

Simple Antenna Positioner
簡易アンテナ・ポジショナ
武安義幸 / JA6XKQ

急がば“回せ”！

Sバンド・パッチ・フィードの測定を開始してみると、

- 方位角(AZ)を設定する
- レベルを読取る
- スプレッドシートに入力



写真-1 : 簡易アンテナ・ポジショナの外觀 … 単なるモータ

• グラフ化

という作業を、例えば方位角を 5度毎に設定すると、360度のデータを取得するのに 72回も繰り返すことになり、手作業での効率の悪さに直面してしまいました。パッチ・フィード自体が完成したものではなく、そのパラメータを変えて特性の変化を確認したいという本来の目的のために、上記の手作業を自動化することにしました。

仕掛けの中心はアンテナ・ポジショナ、つまりローテータです。これで効率的にアンテナを回して、レベル読取りも自動化してしまおうという魂胆です。回転させたいのは方位角と偏波角、これらの測定に共用できるものを目指しました。

自動化のために

自動化の仕掛けは、手動作業に対応して、次のよう構成されています。

- PC で制御されたアンテナ・ポジショナ
- レシーバの Sメータ電圧を読む DMM (デジタル・マルチメータ)
- PC で DMM の値をスプレッドシート(Excel)へ書込み
- スプレッドシートでデシベル dB 換算を行い、グラフ描画
- これらを統合する制御ソフトウェア

アンテナ・ポジション

手作業の場合には5度の読取りでも面倒と感じましたが、自動化するなら1度程度の分解能は欲しいと、欲が出てしまいました。

角度/位置制御としてはクローズド・ループ方式とオープン・ループ方式が考えられます。前者としては、DC モータで回転させてポテンシオメータ(の電圧)で角度検出、後者としてはステッピング・モータ(に与えるパルス数)で予め決めた角度へ回転、という方策を直ぐに思いつきます。今回は、工作とソフトウェアの簡便さから後者のステッピング・モータによるオープン・ループ方式としました。

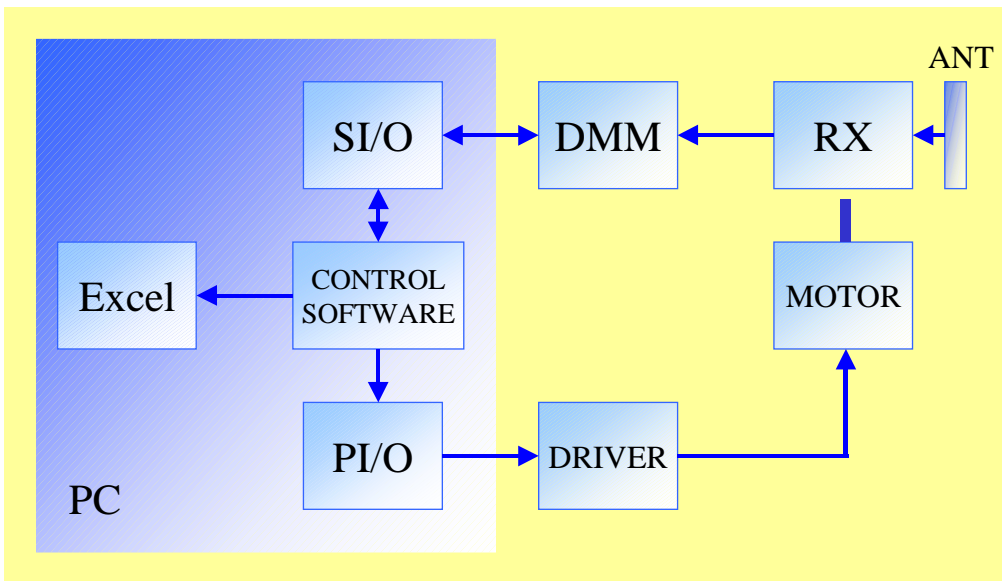


図-1 : 測定システムの構成

工具に限りある素人では、機構部品を自作することは不可能です。今回、特に苦慮したのは、減速機構と軸受けをどうするか?という点です。シートフィーダ・スキャナから分解した小型ステッピング・モータ単体の予備実験では、トルクと滑らかな回転の観点から、減速機構を介してのドライブが必須のように思われました。しかし、そこに拘ると、コストも時間も足りなくなってしまうと判断し、方針を修正。トルクがあり、軸受けがしっかりしたステッピング・モータを、単体で使用することにしました。

毎度おなじみの秋月電子通商で、「2相ユニポーラステッピングモータ(高トルクハイブリッド型)200ステップ」なる多摩川精機製の TS3103N124 が使えると判断し、購入。あいにく、データシートをインターネット上で見つけることはできませんでした。

購入後に気付いた点は、シャフト径が 9 mmであり、ギアやベアリング、軸受けなどで入手しやすい 6 mmではないことです。秋月電子通商のホームページによると、

ステップ数	: 200 (1.8 deg/Step)
駆動電圧	: 12 ~ 24 V
コイル抵抗	: 80 Ω/相
コイル電流	: 140 mA (12 V)
ホールディングトルク	: 6.5 kg・cm

との特性が掲げられています。分解能(ステップ数)は目標よりは粗いものの、今回の目的には十分でしょう。

このステッピング・モータのシャフトは9mmと太く、フィード・アンテ

ナ程度ならそのままサポートすることが可能と判断しました。幸い、シャフトの頭には 3 mmのタップが切っており、これにサポート金具を止めることができるので、軸受け周りの加工を一切せずに済ませることができました。ポジショナと格好よく言っても、ただのモータそのものです。

Windows上のソフトウェアで駆動パルスを発生させ、プリンタ・ポートから4ビットのパルス列としてトランジスタ・アレイでモータの4つのコイルを駆動しています。4ビットのパルス列は写真-2に示すようなパターンで、2つのコイルを同時に駆動する期間を設けた、トルク重視の2相駆動という方式です。下記のウェブサイトの説明が判り易く、参考としました。

<http://homepage1.nifty.com/rikiya/software/114stepping1.htm>

パルス列の発生は、ワンチップ CPU を用いる場合にはローテーション命令を使うのが常套手段だと思いますが、今回用いた Delphi ではシフト命令(右シフトの shr、左シフトの shl)しかなく、シフト命令を用いてローテーション命令を次のように行いました。変数の LastStatus がコイル駆動の4ビットのパルス列を示しています。

右ローテーション

```
LastStatus := (LastStatus shl 1) + (LastStatus shr (4-1));  
LastStatus:=LastStatus AND 15;
```

左ローテーション

```
LastStatus := (LastStatus shr 1) + (LastStatus shl (4-1));
```

```
LastStatus := LastStatus AND 15;
```

4ビットでのローテーションなので、最後に 15(バイナリで 1111)でマスクしています。

これは、下記のウェブサイトを参考としました。

<http://www.merlyn.demon.co.uk/del-bits.htm>

ステッピング・モータを駆動するパルス周波数が速すぎると脱調を生じてしまいますが、どこまで速くできるかは、実際に負荷を

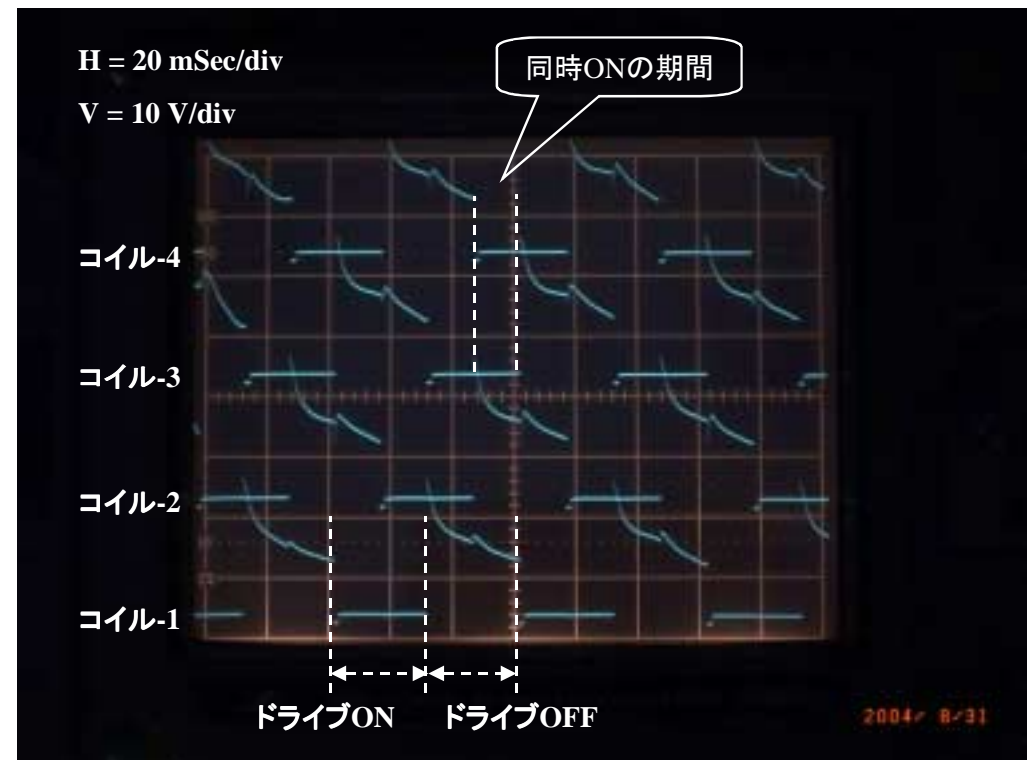


写真-2 : 駆動パルス

かけた状態で試しながら決定することにしました。試行錯誤により、事前にまったく考慮していなかった点が見つかりました。それは、負荷(試験物のフィード・アンテナとダウンコンバータ)の慣性とステッピング・モータのブレーキ動作の関係から、パルス毎に振動(発振)を生じてしまうことです。(写真-3)

この振動と駆動パルスの周波数が同期してしまうと激しい脱調を生じてしまいます。脱調のためにシングル・ステップでの動作も行なうことができません。これは一種の発振であり、系の固有振動周波数を避けた(高く/低く)駆動パルス周波数とすることで

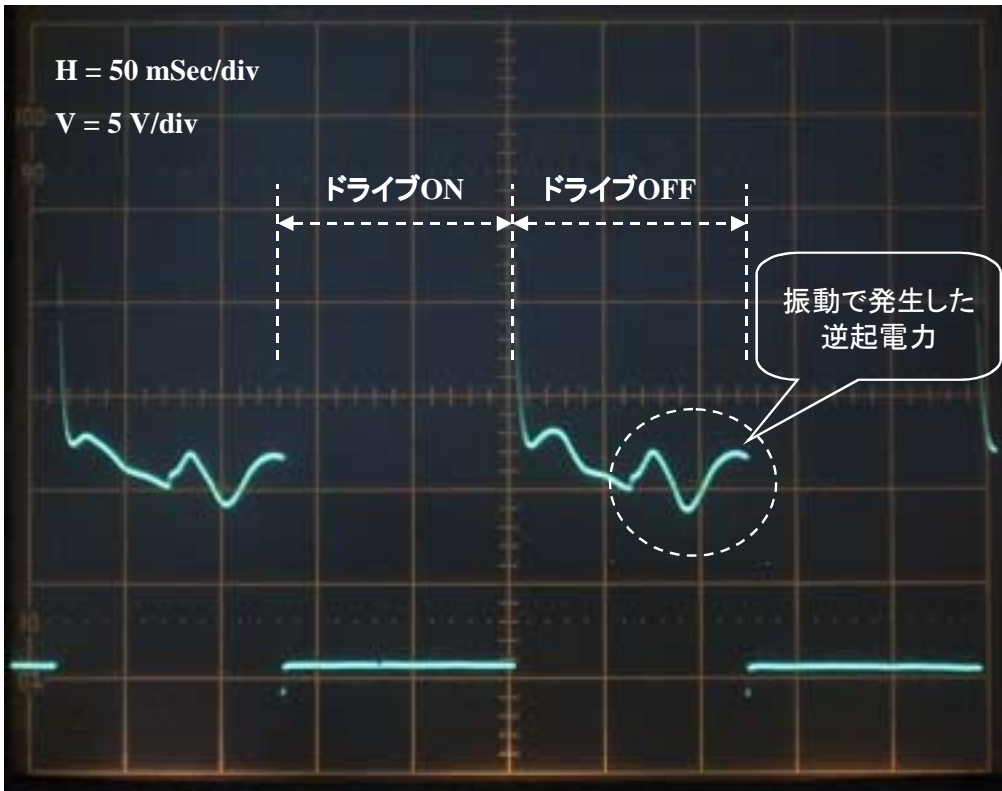


写真-3 : モータの駆動波形

対応できます。また、負荷に応じてモータへの印加電圧を調整することでも対応しました。フィードアンテナ程度では、定格の24 Vではモータのトルクが高すぎて振動が大きくなるので、12 Vまで印加電圧を下げて使用しています。関連して、始動時はパルス周波数を低く、連続回転時はパルス周波数を高く、そして停止時には再度パルス周波数を低くすることで、脱調を防止する工夫を施しました。

このあたりの試行錯誤は、いわゆるクロス開発環境のワンチップCPUとは異なり、開発ターゲットと開発環境が同一 PC と同一 OS (Windows2000) 上で行なえるため、効率の良い(と言うと聞こえが良いが、、、要は、行き当たりばったりの)作業を行なうことができました。

DMM(デジタル・マルチ・メータ)

使用している受信機 (AR8000) 自体に RS-232-C インターフェースが備わっており、これにより受信レベル(Sメータ値)を読み取ることが可能です。しかし、受信レベル対Sメータ値を校正してみると、分解能が荒いため、今回の目的には使用できないことが判明しました。

この結果を受けて、Sメータ値のアナログ電圧を受信機外に取り出し、外部でA/D変換を行なうことに。ISA カードとPCI カードのA/D 変換器を所有しているのですが、室外でのノートPCを用いての測定を想定して、取り回しの良いシリアル・インターフェース付きDMMを使うことにしました。用いた DMM は Metex 社の M3850Dであり、そのサンプリング速度が遅いことで巷の評判が

悪いことが気がかりでした。しかし、その簡便さの魅力と、購入以来、一度もシリアル・インターフェース機能を使用しなかったことがないので、「ここは使わねばなるまい」と決断した次第。と書くのは、やはり応答速度の遅さに後悔したからなのですが、、、応答速度の遅さに対応するために、制御ソフトウェアで約 1秒の待ち時間を入れると、360度 = 200ステップで 200秒となり、何もせずに待っているのは辛い長さです。(楽ができると贅沢になります)

DMM のプロトコルは至極簡単で、PC から ASCII で “D” を送出するとそれを受けて測定値を 14バイト固定長で返答します。今回は電圧値なので、返答の形式は “DC_3.999_V_CR” (_ はスペースを示す)であり、そのなかの数値だけを取り出すことになります。

このように簡単なプロトコルなのでフロー・コントロール無しで、TXD と RXD だけを単純に接続すれば事は足りるだろうと安易に考えていたところ、別の観点で落とし穴がありました。取り扱い説明書にはプログラム例があり、そこには RTS と DTR の設定が記述されています。しかし、その必要性がプロトコルからは見えてきません。予備実験として Windows 付属のターミナル・ソフトウェアで試してみるものの、返答が DMM から帰ってきません。インターネットで検索すると

<http://sigprak2.informatik.hu-berlin.de/multimeter/dm3830.shtml>

に答えがありました。M3850D シリアル・インターフェースをフォ

トカップラでアイソレーションしており、その電源として RTS と DTR を使用しているのです。したがって、電源電圧を与えるために RTS を -12 V に、DTR を +12 V に設定する必要があるのです。これで疑問が解け、ようやく DMM から電圧を読み取ることができました。

デシベル変換

受信機のSメータ電圧をアナログ値として、前述のように DMM を用いて PC に取込むことができました。次のステップは、Sメータ電圧をデシベル(dB)に変換することです。デシベル変換とグラフ描画はスプレッド・シートの Excel を使用します。全体を統合する制御ソフトウェアを Delphi で開発するので、それ自身でデシベル変換とグラフ描画を行なうこともできますが、データ取得後のグラフの各種加工を考慮すると、Excel を使用することが最良と考えました。また、Delphi で開発したソフトウェアから Excel を操作することの習作としての位置づけもあります。

デシベル変換においては、10 dBステップで校正した 6点のSメータ電圧を元にして、輻射パターン取得時のSメータ電圧をデシベルに変換します。10 dBステップの 6点の値から、Sメータ特性の近似式を求めます。Excel のグラフ機能には近似曲線を求める機能があります。しかし、多項式で近似した曲線の係数をグラフに表示することはできても、セル計算に取込むことはできないようです。また、Excel の近似曲線の精度には問題があることは「その筋」では有名な話で、私自身もその問題に遭遇したことがあります。

ということで、Excel のセル計算で自前の多項式近似計算を行なうことに。調べてみると、マイクロソフト自身が問題を認めて(?)おり、その対処法としての多項式近似計算を解説していました。下記のウェブサイトで詳細を知ることができます。

<http://support.microsoft.com/default.aspx?scid=kb;ja;405463>

図-2 は、受信機のSメータ特性を示しています。特性を取得するにあたっては、受信機入力レベルをステップアッテネータで可変して、その時のSメータ電圧を測定しますので、本来は横軸をアッテネータのデシベル値とし、縦軸をSメータ電圧とすべきです。しかし、目的はSメータ電圧からデシベル値を補間すること

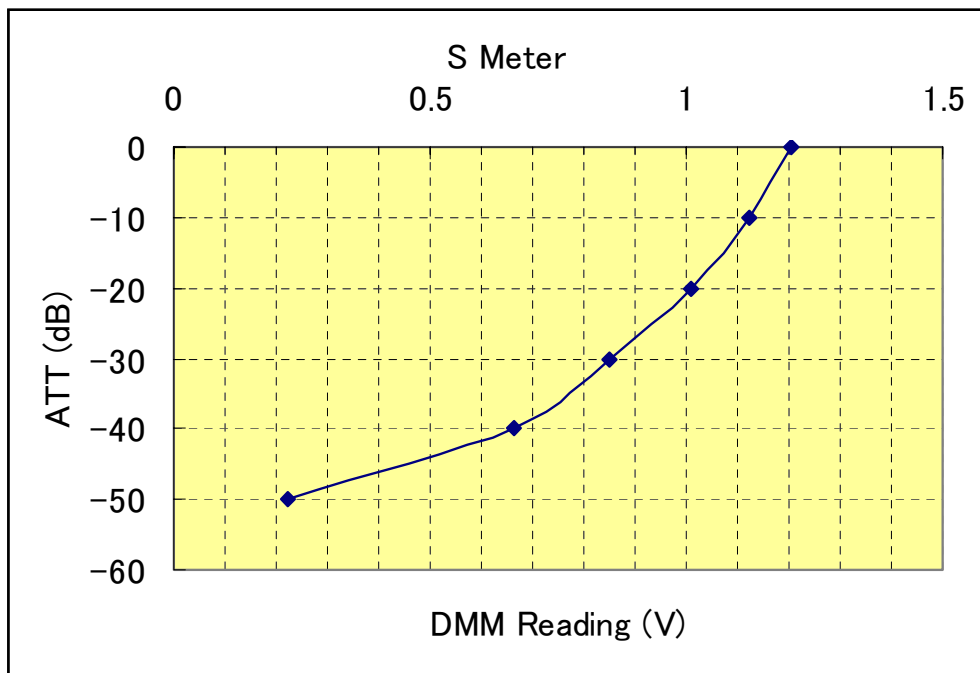


図-2 : Sメータ特性例

なので、軸の取り方を逆にしています。

この特性から判断して、3次の多項式で近似することにします。アッテネータ値をy、Sメータ電圧をxとして、次式で近似します。

$$y = ax^3 + bx^2 + cx + d$$

この多項式の係数 a、b、c と切片 d をマイクロソフトの解説にしたがって求めます。(LINEST関数を使用しますが、Excel のヘルプを読んでも、マイクロソフトのウェブサイトの解説とのつながりがさっぱり解りません、、、マイクロソフトのウェブサイトの解説に、ただ従うのみです)

計算の結果、図-2 に対する係数 a、b、c と切片 d は次のようになりました。

$$\begin{aligned} a &= 33.32898 \\ b &= -20.3282 \\ c &= 20.3859 \\ d &= -53.9232758 \end{aligned}$$

この結果を 図-2 で検算してみましょう。例えば、-30 dB のアッテネータ値では、Sメータ電圧は 0.849 V でした。多項式近似の x に 0.849 を代入すると、y すなわちアッテネータ値は -30.8722 と計算されました。この程度の計算エラーがあることを承知しておきましょう。

以上の計算を輻射パターン取得時に Excel で行います。

グラフ描画

Sメータ電圧を元にデシベル変換(前項)とグラフ描画を行なう Excel の「ブック」の雛形を予め作っておき、制御ソフトウェアからそのブックにアクセスして、輻射パターン取得と同時にグラフを作成させます。

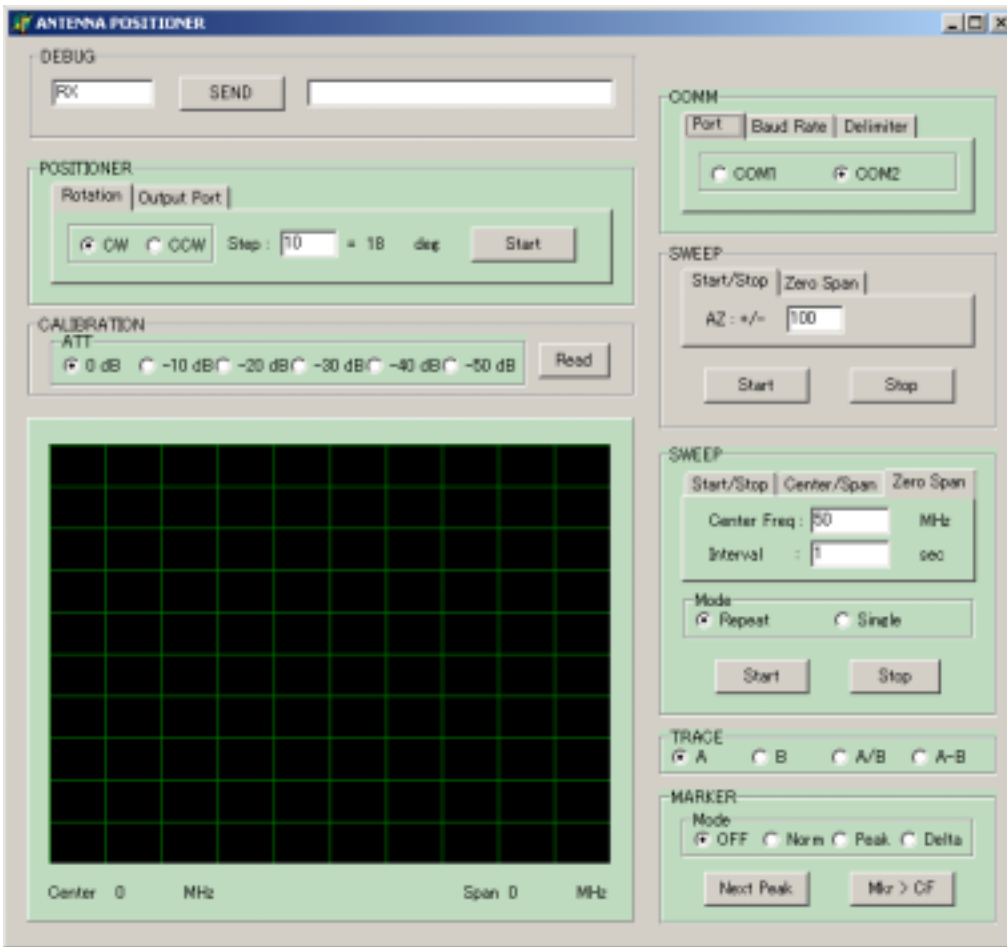


図-3 : 制御ソフトウェアの GUI

制御ソフトウェア

Delphi で作成したソフトウェアで、前述のステッピング・モータの制御、DMM からの電圧取込み、Excel への書出しを行ないます。ソフトウェアの GUI を 図-3 に示します。受信機 (AR8000) を使っての“スペアナもどき”のソフトウェアから改変しましたので、今回は使用しない操作部が残っている点に注意してください。

操作部としては、ステッピング・モータを単独で操作する部分、Sメータ電圧のキャリブレーション部分、そしてそれらを統合してアンテナの輻射パターンをスイープする部分から構成されています。

Delphi から Excel へアクセスする手法は、下記のウェブサイトを参考にしました。

<http://www.borland.co.jp/qanda/delphi/d0003225.html>

実に簡単にアクセス可能であり、「案ずるより、、、」でした。制御ソフトウェア側で Excel シートをクローズする際には、Excel 側でファイルをセーブするか否かのダイアログが自動的に出てくるので、操作についての細かい配慮を個別にする必要がない点が、既存のアプリケーションと連携できる魅力の一つだと感じました。

測定例

測定例は、Sバンド・パッチ・フィードの実験を参照してください。
手動の場合は角度ステップを5度と荒くしていたため、パターンの細かい変化を見落としていたかもしれません。しかし、このシステムを使うことで、1.8度ステップでの取得が可能となり、ほぼ連続的なパターン取得ができるようになりました。

