源装置と測定系にわけられる。光源装置では LED の電流を制御し、LED の強度とスペクトルは LED の温度により変化するので LED の温度を制御した。PD の出力電流が温度により変化することで LED の電流を制御することができなくなるので PD の温度を制御した。各制御にはアナログ入力の分解能が 10bit のマイコン(Arduino)を使用した。

測定系では LED の強度を PD で、スペクトルを分光器で測定し、PC に記録した。

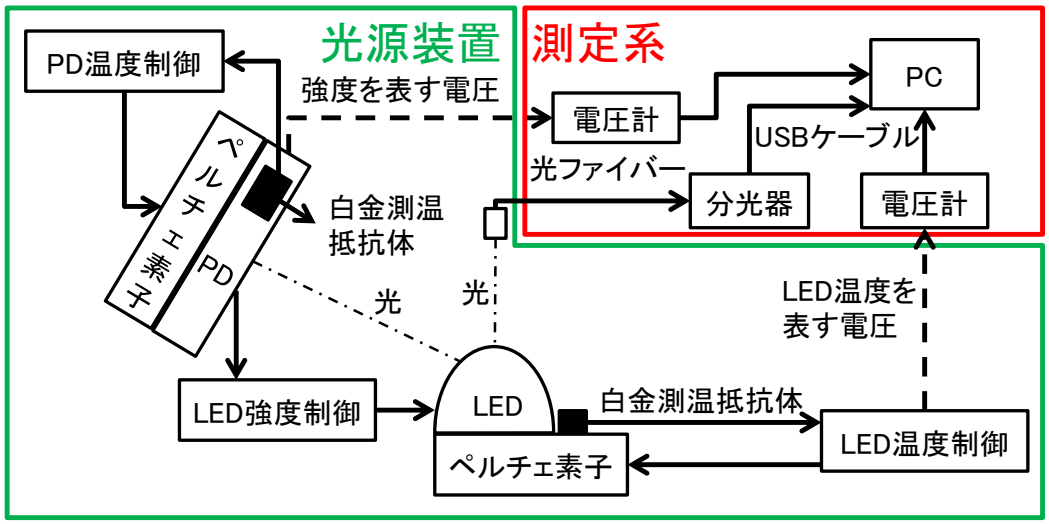

図 2.3　実験装置の構成図

2.2.1. 光検出器(PD)

図 2.4 に PD の等価回路を示す。この等価回路から出力電流I_Oを求めると（2.1）式となる。ここでI_Lは入射光による発生電流 、I_Dはダイオード電流、I_Sは PD の逆方向飽和電流、I′は並列抵抗電流、eは電子の電荷、V_Dはダイオード両端の電圧、k はボルツマン定数、T は PD の絶対温度である。

$$I_O = I_L - I_D - \mathrm{I}' = I_L - I_S\left\{\exp\left(\frac{eV_D}{kT}\right) - 1\right\} - \mathrm{I}' \qquad (2.1)$$

（2.1）式より、入射光強度が一定であっても PD の出力電流は温度によって変化するため、温度を一定にする必要がある。本研究では波長 625nm で受光感度 0.33 A/W の PD である浜松ホトニクス製 S2387-1010R を使用した。

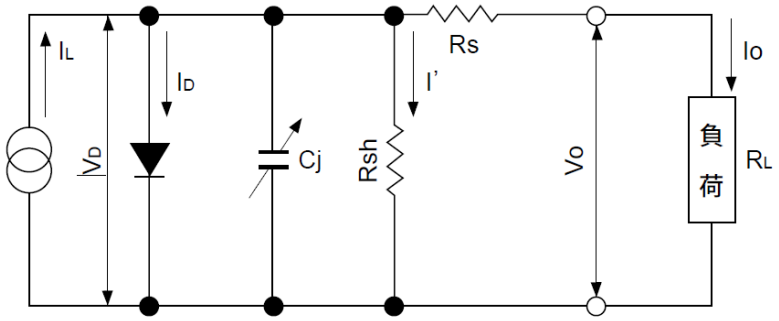

図 2.4　PD の等価回路

2.2.2.　白金測温抵抗体

金属の電気抵抗が温度変化に対して変化する性質を利用した測温抵抗体の一種で、温度特性が良好で経年変化が少ない白金を測温素子に用いたセンサである。抵抗値と温度の関係が対数的ではなく、ほぼ直線的に変化して抵抗値から電圧への変換が簡単であるために、この抵抗体を使用することにした。温度検出に温度係数が 0.39Ω/℃の Heraeus 製 FK 222-100-A を使用した。

2.2.3.　ペルチェ素子

ペルチェ素子は二種類の金属の接合部に電流を流すと、片方の金属からもう片方へ熱が移動するというペルチェ効果を利用した素子である。この効果により、直流電流を流すと一方の面で吸熱して、その熱が移動することで反対面が発熱する。電流の流れる向きを変えることで、熱の移動を逆流させることができることから、加熱および冷却が可能である。また、ヒートポンプに比べて装置が小型化でき、振動を発生しないために LED と PD の温度制御にペルチェ素子を使用することにした。LED の温度制御に最大吸熱量が 53.3W の HB Electronic Components 製 TEC1-12706 を使用して、PD の温度制御に最大吸熱量が 44.5W の HB Electronic Components 製 TEC1-12705 を使用した。

2.3. PID 制御

PID 制御とは、目標値と出力値のずれである偏差に対して比例（P 動作）、積分（I 動作）、微分（D 動作）の三つの動作を組み合わせて、制御の対象となる物に加える操作量を決めるフィードバック制御のことである。

比例制御（P 制御）は（2.2）式に示すように偏差 *e* に比例ゲイン Kp を掛けた値を操作量として制御を行う方法である。

$$K_p e(t) \tag{2.2}$$

（2.2）式で比例ゲインが小さいと偏差が 0 に近づいたときに出力値が小さくなり、目標値と出力値に一定のずれ（定常偏差）が生じる。逆に比例ゲインが大きいと出力値が振動的になる。そのため比例制御のみでは目標値に収束することができない。

積分制御（I 制御）は（2.3）式に示すように偏差 *e* の積分値に積分ゲイン Ki を掛けた値を操作量として制御を行う方法である。

$$K_i \int e(t)dt \tag{2.3}$$

（2.3）式で偏差を積分した値により操作量が決まるため、偏差があれば偏差を 0 にしようと操作量が変化する。比例制御では定常偏差が発生したが、積分制御を加えることで偏差が無くなるまで制御を行うので定常偏差が無くなる。

微分制御（D 制御）は（2.4）式に示すように偏差 *e* の時間変化分である微分値に微分ゲイン Kd を掛けた値を操作量として制御を行う方法である。

$$K_d \frac{de(t)}{dt} \tag{2.4}$$

PI 制御の I 動作は出力値を目標値に近づけるためにはある程度の時間が必要となる。そのため（2.4）式により、D 制御は偏差の変化が大きくなるほど出力が大きくなるので、PI 制御で補えない急激な出力値の変化に対応して、早く目標値に収束することができる。よって、D 制御はフィードバック制御系の制御応答の特性を改善する働きがある。

本研究では、電流制御に PI 制御を用い、温度制御に PID 制御を用いた。電流制御の PID 制御と PI 制御の結果を比較すると PI 制御の結果の方が目標強度に到達した後に強度が安定したため、電流制御に PI 制御を用いることにした。

2.4.　LED の電流制御

図 2.5 に LED 強度測定の構成を示す。LED から放射する光のみを PD で受光するために装置内を黒くして光が乱反射しないようにした。

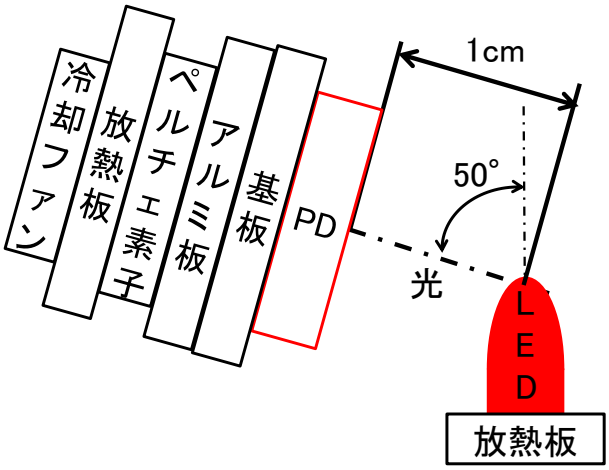

図 2.5　LED 強度測定の構成

図 2.6 に LED 電流制御のブロック線図を示す。サンプリング周波数が 100kHz の マイコンで帰還電圧を A/D 変換して目標電圧との差を使い、PID 演算した結果を 500Hz の PWM 信号で出力した。マイコンの PWM 出力は 8bit であるためデューディー比は 0.4%刻みで変えることができ、PWM 信号の振幅は 5V である。PWM 信号のデューディー比を 0.4%変えることで、LED に流れる電流が 0.1mA 変えられるようにするために、反転加算回路で PWM 信号を 0.05 倍に減衰させた。また、LED に 280mA の電流を流すために PWM 信号に直流電圧 2.8V を加算した。この加算信号は反転増幅したために負電圧が含まれる。マイコンの入力電圧範囲は 0～5V であるため、負電圧を多重帰還型のローパスフィルタに通すことで、正電圧の平滑化した制御電圧を出力した。次に制御電圧を電流に変換して LED に流した。

フィードバック部の強度検出回路では PD で発生した電流を電圧に変換してから、差動増幅回路で増幅し、この電圧を強度として測定した。このとき、マイコンの最大入力電圧$V_{A/D} = 5V$、量子化ビット数$\mathrm{n} = 10\mathrm{bit}$であるため A/D 変換の刻み幅$\Delta V_{A/D}$[V]は（2.5）式となる。

$$\Delta V_{A/D} = \frac{V_{A/D}}{2^n - 1} \approx 4.89 \times 10^{-3}[V] \qquad (2.5)$$

そのため、差動増幅回路では刻み幅$\Delta V_{A/D}$が 4.9mV より大きい値に検出電圧を増幅した。次

に増幅電圧をカットオフ周波数が PWM 信号の周波数 500Hz の約 10%である 48Hz の 2 次の RC ローパスフィルタ（LPF）に通してからマイコンに入力した。

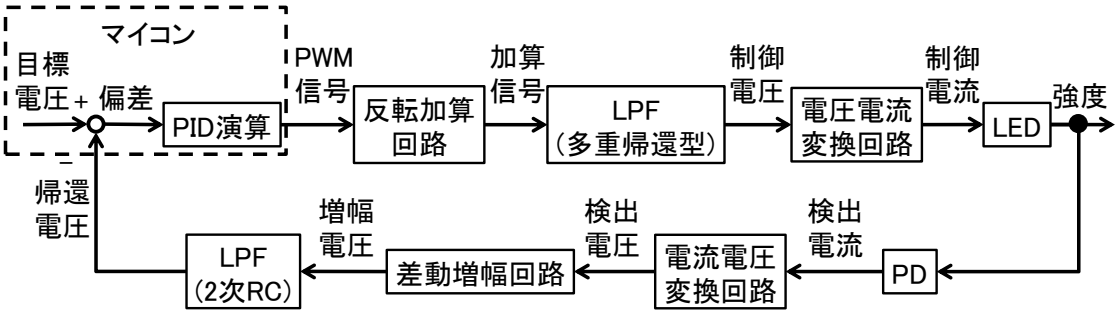

図 2.6　LED 電流制御のブロック線図

図 2.7 に多重帰還型ローパスフィルタのボード線図を示す。カットオフ周波数が PWM 信号の周波数 500Hz の約 10%である 48Hz で、減衰域の傾きは-40dB/dec である。図 2.8 に 2 次 RC ローパスフィルタのボード線図を示す。カットオフ周波数は 48Hz で、減衰域の傾きは-40dB/dec である。

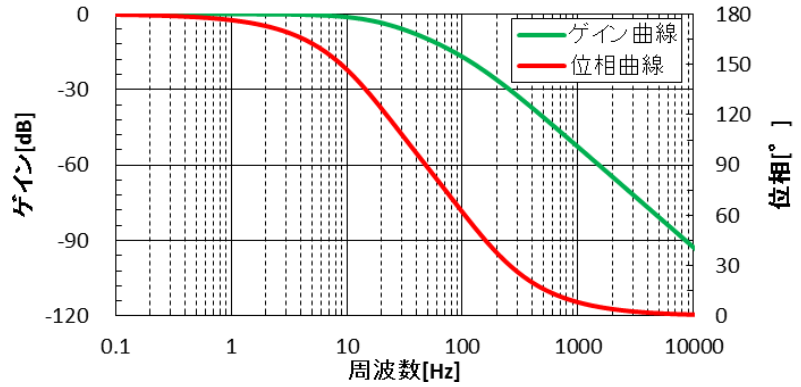

図 2.7　多重帰還型のローパスフィルタのボード線図

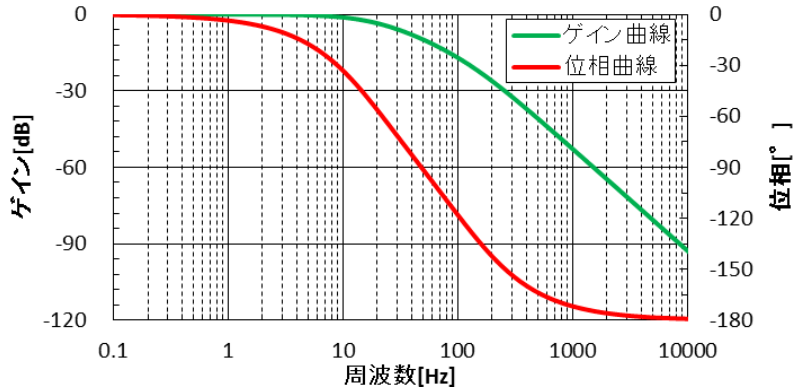

図 2.8　2 次 RC ローパスフィルタのボード線図

2.5.　LED と PD の温度制御

図 2.9 に LED 温度測定の構成を示す。LED を放熱板チップにはんだ付けし、LED のリード部分に白金測温抵抗体を接続させて、LED の温度を測定した。放熱板チップとアルミ板、アルミ板と放熱板をねじで止めて、ペルチェ素子などの各部品の接合面にはシリコングリスを塗った。ペルチェ素子の放熱側に放熱板と冷却ファンを取り付けてペルチェ素子を放熱させた。

図 2.10 に PD 温度測定の構成を示す。PD の温度は LED と同様の方法で制御した。PD の側面に白金測温抵抗体を接続させて PD の温度を測定した。PD とアルミ板は基板を介してねじ止めして、接合面にシリコングリスを塗った。

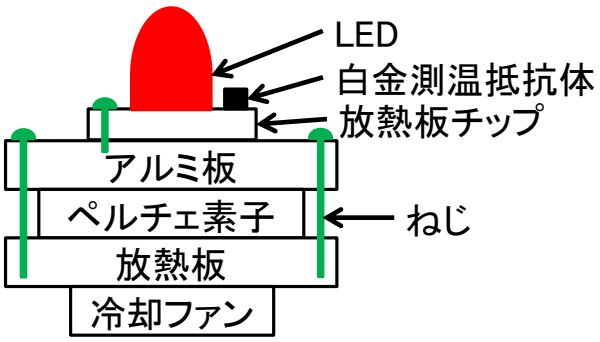

図 2.9　LED 温度測定の構成

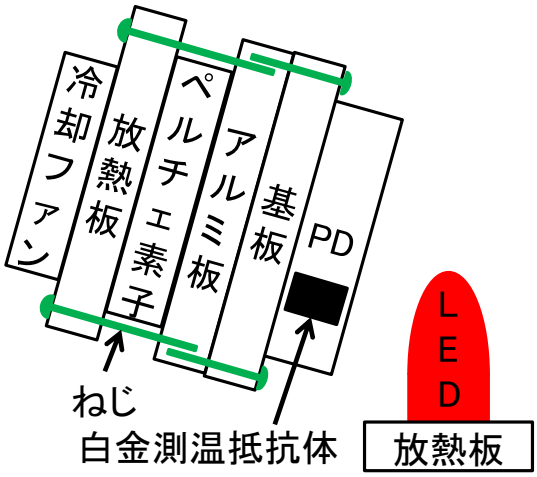

図 2.10　PD 温度測定の構成

図 2.11 に LED と PD 温度制御のブロック線図を示す。マイコンで帰還電圧を A/D 変換して目標電圧との差を使い、PID 演算した結果を 500Hz の PWM 信号で出力した。PWM 信号を H ブリッジ回路に加えて、ペルチェ素子に流れる電流値と電流の流れる向きを変化させて LED と PD を加熱または冷却して温度を制御した。

フィードバック部の温度検出回路では LED と PD の温度で白金測温抵抗体の抵抗値が変化するので、白金測温抵抗体に 1mA の定電流を流して抵抗値を電圧に変化させた。このとき、温度T [℃]、白金測温抵抗体の温度係数ΔR [Ω/℃]、0℃のときの抵抗R_0 [Ω]、抵抗体に流れる電流I [A]、定電流回路の電流検出抵抗R_d [Ω]とすると検出電圧V_d [V]は（2.6）式となる。

$$V_d = (\Delta \mathrm{RT} + R_0)\mathrm{I} + R_d I$$
$$= \Delta \mathrm{RIT} + (R_0 + R_d)\mathrm{I} \quad (2.6)$$

(2.6) 式で発生するオフセット電圧$V_{offset} = (R_0 + R_d)\mathrm{I} = 2.8\mathrm{V}$ を差動増幅回路で引いた上で、この電圧を増幅した。このとき、差動増幅回路の増幅度A_v [-]とすると増幅電圧V_A [V]は（2.7）式となる。

$$V_A = A_v(V_d - V_{offset})$$
$$= A_v \Delta \mathrm{RIT} \quad (2.7)$$

増幅電圧をカットオフ周波数が 48Hz の 2 次の RC ローパスフィルタでノイズを除去した後にマイコンに入力して A/D 変換を行った。温度の測定範囲を 0℃～35.8℃に決めて、マイコンの入力電圧V_Iの範囲が 0V～5V であることから差動増幅回路の増幅度A_vは（2.7）式より、

$$A_v = \frac{V_I}{\Delta \mathrm{RIT}} \quad (2.8)$$

（2.8）式となり、$V_I = 5V$、$T = 35.8℃$、$\Delta \mathrm{R} = 0.388\,\Omega/℃$、$\mathrm{I} = 1 \times 10^{-3}\mathrm{A}$を代入すると$A_v \approx 360$となるので差動増幅回路の増幅度は 360 とした。

（2.5）式から、最大測定温度$T_{A/D} = 35.8℃$、量子化ビット数$\mathrm{n} = 10\mathrm{bit}$であるため、温度の刻み幅$\Delta T_{A/D}$[℃]は（2.9）式となる。

$$\Delta T_{A/D} = \frac{T_{A/D}}{2^n - 1} \quad (2.9)$$
$$\approx 3.50 \times 10^{-2}[℃] = 0.0350[℃]$$

よって、温度は 0.0350℃刻みで測定できる。

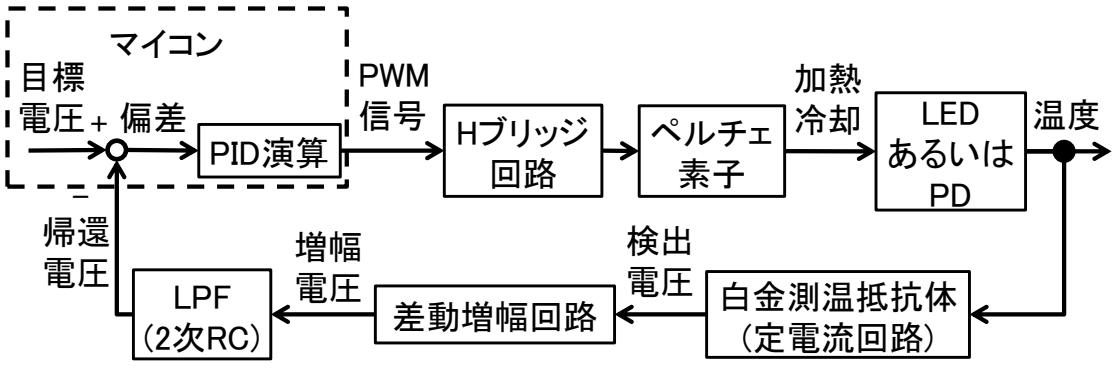

図 2.11　LED と PD 温度制御のブロック線図

3. 実験結果

電流制御と温度制御における PID ゲインを決定した後に、24 時間継続して強度とスペクトルの測定を行った。

3.1. 電流制御における PID ゲイン

図 3.1 に比例ゲインを 3、5、7 に変化させた場合、電流制御結果にどのような違いがあるのかを示す。強度を PD 電圧として測定して、目標電圧は 4.0V とした。比例ゲインが 3 と 5 の結果を比較すると、比例ゲインが 5 の場合に残留偏差が 0.3V と小さくなったことと、比例ゲインが 7 の結果で PD 電圧の変動値が 0.5V と最大となったことから比例ゲインは 5 とした。このとき、変動値は定常状態における出力の最大値と最小値の差と定義した。

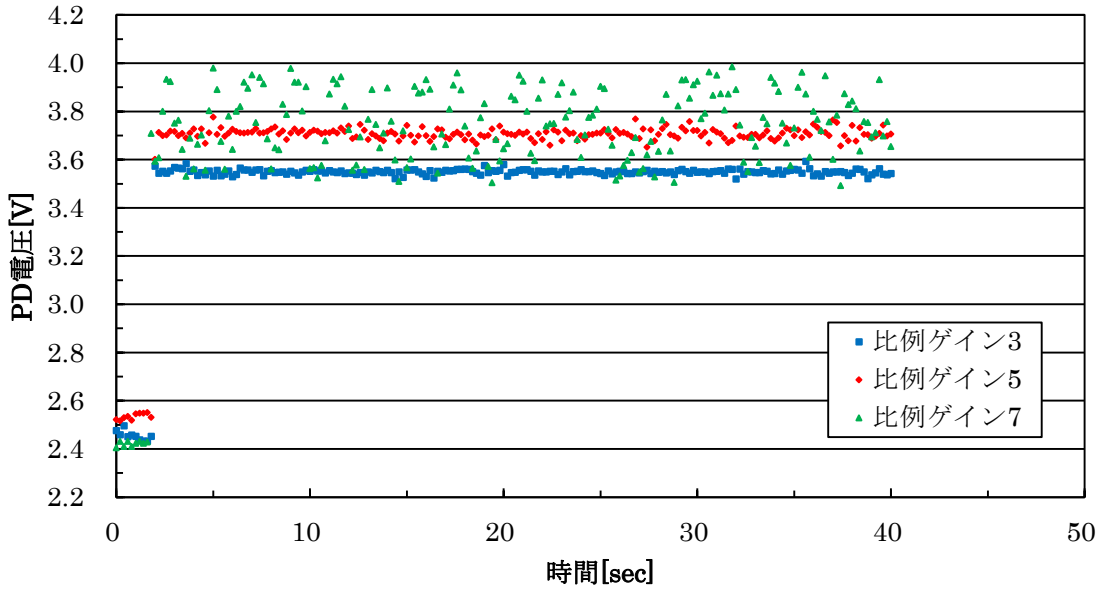

図 3.1　比例ゲインを変化させた場合の電流制御結果

比例制御のみでは、定常状態で目標値と出力値に、ある一定値の偏差(残留偏差)が存在するので、残留偏差を取り除くために比例制御に積分制御を加えることにした。図 3.2 に積分ゲインを 0.5、0.7、0.9 に変化させた場合、電流制御結果にどのような違いがあるのかを示す。積分ゲインが 0.5 と 0.7 の結果を比較すると、積分ゲインが 0.7 の方が目標電圧に到達する時間が早くなった。次に積分ゲインが 0.7 と 0.9 の結果から、目標電圧に到達する時間が 25 秒で同じとなり、PD 電圧の変動が小さいことから積分ゲインは 0.7 とした。

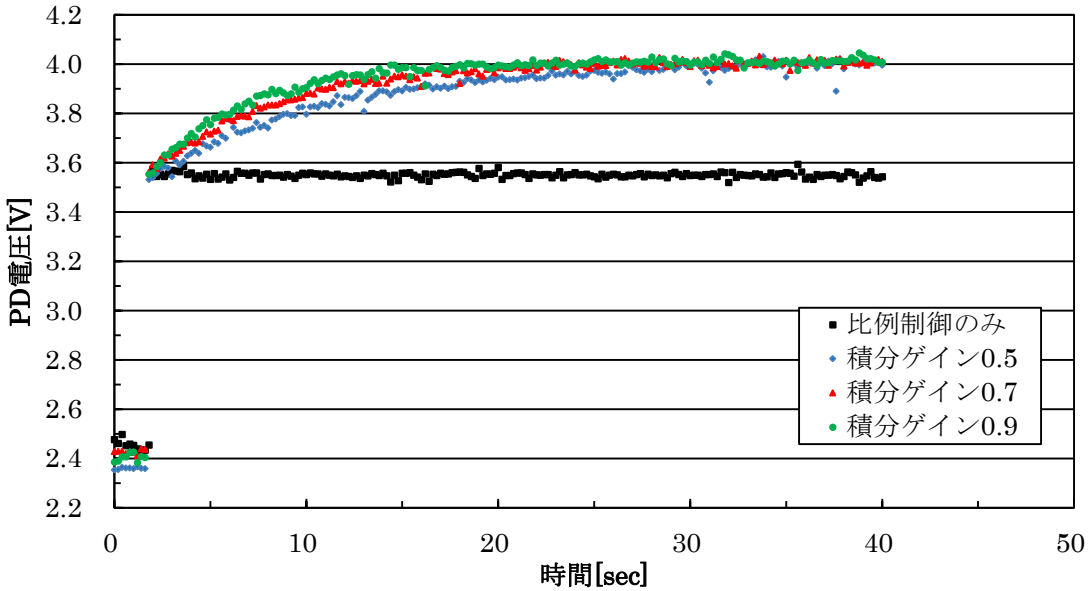

図 3.2　積分ゲインを変化させた場合の電流制御結果

図 3.3 に微分ゲインを 0.1、0.5 に変化させた場合、電流制御結果にどのような違いがあるのかを示す。目標電圧に収束した後の PD 電圧の変動値を比べると、比例+積分制御では 0.02V、微分ゲインが 0.05 では 0.2V、微分ゲインが 0.1 では 0.5V となったことと、目標電圧に到達する時間は 20 秒で同じとなったことから、微分ゲインは 0 として比例+積分制御で電流を制御することにした。

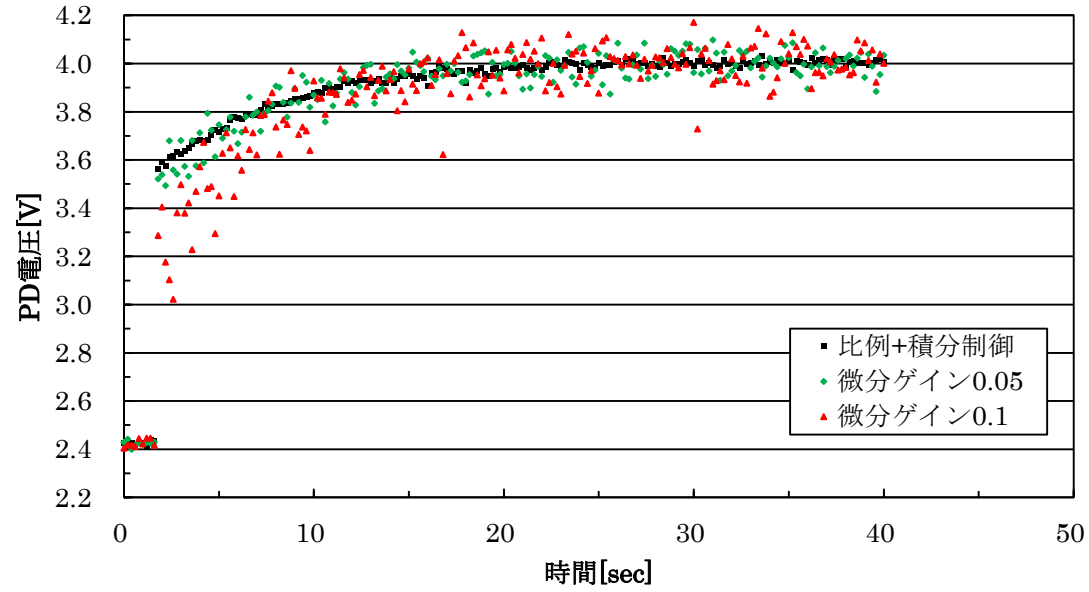

図 3.3　微分ゲインを変化させた場合の電流制御結果

3.2. LED 温度制御における PID ゲイン

図 3.4 に比例ゲインを 30、35、40 に変化させた場合、LED 温度制御結果にどのような違いがあるのかを示す。このとき、目標温度は 30℃として、LED 温度の変化を調べた。比例ゲインが 30 の場合は LED 温度の変動が最も少ないが、29.7℃に到達するまでに 70 秒かかり、他のゲインと比べて最も遅くなった。比例ゲインが 40 の場合は LED 温度の変動が最も大きくなり、変動が収まるまで時間がかかった。比例ゲインが 35 の場合は残留偏差が比例ゲイン 30 のときより早く、LED 温度の変動が比例ゲイン 40 のときより小さいために、比例ゲインは 35 とした。

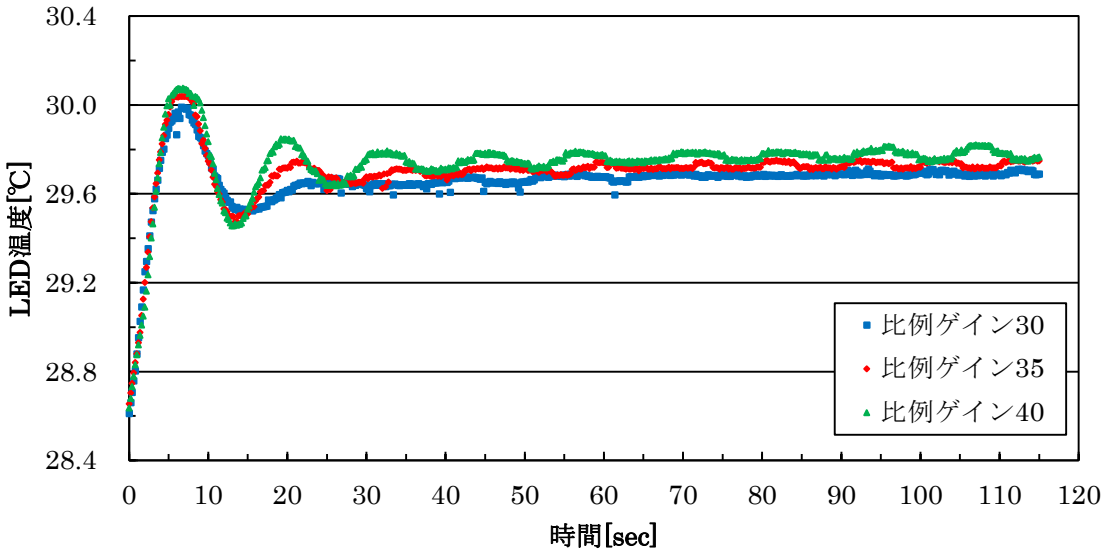

図 3.4　比例ゲインを変化させた場合の LED 温度制御結果

図 3.5 に積分ゲインを 0.1、0.3、0.5 に変化させた場合、LED 温度制御結果にどのような違いがあるのかを示す。図 3.5 から積分ゲインが 0.3 の場合は行き過ぎ量が 0.3℃で、積分ゲインが 0.1 のときと比べて 0.1℃しか変化しなかったことと、最も早く目標温度の 30℃に収束したことから積分ゲインは 0.3 とした。

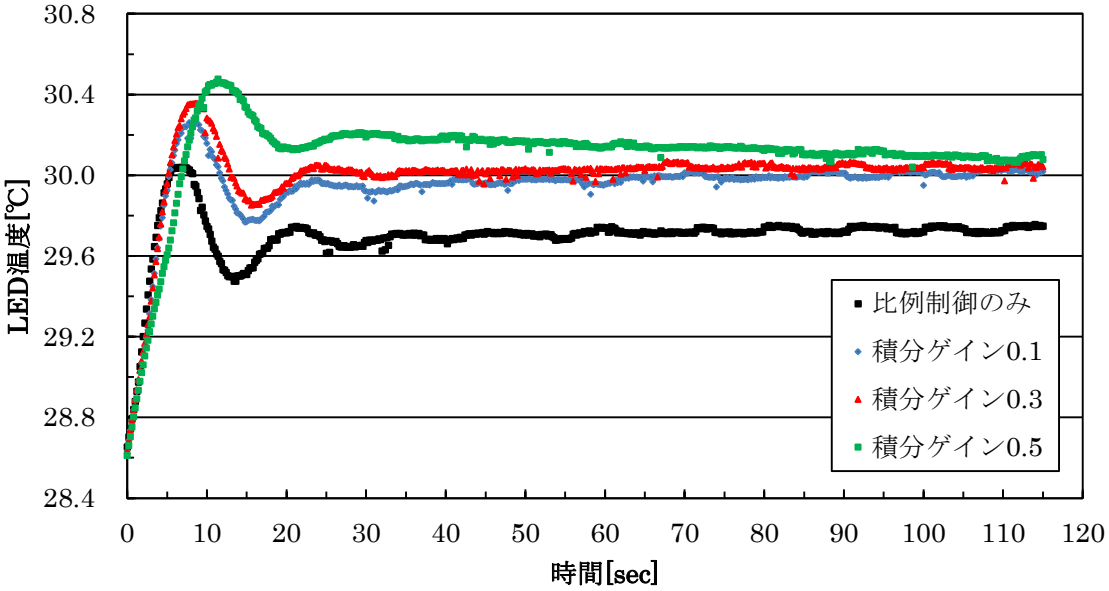

図 3.5　積分ゲインを変化させた場合の LED 温度制御結果

図 3.6 に微分ゲインを 0.1、0.5 に変化させた場合、LED 温度制御結果にどのような違いがあるのかを示す。図 3.6 から行き過ぎ量を比べると微分ゲインが 0.5 の方が小さくなったが、目標温度に収束した後の LED 温度の変動は微分ゲインが 0.1 の方が小さくなり、温度の安定性を重視して微分ゲインは 0.1 とした。

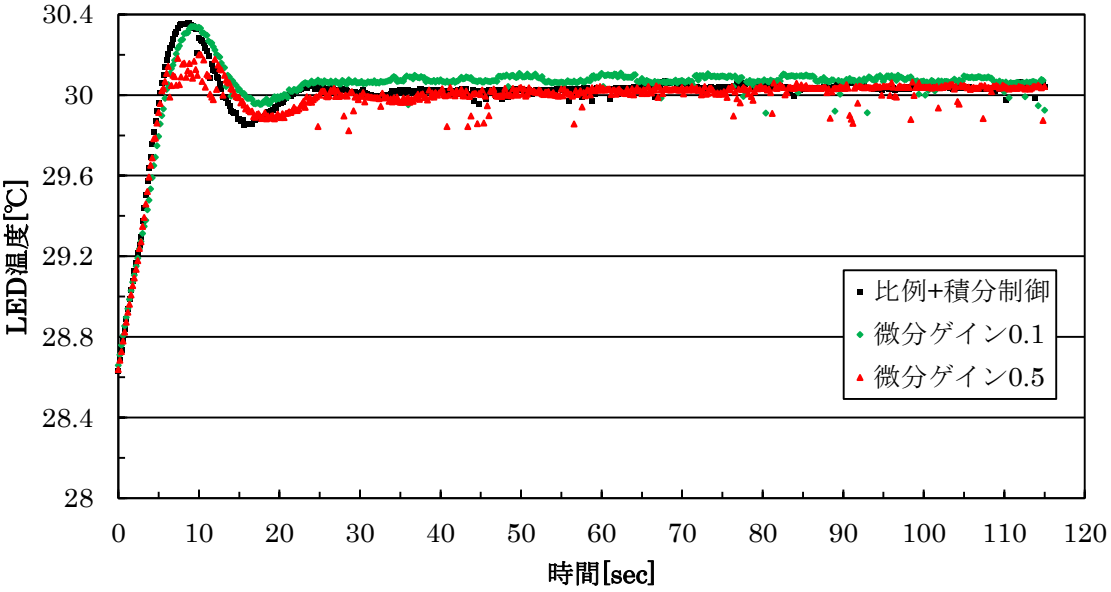

図 3.6　微分ゲインを変化させた場合の LED 温度制御結果

3.3.　PD 温度制御における PID ゲイン

図 3.7 に比例ゲインを 30、40、50 に変化させた場合、PD 温度制御結果にどのような違いがあるのかを示す。このとき、目標温度は 30℃として、PD 温度の変化を調べた。比例ゲインが 40 のときに 28 秒で一定温度に収束して、定常状態になる時間が最も早くなったので、比例ゲインは 40 とした。

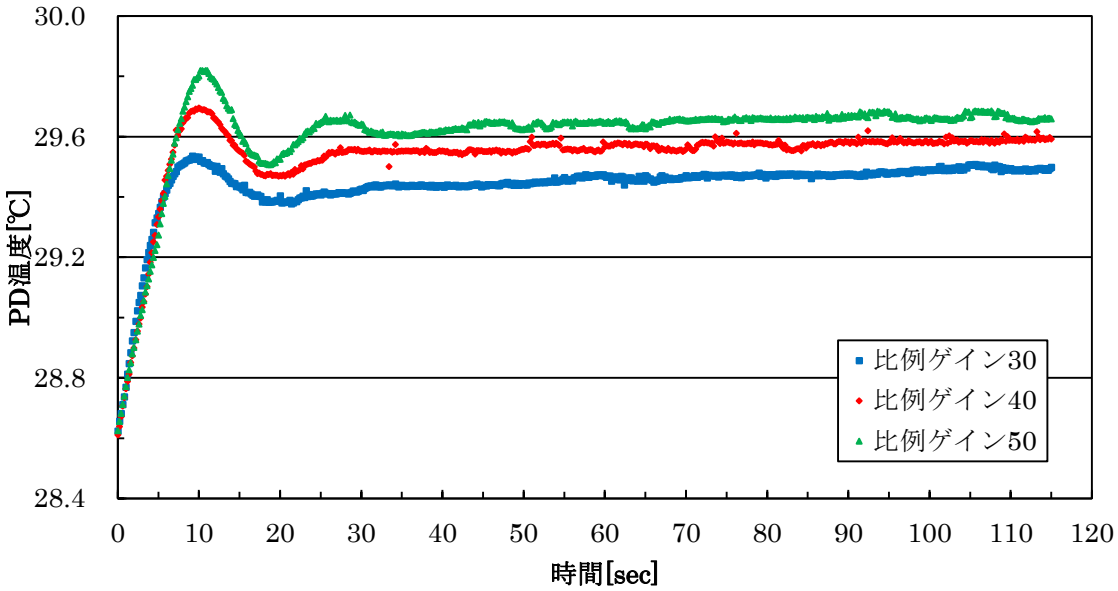

図 3.7　比例ゲインを変化させた場合の PD 温度制御結果

図 3.8 に積分ゲインを 0.7、1.0、1.3 に変化させた場合、PD 温度制御結果にどのような違いがあるのかを示す。積分ゲインが 1.0 のときに、目標温度 30℃に収束する時間が 35 秒で最も早くなったので、積分ゲインは 1.0 とした。

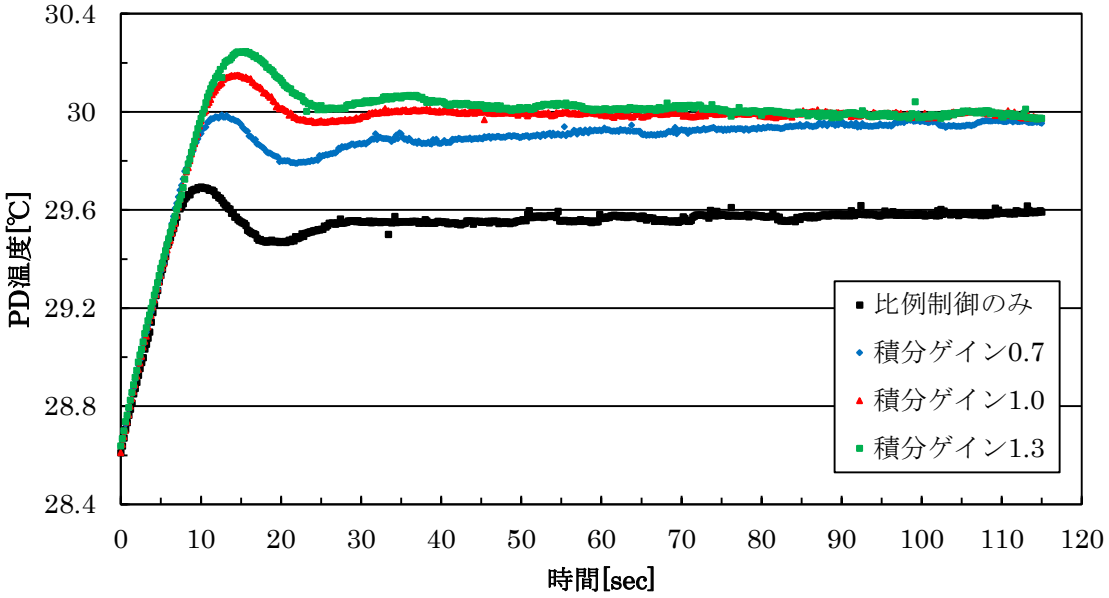

図 3.8　積分ゲインを変化させた場合の PD 温度制御結果

図 3.9 に微分ゲインを 0.1、0.2 に変化させた場合、PD 温度制御結果にどのような違いがあるのかを示す。微分ゲインが 0.1 と 0.2 の結果を比べると目標温度に収束する時間と、PD 温度の変動値が同じであるから、温度の安定性を重視して微分ゲインは 0.1 とした。

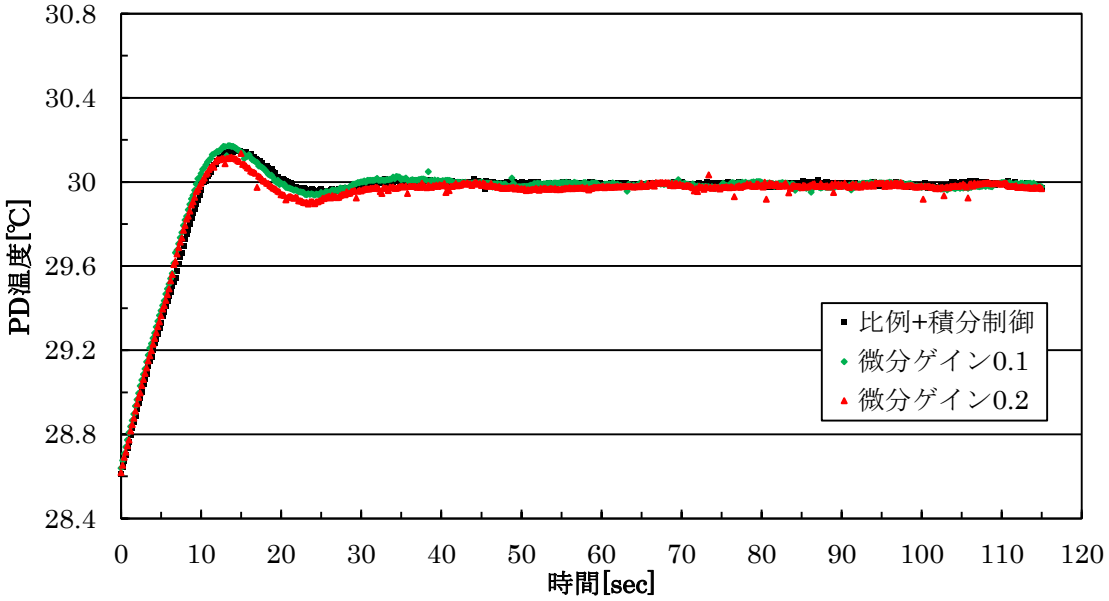

図 3.9　微分ゲインを変化させた場合の PD 温度制御結果

3.4. 強度変化測定結果

図 3.10 に電流と温度を制御した場合と制御しないの場合の強度の時間変化を示す。制御なしの場合、測定を始めたときから強度が減少して、24 時間後に変化率が約-40%になり、変化率の標準偏差は 2.3%になった。電流と温度を制御した場合、変化率が最大で 5.9%になり、変化率の標準偏差は 0.4%になった。

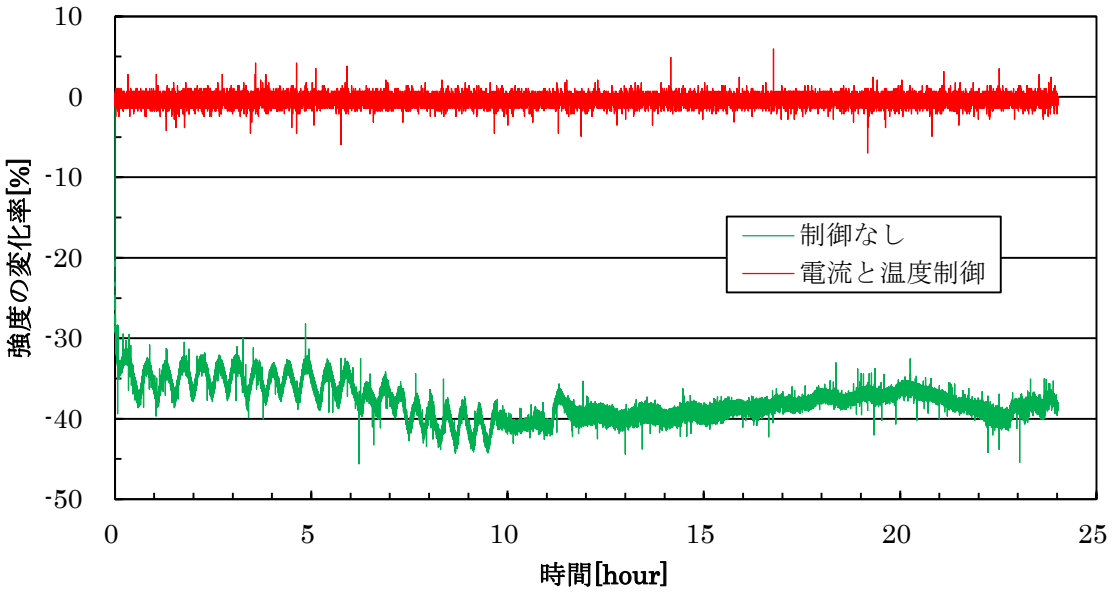

図 3.10　強度の時間変化

3.5. スペクトル変化測定結果

図 3.11 に制御しない場合のスペクトルの測定結果を示す。測定開始から 1 分後と 24 時間後のスペクトルを比較すると各波長の強度の変化率は波長 643nm が最大で 7.2%となった。1 分後と 24 時間後のスペクトル測定結果で強度が最大となったピーク波長は 631nm で同じとなった。

図 3.12 に制御した場合のスペクトルの測定結果を示す。測定開始から 1 分後と 24 時間後のスペクトルを比較すると各波長の強度の変化率は波長 653nm が最大で-3.1%となった。全体としてスペクトルは 0.5nm ずれて短波長化した。

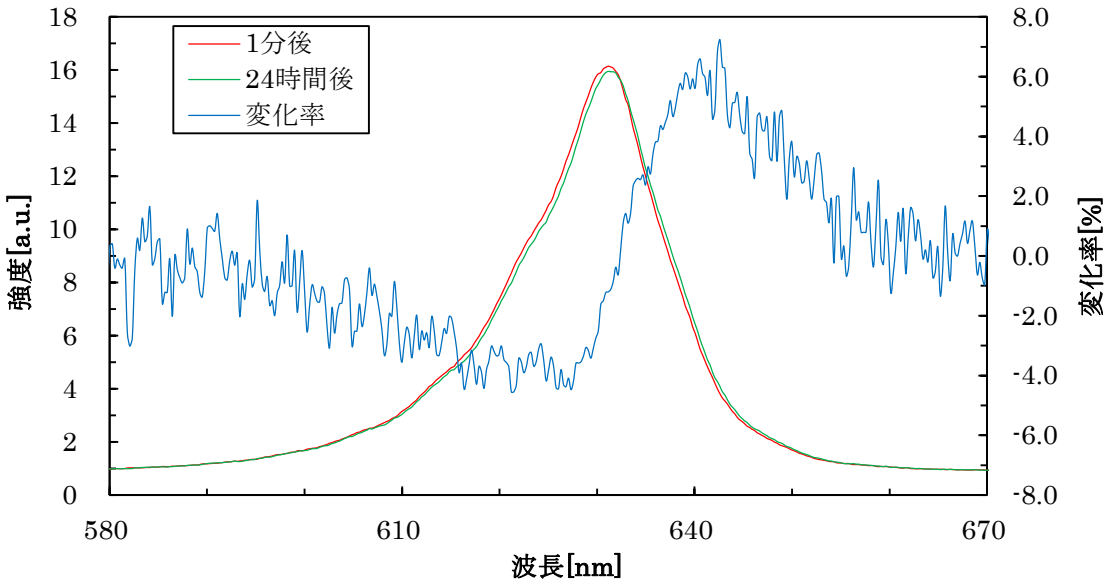

図 3.11　制御しない場合のスペクトル測定結果

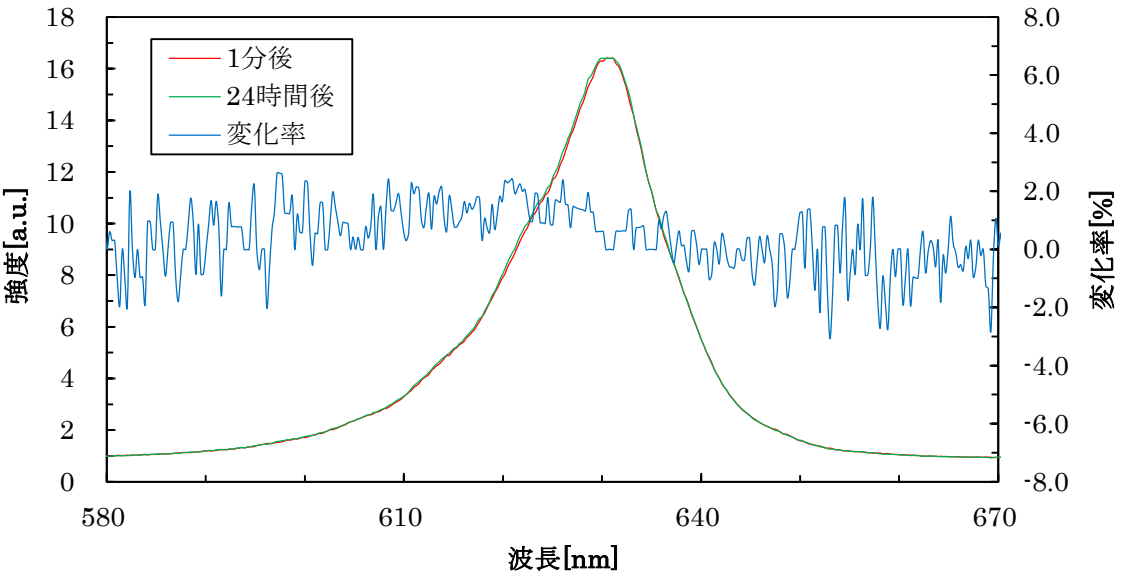

図 3.12　制御した場合のスペクトル測定結果

3.6. LED 温度測定結果

図 3.13 に電流と温度を制御した場合と制御しないの場合の LED 温度の時間変化を示す。制御なしの場合、LED の温度は 26.8℃から 29.3℃に上昇した。制御した場合、LED の温度は 24.7℃から 25.3℃の間を変化して、標準偏差は 0.03℃となった。

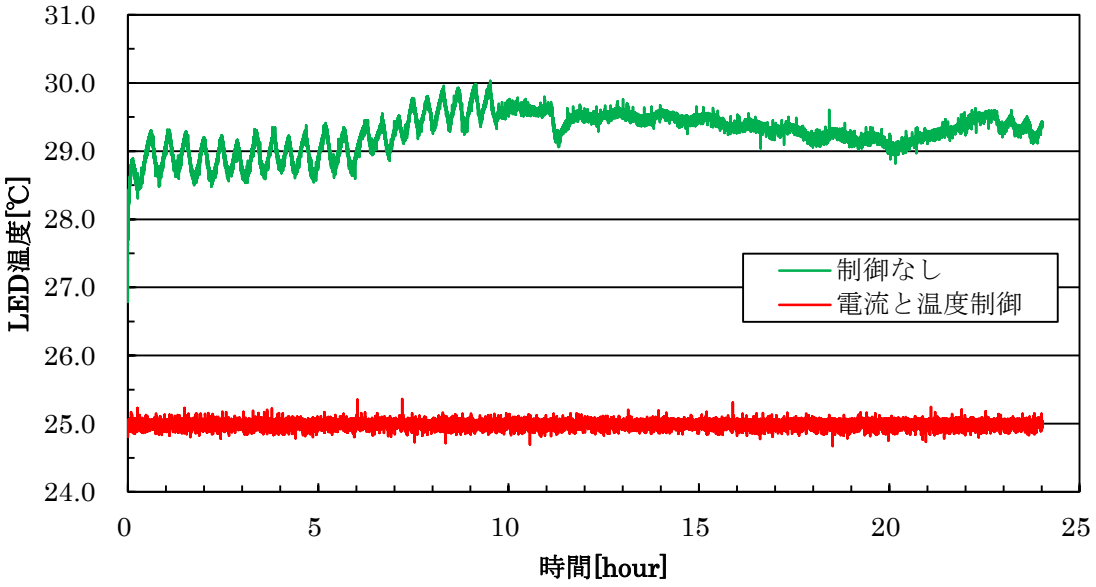

図 3.13　LED 温度の時間変化

3.6.　電流測定結果

図 3.14 に制御した場合の電流測定結果を示す。LED に流れる電流は 283.5mA から 283.1mA の間を変化した。強度を表す PD 電圧は 3.0V から 2.7V を変化して、標準偏差は 0.02V となった。

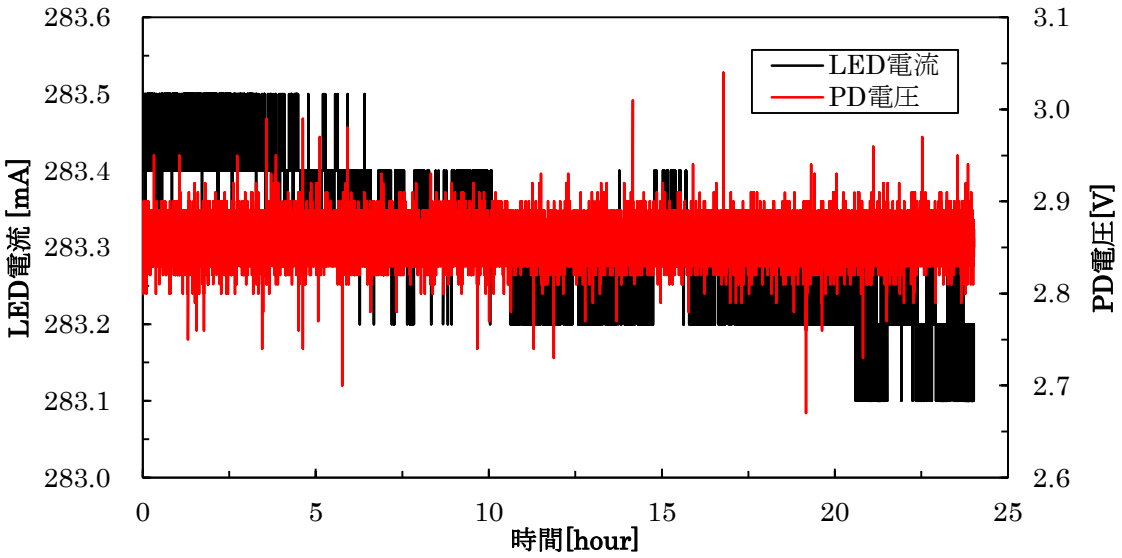

図 3.14　電流測定結果

3.8.　PD 温度測定結果

図 3.15 に電流と温度を制御した場合の PD 温度の時間変化を示す。PD の温度は 24.7℃から 25.2℃の間を変化して、標準偏差は 0.03℃となった。

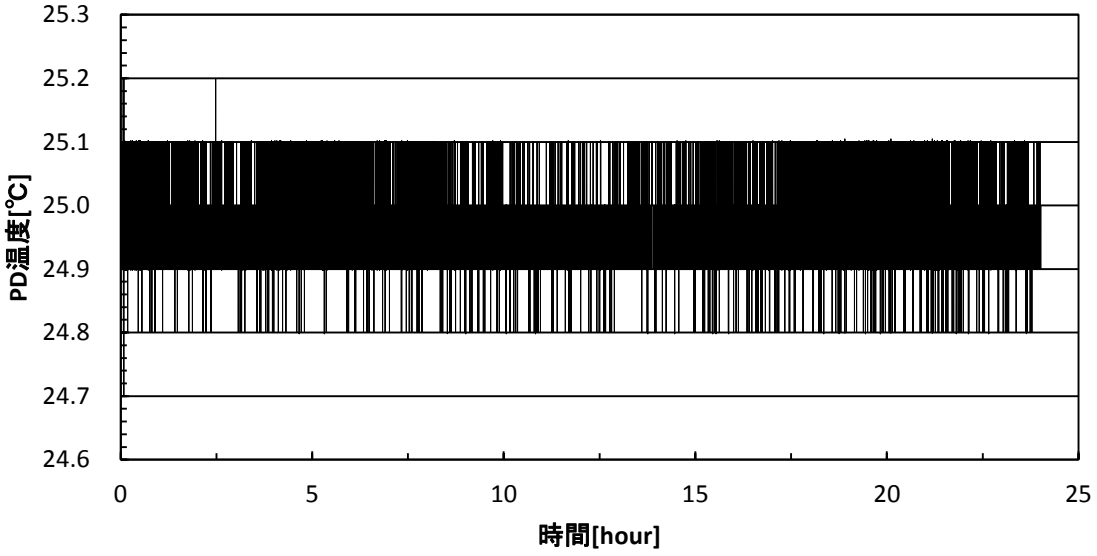

図 3.15　PD 温度の時間変化

4. まとめと考察

LED を使って強度とスペクトルの波長変化率が 2%以下になる高安定光源を開発することを目指した。LED の強度と PD の出力電流は温度により変化するため、LED は電流と温度、PD は温度を制御する装置を作製して、強度とスペクトルを測定した。

24 時間の測定で、制御をした場合に強度の変化率の標準偏差は 0.4%になり、スペクトルの変化率は波長 653nm が最大で-3.1%になった。スペクトルの変化率が大きくなった原因は LED の温度上昇における短波長化だと考えられる。制御なしの場合にスペクトルの変化率は波長 643nm が最大で 7.2%になり、LED 温度は 1 分後から 24 時間後に 1.3℃高くなった。これからスペクトルの変化率は 1℃あたりに 5.5%変化するので、2%以下にするには LED の温度変化を 0.36℃以下にしなければならない。LED の温度をさらに安定化することにより、スペクトルの波長変化率を小さくして、長時間使用した場合でも安定性の高い光源にする必要がある。また、電気的なノイズを低減することで温度と強度を安定化する必要がある。

表 1 強度測定結果

	PD 電圧[V]				変化率[%]	
	最大値	最小値	最大と最初の電圧差	標準偏差	最大値	標準偏差
制御あり	3.0	2.7	0.3	0.02	5.9	0.4
制御なし	2.9	1.6	1.3	0.07	45.6	2.3

表 2 温度測定結果

	最高温度[℃]	最低温度[℃]	最高温度と最低温度の差[℃]	標準偏差[℃]
LED	25.3	24.7	0.6	0.03
PD	24.7	25.2	0.5	0.03

表 3 スペクトル測定結果

	最大変化率[%]	ピーク波長[nm]		ピーク波長の差[nm]
		1 分後	24 時間後	
制御あり	−3.1	630.4	629.9	0.5
制御なし	7.2	631.1	631.1	0.0

5. 参考文献

[1]フォトダイオ-ドの特性と使い方, 浜松ホトニクス, <http://www.hepl.hiroshima-u.ac.jp/phx/constr/toyoda/Mthesis/Figure/si_pd_technical_information.pdf>, 参照 2013-09-09

[2]椹木 義一, 添田 喬 : わかる自動制御, 日新出版, 1999

[3]Hank Zumbahlen : OP アンプによるフィルタ回路の設計, CQ 出版, 2005

[4]松井邦彦 : トランジスタ技術 SPECIAL No.37 特集　実用電子回路設計マニュアルⅡ, CQ 出版, 1993

6. 謝辞

本研究を通してご指導、助言を頂きました由井四海先生、装置の製作にあたり技術的に支援してくださった飯田拓也技官、島政司技官、早川幸弘技官にお礼申し上げます。

7. 付録

7.1. 回路図

図 7.1 に電流制御回路、図 7.2 に温度制御回路を示す。温度制御回路は LED 温度制御回路と PD 温度制御回路の合わせて 2 つの回路を作製した。

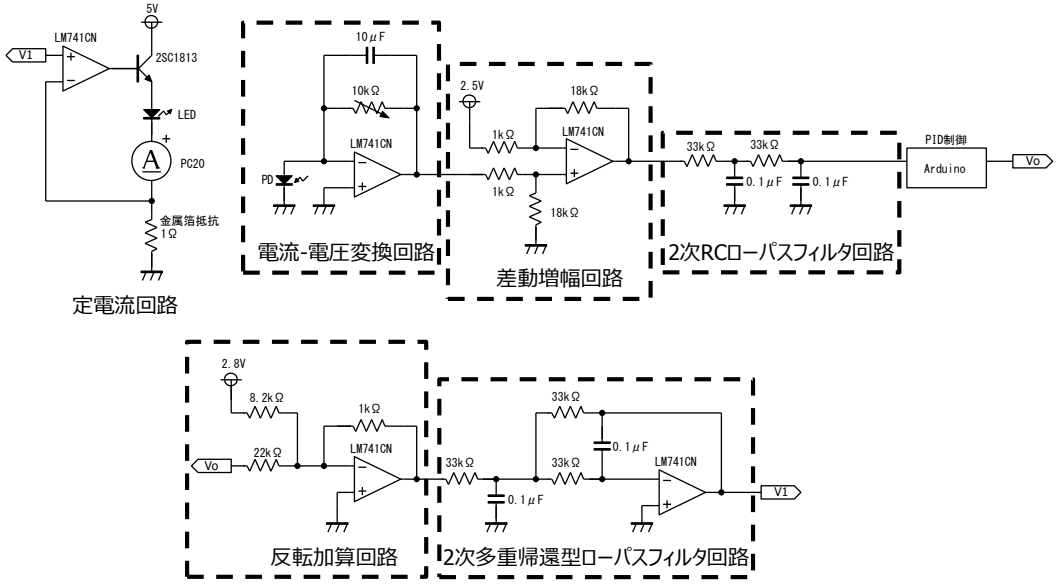

図 7.1　電流制御回路

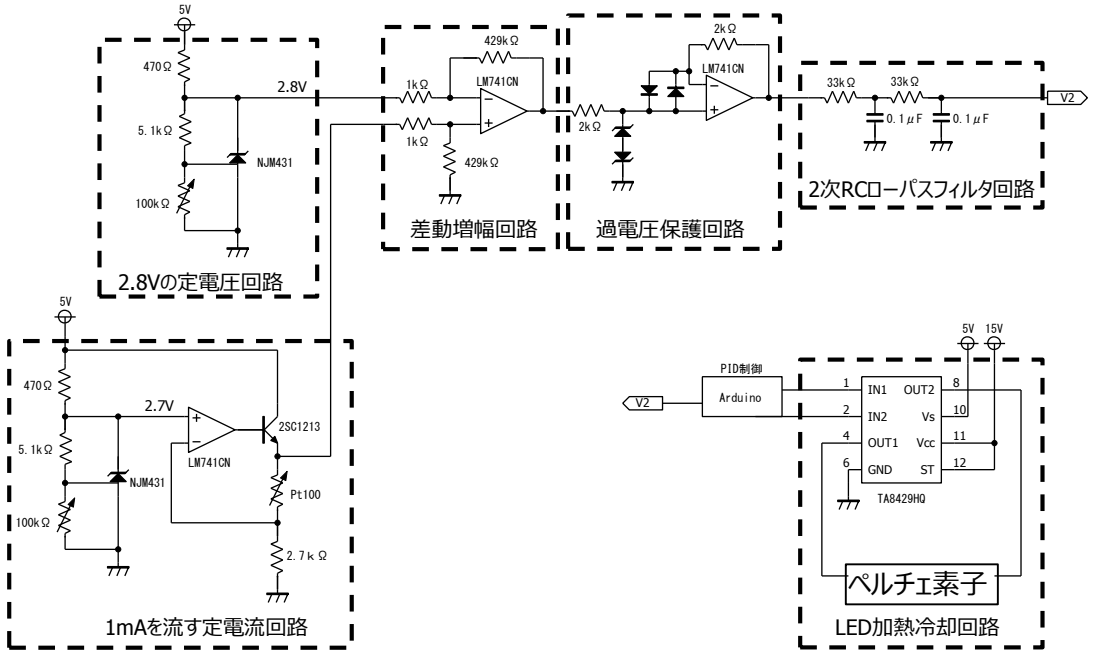
5V
470Ω
5.1kΩ
100kΩ
NJM431
2.8V
2.8Vの定電圧回路
429kΩ
1kΩ
LM741CN
1kΩ
429kΩ
差動増幅回路
2kΩ
LM741CN
2kΩ
過電圧保護回路
33kΩ
33kΩ
0.1μF
0.1μF
V2
2次RCローパスフィルタ回路
5V
470Ω
2.7V
5.1kΩ
LM741CN
2SC1213
NJM431
Pt100
100kΩ
2.7kΩ
1mAを流す定電流回路
5V
15V
PID制御
V2
Arduino
1
2
4
6
IN1
IN2
OUT1
GND
OUT2
Vs
Vcc
ST
8
10
11
12
TA8429HQ
ペルチェ素子
LED加熱冷却回路

図 7.2　温度制御回路

7.2.　主要部品および測定機器表

表 4　主要部品および使用機器

名称	型番	規格	製造元	個数
赤色パワーLED	OSR5XME1C1E	max.400mA,ピーク波長 625nm	OptoSupply Limited	1 個
ペルチェ素子	TEC1-12705	サイズ 30cm×30cm×4.1mm, 最大電流 5A, 最大使用電圧 15.4V, 最大吸熱量 44.5W,最大温度差 68℃, 抵抗約 2.40Ω	HB Electronic Components	1 個
	TEC1-12706	サイズ 40cm×40cm×4.1mm, 最大電流 6A, 最大使用電圧 15.4V, 最大吸熱量 53.3W,最大温度差 68℃, 抵抗約 1.98Ω		1 個
分光器	HR4000CG-UV-NIR	分解能 0.75 nm	OceanOptics	1 台
フォトダイオード (PD)	S2387-1010R	受光感度 0.58 A/W, 感度波長範囲 340～1100 nm,受光面サイズ 10 x 10 mm	浜松ホトニクス	1 個
白金測温抵抗体	FK 222-100-A	温度係数 0.39Ω/℃, 定格電流 1mA,精度 0.1℃	Heraeus	2 個
冷却ファン	D02X-05TS1 02	定格電圧 5V,定格電流 0.05A	Nidec	3 個
モータドライバ	TA8429HQ	最大出力電流 3.0A	TOSHIBA	2 個
金属箔抵抗	TO-220	1Ω, 温度係数-5～5ppm/℃, 許容差±0.5%	アルファ・エレクトロニクス	1 個
マイコン	ARD-UNO	分解能 12bit,ATmega328 搭載	Arduino	3 個
電流計	PC20	測定電流 0.4mA～10A	sanwa	1 台
DAQ アシスタント	NI USB-6221	分解能 16bit	National Instruments	1 台

7.3. プログラム

Arduino(マイコン)に書き込んだプログラムを示す。

```
//PID 制御を行うプログラム
//LED に加える初期電圧 2.5V を表す定数を定義する
#define INITIAL_INF_INPUT 124
//処理周期 DELTA_T を 2.181msec とする pwm 信号の周期が 2.041m+アナログ信号読み取り 0.1m+その他 0.04m
#define DELTA_T 0.002181

//制御で用いる目標値を格納する変数を宣言する
int infSetpoint,temLEDSetpoint,temPDSetpoint;
//PID ゲインの設定
double temLEDKp=35.0, temLEDKi=0.5, temLEDKd=0.1;
double temPDKp=40.0, temPDKi=1.0, temPDKd=0.3;
double infKp=-3.0, infKi=-0.7, infKd=-0.0;
//入力ピンの番号を決める
int inInfPin = 3,inTemLEDPin = 5,inTemPDPin = 4;
//ピン 5 と 6 の PWM 出力はデューティ比が高めになるので避ける(リファレンス参照)
int outInfPin = 3,outTemLEDPin = 9,outTemLEDDrivePin = 8,outTemPDPin = 11,outTemPDDrivePin = 12;
//関数で引数として渡すために、PID ゲインを配列に格納する。
double infPID[3] = {infKp,infKi,infKd};
double temLEDPID[3] = {temLEDKp,temLEDKi,temLEDKd};
double temPDPID[3] = {temPDKp,temPDKi,temPDKd};

//電流制御を行う関数
void setOutVoltage(int inputPin,int outPin,double pid[],int setPoint,int constant){
  double deviation,output;
  int input,outputVoltage;
  input = analogRead(inputPin);
  deviation = (input-setPoint)*5/(double)1023;
  output = computeOutput(pid,deviation)+constant;
  output *= 51;//電圧を PWM の出力値(0～255)に変換
  if(output > 255){
    outputVoltage = 255;
  }
  else if(output < 0){
    outputVoltage = 0;
```

```
    }
    else{
      //int 型は-32768 から 32767 なので 0＜output＜255 なら int 型にキャストしても値が反転しないため
      //int 型の制御量(outputVoltage)に代入できる
      outputVoltage = (int)output;
    }
    analogWrite(outPin,outputVoltage);
}
//温度制御を行う関数
void setOutVoltage(int inputPin,int outPin,int outDrivePin,double pid[],int setPoint){
    double deviation,output;
    int input,outputVoltage,outAbsVoltage;
    input = analogRead(inputPin);
    deviation = (input-setPoint)*5/(double)1023;
    output = computeOutput(pid,deviation);
    output *= 51;
    if(output > 0){//冷却
      if(output > 255){
        outputVoltage = 255;
      }
      else{
        outputVoltage = (int)output;
      }
      digitalWrite(outDrivePin,LOW);
    }
    else if(output < 0){//加熱
      if(output < -255){
        outputVoltage = 0;
      }
      else{
        outputVoltage = 255+(int)output;
      }
      digitalWrite(outDrivePin,HIGH);
    }
    else{//output(制御量)が 0 のとき
      outputVoltage = 0;
```

```
      digitalWrite(outDrivePin,LOW);
    }
    analogWrite(outPin,outputVoltage);
}

void setup(){
    //LED 温度制御の設定
    pinMode(outTemLEDDrivePin,OUTPUT);
    digitalWrite(outTemLEDDrivePin,LOW);
    analogWrite(outTemLEDPin,0);//温度初期電圧 0V を印加する
    temLEDSetpoint = 712;
    //

    //PD 温度制御の設定
    pinMode(outTemPDDrivePin,OUTPUT);
    digitalWrite(outTemPDDrivePin,LOW);
    analogWrite(outTemPDPin,0);//温度初期電圧 0V を印加する
    temPDSetpoint = 712;

    //LED 電流制御の設定
    analogWrite(outInfPin,INITIAL_INF_INPUT);//初期電圧 2.5V を印加する
    delay(5000);//5 秒待つ
    infSetpoint = analogRead(inInfPin);
    infSetpoint = (infSetpoint+analogRead(inInfPin))/2;
    //
}

void loop(){
     //LED 電流制御
     setOutVoltage(inInfPin,outInfPin,infPID,infSetpoint,INITIAL_INF_INPUT);
     //LED 温度制御
     setOutVoltage(inTemLEDPin,outTemLEDPin,outTemLEDDrivePin,temLEDPID,temLEDSetpoint);
     ////PD 温度制御
     setOutVoltage(inTemPDPin,outTemPDPin,outTemPDDrivePin,temPDPID,temPDSetpoint);
     //PWM 周期 2041us だけ待機する
     delayMicroseconds(2041);
```

```
}

double computeOutput(double pid[],double deviation){
    double pastDeviation,deltaOut;
    static double newDeviation,out,integral;
    pastDeviation = newDeviation;
    newDeviation = deviation;
    integral += (newDeviation+pastDeviation) / 2.0 * DELTA_T;
    out = pid[0] * newDeviation + pid[1] * integral + pid[2] * (newDeviation - pastDeviation) / DELTA_T;
    return out;
}
```