

原子物理実験用デジタル制御周波数安定化レーザー

先進理工学専攻 中川研究室 山本真稔

現在、我々の研究室では、リドベルグブロッケード実験をはじめとした原子物理実験を行っており、複数台のレーザーを安定化している。本研究は、今までアナログ回路で制御していた光学系をデジタル回路に置き換えることで、制御回路の作製と実験そのものの効率化を図るため行われた。

1 研究背景

近年、原子やイオンを扱う実験が盛んに行われており、その多くに安定化したレーザーが要求されている。例えば、リドベルグブロッケードの実験で使われる 960nm のレーザー光は、図 1 のように原子の吸収線に安定化したレーザー 1 を基準として、周波数オフセットロックしたレーザー 2 を元に共振器を安定化して、このロックした共振器を基準に安定化されている [1]。つまり、960nm のレーザー光一つを安定化するだけで、四箇所制御を行う必要がある。

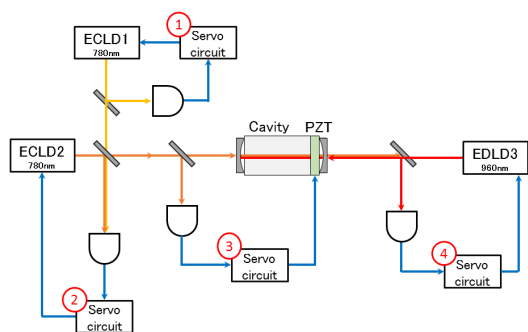


図 1 960nm 光安定化の光学系

これらのレーザー安定化には、従来アナログ回路が使用されてきた。1985 年には、D. W. Page らによって最初のプログラム可能なロジックアレイの特許 [2] が出されているが、デジタル回路では、制御したい帯域まで遠く及ばなかったためである。しかし、アナログ回路は一つ作製する度に、半田付けなどで人手と時間がかかる。また、現実の素子は理想的な挙動とは異なるため、設計どおり動かないこともある。一方、デジタル回路は、特性をその場ですぐに変更でき、一度設計してしまえば、複製も容易である。近年、デジタル技術は大幅に性能が向上し、レーザーの制御も可能となりつつある [3][4]。

2 研究目的

本研究の目標は、次の二つである。

- ・汎用的なデジタルフィルタの作製
- ・制御作業の自動化

FPGA を用いて、作製や最適化の時間と手間を大幅に削減できる汎用的なフィルタを開発する。本研究では、これを用いて図 1 の実験系の ③ 部分、共振器長の安定化も行う。また、今までの制御では、制御点を手動で捜していた。実験をより効率良く行うため、この制御点を捜す手順の自動化を目指す。

3 原理

3.1 FPGA

FPGA(Field Programmable Gate Array) は、図 2 のように多数の基本ロジックセルを格子状に並べた構造のデバイスである。各セルの入出力は縦横方向に張り巡らされた配線部に接続されており、セル同士を繋ぐことで任意の回路を構成することができる。

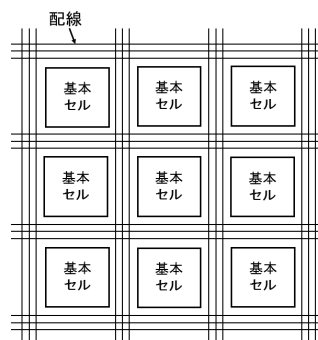


図 2 FPGA の構造

この FPGA と PC の大きな違いは、応答速度である。我々が行っている実験では、レーザーの線幅を 100kHz 以下に抑えたい。そのためには、デジタルの

応答は、2倍以上は必須であり、10倍の1MHz程度が望ましい。現在のPCでは、速いものでも100kHz程度であり、不十分である。一方、FPGAは、数MHzのものまで開発されている。この違いは、構造と処理方式に起因する。PCの処理を図3に、FPGAの処理を図4に示す。PCは、既に作られたハードウェア部分を組み合わせて使い、データを処理する。この処理の度にレジスタにデータを蓄え、処理も一つ一つしか行えないため、速度が遅い。一方、FPGAは、ハードウェア部分を自由に作り変え、行いたい処理に適した専用の回路を構築する。これによって、PCと違いデータをレジスタに蓄えることなく、データが来たらすぐに処理を行える。また、処理も並列に行うことができるため、速い速度での処理が可能となる。

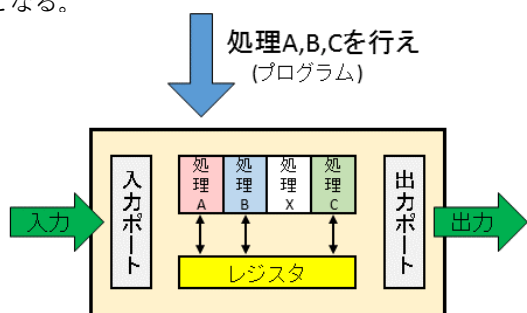


図3 PCの構造と処理方式

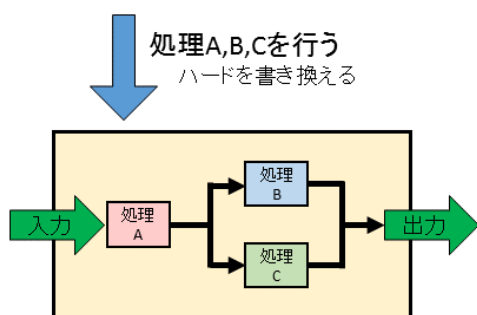


図4 FPGAの構造と処理方式

今回、FPGAボードはXilinxのZ-7010を使用した。Z-7010はDSPスライスが80個あり、A/Dコンバーター、D/Aコンバーター込みの遅延が $8\mu\text{s}$ なので、125kHz以上の信号から位相が 180° 回る。後述するIIRフィルター一つを作るのに必要なDSPスライス数が3~5個であり、本研究で行う共振器長のロックでは、PZTを制御対象に数十kHz程度までしか制御しないため、十分な性能である。

3.2 デジタルフィルタ

デジタルフィルタには、FIR (finite impulse response : 有限インパルス応答) フィルタとIIR (infinite impulse response : 無限インパルス応答) フィルタの2種類がある。FIR フィルタは、有限個の入力データを用いて出力を決めるデジタルフィルタであり、一般式は

$$y_k = \sum_{n=0}^N a_n x_{k-n} \quad (1)$$

で表せる。 x_k は現在の入力値、 y_k は現在の出力値である。これに対し、IIR フィルタは、

$$y_k = \sum_{m=1}^M a_m y_{k-m} + \sum_{n=0}^N b_n x_{k-n} \quad (2)$$

で表され、現在の出力値を決める際に、入力だけでなくそれ以前の出力値を用いるため、無限に前のデータまで参照して値を決めている。IIR フィルタは、FIR フィルタよりも低次数の計算で良好な周波数振幅特性が得られるので、本研究ではIIR フィルタを用いた。

ここで、フィルタ設計に必要なパラメータ a_m, b_n について考える。例えば、ローパスフィルタの場合、その伝達関数は、

$$H(s) = \frac{K}{1 + \frac{s}{2\pi f_0}} \quad (3)$$

と表される。ここで、 K は比例ゲイン、 f_0 はカットオフ周波数、 s は角速度 ω を用いて、 $s = i\omega$ と表される複素数である。この連続した伝達関数 $H(s)$ を双1次変換を用いて、不連続な伝達関数 $H(z)$ に書き換える。双1次変換は、FPGAのサンプリング時間 T_s を用いて、

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

と表せる。式(3)に双1次変換を適用すると

$$H(z) = \frac{K \frac{\pi f_0 T_s}{1 + \pi f_0 T_s} + K \frac{\pi f_0 T_s}{1 + \pi f_0 T_s} z^{-1}}{1 - \frac{1 - \pi f_0 T_s}{1 + \pi f_0 T_s} z^{-1}} \quad (5)$$

と変形できる。

表1 IIR フィルタの係数

フィルタ型	伝達関数	a_1/a_0	a_2/a_0	b_0/a_0	b_1/a_0	b_2/a_0
ローパス	$\frac{K}{1+\frac{s}{2\pi f_0}}$	$\frac{1-\tilde{f}_0}{1+\tilde{f}_0}$		$\frac{K\tilde{f}_0}{1+\tilde{f}_0}$	$\frac{K\tilde{f}_0}{1+\tilde{f}_0}$	
ラグリード	$K\frac{1+\frac{s}{2\pi f_0}}{\frac{1}{g}+\frac{s}{2\pi f_0}}$	$\frac{1-\tilde{f}_0/g}{1+\tilde{f}_0/g}$		$K\frac{1+\tilde{f}_0}{1+\tilde{f}_0/g}$	$-K\frac{1-\tilde{f}_0}{1+\tilde{f}_0/g}$	
ノッチ	$\frac{K[1+(\frac{s}{2\pi f_0})^2]}{1+\frac{s}{2\pi f_0 Q}+(\frac{s}{2\pi f_0})^2}$	$\frac{2[1-\tilde{f}_0^2]}{1+\frac{\tilde{f}_0}{Q}+\tilde{f}_0^2}$	$-\frac{1-\frac{\tilde{f}_0}{Q}+\tilde{f}_0^2}{1+\frac{\tilde{f}_0}{Q}+\tilde{f}_0^2}$	$\frac{K[1+\tilde{f}_0^2]}{1+\frac{\tilde{f}_0}{Q}+\tilde{f}_0^2}$	$-\frac{2K[1-\tilde{f}_0^2]}{1+\frac{\tilde{f}_0}{Q}+\tilde{f}_0^2}$	$\frac{K[1+\tilde{f}_0^2]}{1+\frac{\tilde{f}_0}{Q}+\tilde{f}_0^2}$

また、IIR フィルタの伝達関数 $H(z)$ は、式 (2) の両辺を z 変換して

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots}{a_0 - a_1 z^{-1} - a_2 z^{-2} - \dots} \quad (6)$$

と表せる。IIR フィルタの一般式は、この係数 $a_0, a_1, a_2, \dots, b_0, b_1, b_2, \dots$ を用いて

$$y_n = (a_1 y_{n-1} + a_2 y_{n-2} + \dots + b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} + \dots) / a_0 \quad (7)$$

と書くこともできる。以上の式 (5), (6) より、ローパスフィルタの係数は

$$a_0 = 1 \quad (8)$$

$$a_1 = -\frac{1 - \pi f_0 T_s}{1 + \pi f_0 T_s} \quad (9)$$

$$b_0 = K \frac{\pi f_0 T_s}{1 + \pi f_0 T_s} \quad (10)$$

$$b_1 = K \frac{\pi f_0 T_s}{1 + \pi f_0 T_s} \quad (11)$$

と求められる。

同様に、ラグリードフィルタ、ノッチフィルタの係数も計算でき、それぞれ表 1 のように表される。ここで、 $\tilde{f}_0 = \pi f_0 T_s$ である。

4 FPGA を用いたフィルタ作製

表 1 のパラメータを用いて製作したローパスフィルタ (遮断周波数 200Hz, ゲイン 1 倍) を図 5, 6 に示す。設計通り低周波領域ではゲインが 0dB で一定、周波数が上がると減衰し始め、200Hz でゲインが -3dB となっており、それ以降は周波数が 2 倍になるごとに -6dB で減衰 (= -20dB/dec.) している。位相は 200Hz で -45° となり、そこから傾きがゆるやかになっている。同じく作製したラグリードフィルタ (ラグ周波数 50Hz, リード周波数 500Hz) の結果を図 7, 8 に示す。設計通り低周波領域でのゲ

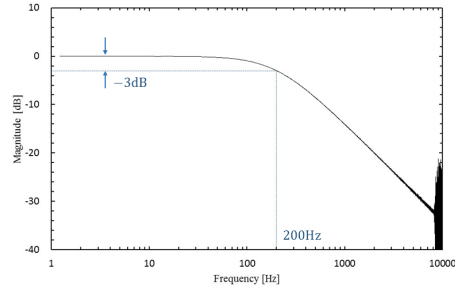


図 5 ローパスフィルタの周波数特性 (ゲイン)

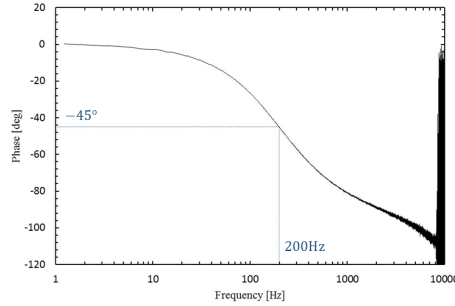


図 6 ローパスフィルタの周波数特性 (位相)

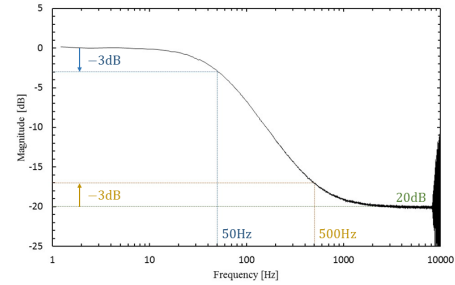


図 7 ラグリードフィルタの周波数特性 (ゲイン)

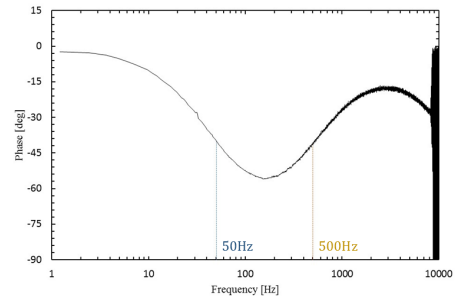


図 8 ラグリードフィルタの周波数特性 (位相)

インが0dBで、50Hzでゲイン-3dBとなり、それ以降は-20dB/dec.で減衰していき、高周波領域でゲインが-20dBで一定、500Hzで-20dBより+3dB高くなっている。また、後述する共振器長の安定化で、特定の周波数を除去する必要があるため、ノッチフィルタも作製した。ただし、ノッチフィルタは表1の通り、2次の a, b まで用いた。10.534kHzの周波数を除去するノッチフィルタの結果を図9に示す。

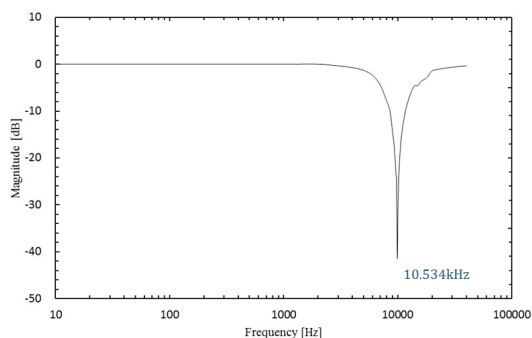


図9 ノッチフィルタの周波数特性 (ゲイン)

設計通り周波数 10.534kHz の成分のみゲインを下げ、他の周波数成分のゲインは0dBとなっている。これらのフィルタは、アナログ回路だと素子の特性に微妙にばらつきがあることや理想的な挙動ではないことから、遮断周波数などが設計通りにいかないことがある。デジタル回路で構築することで設計通りの特性が確かに得られた。

5 FPGA を用いた共振器長の安定化

図10のように Pound-Drever-Hall (PDH) 法で誤差信号を取得し、FPGA を用いて PZT にフィードバックすることで、共振器長を安定化した。Phase shifter として、同軸ケーブルの長さを変えることで

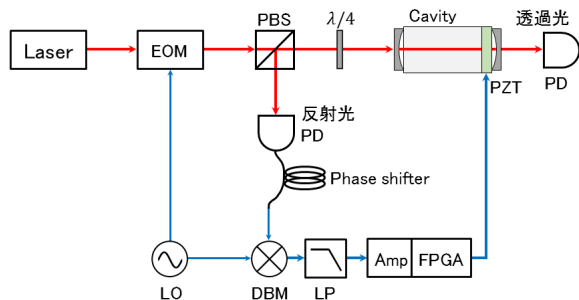


図10 FPGA を用いた共振器長安定化の実験系

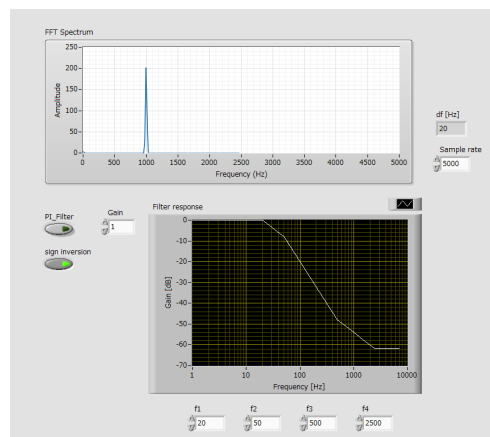


図11 共振器安定化フィルタの制御画面

位相を調節した。また、EOM の変調周波数成分をローパスフィルタでカットした後で、分解能を上げるため、増幅してから FPGA に入力した。

図11が、開発した汎用フィルタの制御画面である。上部のグラフが入力 (誤差信号) の高速フーリエ変換 (FFT) の結果である。図11では、分かりやすいように周波数 1kHz の正弦波を入力している。下部のグラフは、現在のデジタルフィルタの特性を表示している。これにより、周波数ノイズスペクトルを見ながら、フィルタ特性を変えることで、最適なフィルタを簡単に作製できる。通常時は0Vを出力し、[PLFilter] がオンになると、制御信号が出力される。制御対象によっては信号を反転させる必要があるため、[signinversion] ボタンで、信号を反転させて出力するか反転させないかを変更できる。

低周波領域でゲインを稼ぎつつ高周波領域では位相を戻すため、ラグリードフィルタを2段にして図12に示す形のフィルタで共振器長をロックした。

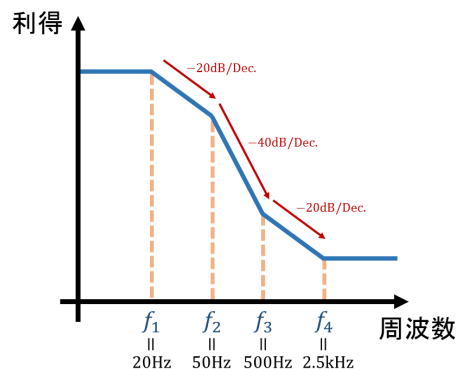


図12 使用した2段のローパスフィルタの概要図

図 13 は、図 12 のフィルタでロックしたときの周波数ノイズスペクトルである。

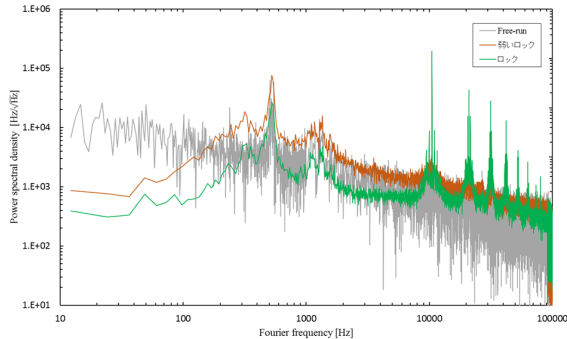


図 13 ロック時の周波数ノイズスペクトル (ラグリードフィルタ 2 段)

図 13 より、10kHz, 20kHz, 30kHz, 40kHz, 50kHz, ... で発振していることが分かる。この発振周波数は、リード周波数を変え、位相やゲインを変えても、変化しなかったため、PZT の機械共振だと考えられる。この発振のため、ゲインを図 13 以上に上げられず、制御帯域も制限されている。図 13 の発振周波数は、正確には 10.534kHz, 21.069kHz, 31.604kHz, 42.138kHz, 53.198kHz, ... であり、20kHz 以上の信号は 10.534kHz の高調波が検出されていると考えられる。機械共振だとすると、10kHz で位相が 180° 回っているため、位相を戻すことでの解決は難しい。そこで、低周波のゲインを上げたときでも 10kHz でゲインを 1 未満にすることで対処した。図 12 の 2 段のラグリードフィルタに図 9 に示した周波数 10.534kHz のみを除去するノッチフィルタを組み合わせ、図 14 となるフィルタを作製した。

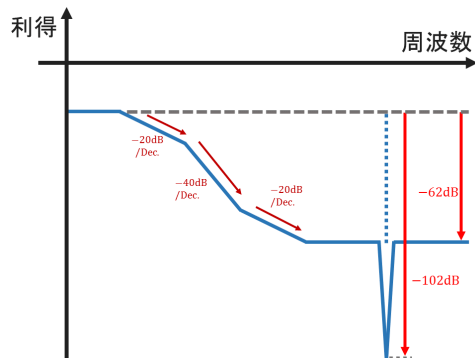


図 14 共振器長ロックのため設計したフィルタの概要図

このフィルタを用いてロックした結果が図 15 である。10kHz での発振が抑えられたためゲインを上げることができ、制御帯域も 37kHz まで上げることができた。

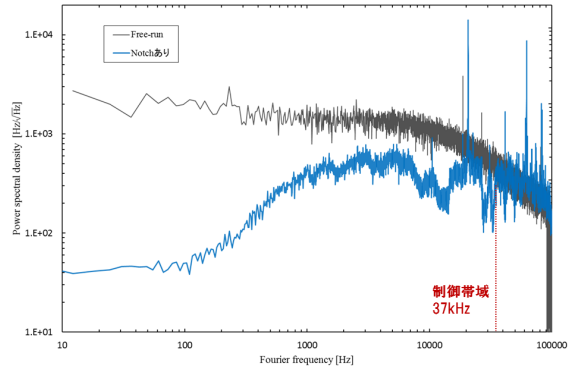


図 15 ロック時の周波数ノイズ (Notch フィルタ追加)

また、ロック時のアラン分散は、図 16 のようになった。傾きは -2.13 であり、共振器長のドリフトを抑えることができたと言える。

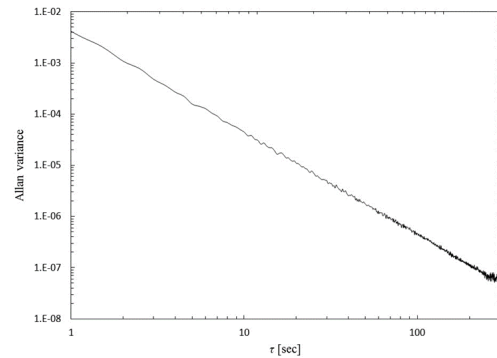


図 16 誤差信号のアラン分散

6 共振器長の自動ロック

今までの共振器長安定化の実験系は、PZT に加えている信号を詳しく書くと、図 17 のように三角波を

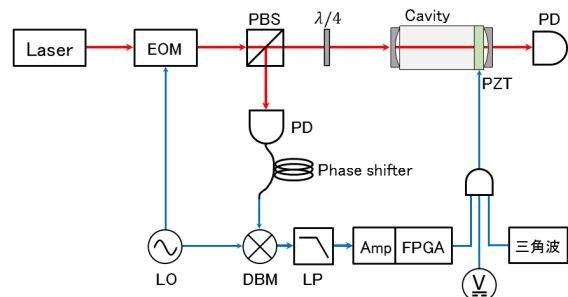


図 17 今までの共振器長安定化実験系の詳細図

外部から加算して、ロック点を手動で探し、見つかったロック点を維持するようにバイアスをかけた状態で、FPGA の制御信号も加えることで、ロックしていた。本節では、この手動部分を廃し、スイッチ一つを押すだけで、自動でロック点を捜しロックまで行うシステムの構築について述べる。このときの実験系では、実際に FPGA の出力のみを図 10 のように PZT に加えている。ロック点を捜す信号には、制御信号に利用している誤差信号（反射光）を用いた。ただし、反射光を用いた誤差信号は、図 18 の青線のような信号が得られ、ロックの確認には不向きである（赤点がロックしたい 00 モード）。一方、透過光では、00 モードにロックしたとき、他モードやモードの間に比べて強度が強くなり、正しい位置にロックしているかがはっきり分かる。そこで、ロックがかかっているかは、透過光で確認した。

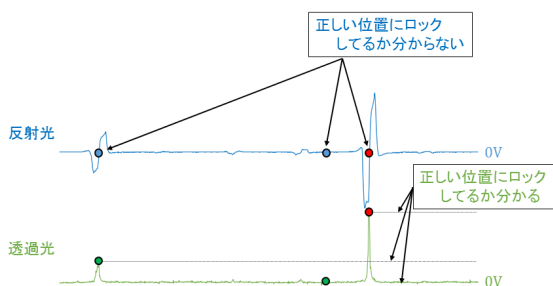


図 18 反射光と透過光の違い

自動ロックのフローチャートを図 19 に示す。

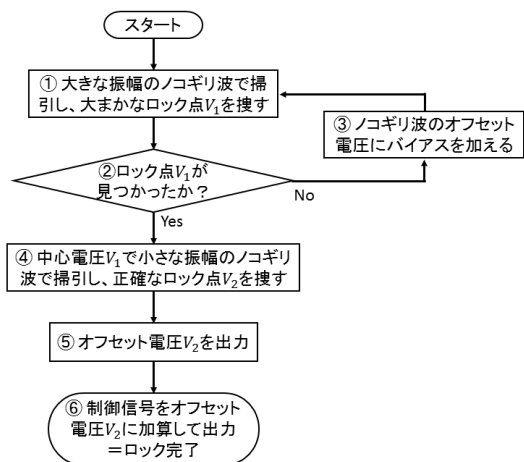


図 19 自動ロックのフローチャート

図 20 が、自動ロックをしたときの FPGA 出力と透過光の時間変化である。透過光が、ロックしたい 00 モード時の強度で安定化しているの、自動ロックできたと分かる。透過光の強度がロック後も揺らいでいるのは、PZT で追えない高周波のノイズである。これは、現在、フリーランで実験しているレーザーを安定化することで抑えることができると考えられる。

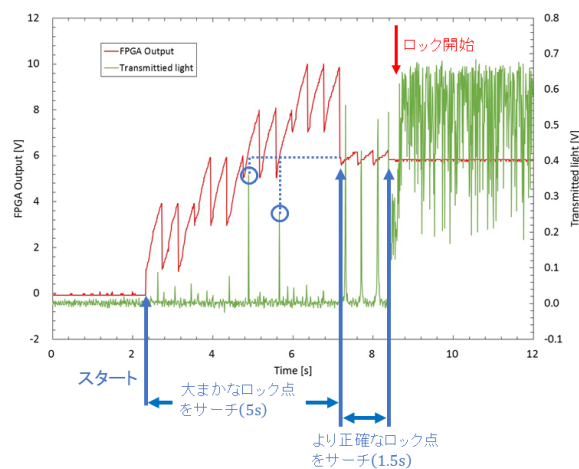


図 20 共振器長の自動ロック

7 まとめと今後の予定

汎用的なフィルタとして、周波数雑音を見ながらその場でフィルタ特性を変えて最適化でき、半田付け等の手間が要らず複製の容易なデジタル回路を FPGA を用いて作製した。これを用いて実際に今までアナログ回路で制御していた光学系の一部（共振器長）の安定化も行った。また、今まで手動で行っていたロックを自動化することができた。これにより実験がより効率良く行えるようになった。

今後は、より帯域の広い FPGA を用いて、LD の制御も含め様々な制御に拡張する予定である。

参考文献

- [1] D. Jaksh, et al., *Phys. Rev. Lett.* **85**, 2208 (2000).
- [2] D. W. Page and L. R. Peterson, "Re-programmable PLA". US4508977 (1985).
- [3] A. Schwettmann, et al., *Rev. Sci. Instrum* **82**, 103103 (2011)
- [4] G. Yang, J. F. Barry, et al., *J. Inst.* **7** P10026 (2012).