ANALOG DEVICES

TNJ-084

アナログ電子回路技術ノート

足す引くで三角関数計算ができちゃうデジタル演算 CORDIC（前編）

ながく理解に挫折していた経験から CORDIC を分かりやすく解説してみたい

著者: 石井 聡

現実世界はアナログかデジタルか

新緑の息吹を感じる季節（図 1）、ワタシはふと思いました。「さて、現実世界はアナログなのかデジタルなのか」。アナログ・デバイセズという社名はその名のとおり、アナログである物理量を取り扱うという意味があります。

「電圧量や電流量はアナログ値」とはいいますが、新緑の息吹を感じるこのアナログな季節に、「電子の電荷量eは

$$e = -1.6 \times 10^{-19}\ \mathrm{C}$$

だよな。0.00016pA の電流量は電子

$$n = \frac{Q}{e} = \frac{It}{e} = \frac{16 \times 10^{-17}}{1.6 \times 10^{-19}} = 1000$$

個が、1 秒間に導体中を移動することになるわけだ！」と思いました。1000 個、つまり離散値として数えることができるわけですから、「電流量はデジタル量といえるな！」に至りました。

また PN 接合を電流I [A] （電子）が流れるときに生じるショット雑音[1]の 1Hz 電流密度

$$i_N = \sqrt{2eI}$$

も、電子 1 個が PN 接合を横切るときに発生する電流性ノイズの合算ですので、「電流量はデジタル量」という考えを補強するものといえるでしょう。

図 1. コロナ禍も 1 年が経過し、あらためて新緑の季節となった。その昨今、妻がガーデニングに覚醒し、いろいろと始めた。この写真は種から発芽させたミニ・ひまわりの新芽。それを見ている私は、明るい新緑の季節に何をしようかと思いめぐらすところだが、アナログ技術セミナー2021 のネタ仕込みでも暗い自室でし始めるかと、なんともはやの思考回路（笑）

ところでアナログ・デバイセズでも電流検出回路のことを「クーロン・カウンタ」（カウンタ！）とか呼んだりしますので、これも驚きをもって受け入れています（笑）。

でも量子力学まで考慮すると、どうなるんでしょうね。それでも上記の考え方でふるまいの連続性がいったん途切れるのでしょう。詳しい方で素人にも分かりやすくご説明いただける方、ぜひご一緒にお茶のみでもいたしましょう！

CORDIC を分かりやすく解説してみたい

CORDIC（COordinate Rotation DIgital Computer）はデジタル演算処理で三角関数など（調べてみると、これがまた、かなり多彩な演算を実現できるようで、びっくりですが）を、簡易な論理回路構造で実現できるアルゴリズムです。関数電卓内のデジタル回路の論理構造（回路でなくソフトウェア演算のようですが…。詳しくは分からず…）にも活用されていますし、マイコンなどにもペリフェラルとして CORDIC 演算回路が内蔵されています。

もう少し言っておくと、CORDIC での演算は足し算、引き算、ビット・シフトを繰り返すことで実現できるものです。デジタル回路では実現しやすいアルゴリズムなのです。

私も CORDIC については、名前と基本的な考え方は知ってはいましたが、具体的なアルゴリズムやインプリメント（実装）方法は分かっていませんでした。これまでも適切な記事を探していましたが、数点ほど読んでみても、いまひとつ理解できませんでした。しかし参考文献[2]を見つけて「なるほど！」となりました。その CORDIC の理解が浅かったころ、とあるサイトに GNU の VHDL ソースがありました。それを使って実際の実験もおこなったこともありました（次の技術ノートでその実験結果も示します）。

今回と次回の技術ノートでは、ネットに多数解説記事は出ているものの、どうも理解不能というか腹落ち感の低い、この CORDIC を、我ながらの手法で解説してみようと思って書き始めております。いつもながら、書き始めた段階では、まだトンネルの出口には到達していない、手探り状態というところです。

CORDIC の計算手順（まずは答えの得かたのご紹介）

これまでいろいろな記事を読んでも良く分からなかった（というよりは読み始めてすぐに挫折していた）CORDIC は、実はとても簡単な計算手順なのでした。

ところで今回の技術ノートでは、文章と数式の説明をビジュアル的にも理解いただけるように、説明の部分付近に、その説明に該当する図を挿入するレイアウトにしてみました。いくばくかでもご理解が深まれば幸いです。

Rev. 0

アナログ・デバイセズ株式会社

本　　社／〒105-6891　東京都港区海岸 1-16-1　ニューピア竹芝サウスタワービル
電話 03（5402）8200

大阪営業所／〒532-0003　大阪府大阪市淀川区宮原 3-5-36　新大阪トラストタワー
電話 06（6350）6868

初期値の設定

目的の角度をθとし、その SIN = 正弦、COS = 余弦を生成することを考えます。CORDIC 計算開始の初期値（$i = 0$）として

$$\left.\begin{aligned} x(0) &= \frac{1}{A_N} \\ y(0) &= 0 \\ z(0) &= \theta \end{aligned}\right\} \quad (1)$$

と設定します。詳しくは以降に示しますが、これは図 2 のイメージになります。$z(0)$に目的の角度θを設定するわけなのですね…（これがポイント）。図 2 では$z(0) = 60^\circ$と仮定しています。

以降で詳しく説明していきますが、十分な繰り返し演算が行われると、答えとして

$$\left.\begin{aligned} x(N) &= \cos z(0) = \cos\theta \\ y(N) &= \sin z(0) = \sin\theta \\ z(N) &= 0 \end{aligned}\right\} \quad (2)$$

が得られるというアルゴリズムです。答えとしては$z(N)$は不要なものです。

ここでA_Nはスケーリング係数と呼ばれる定数なのですが、これは以降で示します（繰り返し回数が十分おおきければ 1.64676…に収束するものです）。この$x(0)$に$1/A_N$を設定することなく、

$$x(0) = 1$$

を設定して、最後に得られた答えをA_Nで割っても「数学的」には問題ありません。しかし実デジタル回路での計算回路規模を減らす目的からすれば、このやり方だと「本末転倒的」になってしまいます。

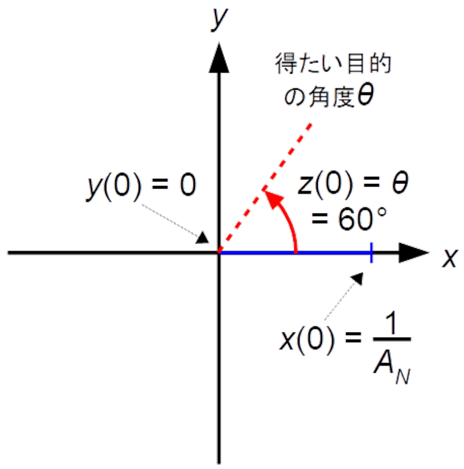


図 2. CORDIC での各パラメータ初期設定のイメージ

ステップ演算を繰り返す漸化式

つづいてiがゼロからスタートする漸化式（漸化式とは、ひとつ前のステップの演算結果を用いてそのステップでの計算結果を得て、それを繰り返すもの）

$$\left.\begin{aligned} x(i+1) &= x(i) - D(i) \times [y(i) \gg i] \\ y(i+1) &= y(i) + D(i) \times [x(i) \gg i] \\ z(i+1) &= z(i) - D(i)\tan^{-1}\left(\frac{1}{2^i}\right) \\ &= z(i) - D(i)\cdot s(i) \end{aligned}\right\} \quad (3)$$

を考えます。$x(i), y(i), z(i)$は単なる数値（デジタル値）です。$D(i), s(i)$は以降で示します。図 3 に、1 回目の計算を直交座標としてグラフィカルに表現したようすを、そして図 4 に 2 回目の計算のようすを、それぞれ図 2 の設定［$z(0) = 60^\circ$］を例にして示します。

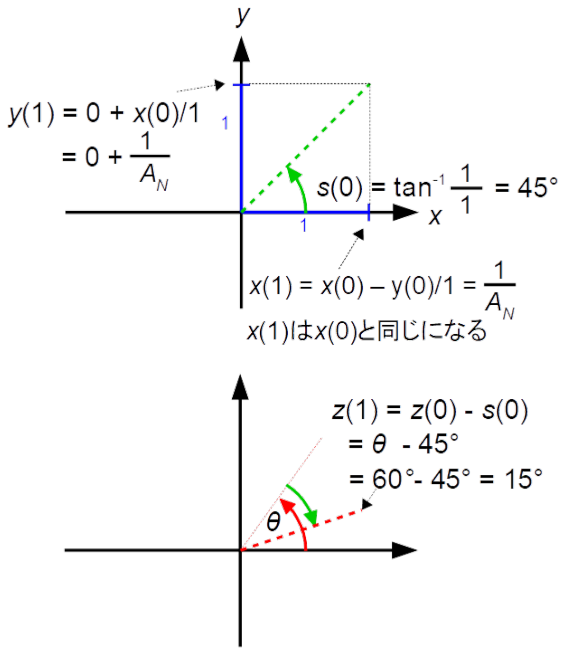
$D(i)$は CORDIC の回転方向（極性）で

$$D(i) = \begin{cases} -1, & z(i) < 0 \\ +1, & \text{otherwise} \end{cases} \quad (4)$$

$z(i)$がマイナスなら$D(i) = -1$、$z(i)$がプラスなら$D(i) = +1$とするだけです。図 3、図 4 においては$D(0) = +1, D(1) = +1$という設定になっています［$z(0)$が 45° を超えている条件なので］。

$\gg i$はiビットの右シフト演算です。もっと数学的に書くと

$$\left.\begin{aligned} x(i+1) &= x(i) - D(i) \times \left[\frac{y(i)}{2^i}\right] \\ y(i+1) &= y(i) + D(i) \times \left[\frac{x(i)}{2^i}\right] \end{aligned}\right\} \quad (5)$$



3. CORDIC での繰り返し 1 回目の計算（青字は比率）

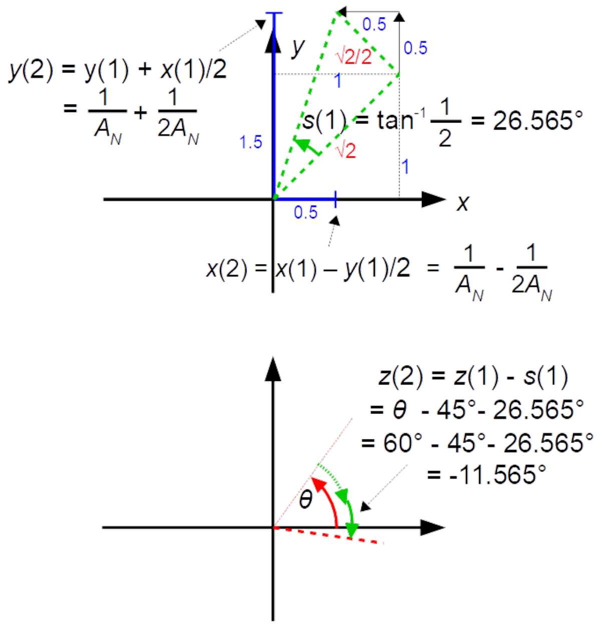


図 4. CORDIC での繰り返し 2 回目の計算（赤字・青字は比率）

とも表現できます。式(3)の$z(i)$の式中や、図 3 や図 4 に

$$s(i) = \tan^{-1}\left(\frac{1}{2^i}\right) \tag{6}$$

がありますが、これは 1 ステップの演算における角度変化量です。これはあらためて以降で説明します。

$s(i)$のイメージを図 5 に示します。ここでは$s(0), s(1), s(2)$を求めるようすをビジュアルに示しています。

デジタル回路の実装においては、$s(i)$は計算済みテーブルとして事前に用意しておけばいいだけです。テーブル$s(i)$の単位（たとえば弧度法なのかラジアンなのか、ビット表記や小数点位置なども含めて）は、$z(0)$に設定する目的の角度θの単位と同じにしておく必要があります［$s(i)$は$z(i)$と足し引きされるため］。これは重要なポイントです。

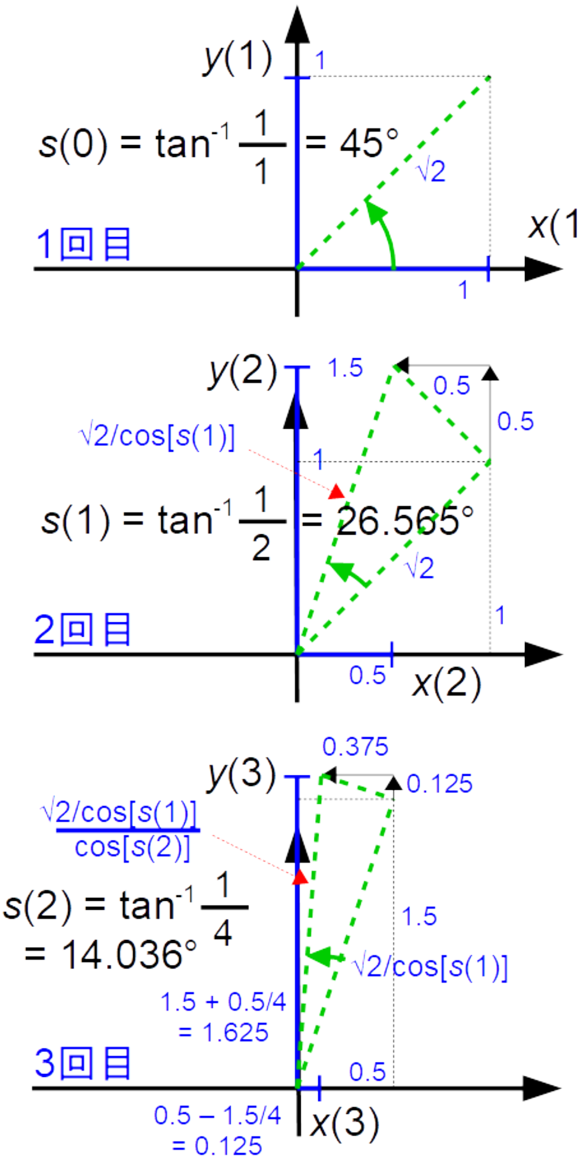


図 5. CORDIC でのテーブル値$s(i)$をビジュアルに理解する［繰り返し 1 回目、2 回目、3 回目での$s(i)$の値の求め方。ただし**それぞれの演算で$z(i)$がプラス、つまり$D(i)$がプラスの条件としたもの**。図 4 は$D(2)$がマイナスとなる条件。青字は比率］

繰り返し回数を十分に大きくすれば答えが得られる

ここでiを十分大きく、つまり演算繰り返しを十分な回数Nにすると（$i+1=N$と表します）、

$$\left.\begin{aligned} x(N) &= A_N[x(0)\cos z(0) - y(0)\sin z(0)] \\ y(N) &= A_N[y(0)\cos z(0) + x(0)\sin z(0)] \\ z(N) &= 0 \end{aligned}\right\} \tag{7}$$

に収束します。この式を理解しようとすると少し難儀するかもしれませんので、ひとまずスルーして OK です（今回の後半と次回に詳しく説明します）。

実際の CORDIC の使い方として sin/cos を生成する目的であれば、最初の説明のように、それぞれの初期値を$x(0) = 1/A_N$, $y(0) = 0$とし、また$z(0) = \theta$として目的の角度θを設定します。そうすると上記の式(7)は、

$$\left.\begin{aligned} x(N) &= \cos z(0) = \cos\theta \\ y(N) &= \sin z(0) = \sin\theta \\ z(N) &= 0 \end{aligned}\right\} \tag{8}$$

となり、N回の繰り返し演算でこれらが得られるというお話です。このようすを図 6 に示します。

繰り返し回数Nは、テーブル値$s(i) = 0$になるところ（デジタル演算として 1 ビット以下になるところ）を目安とします。$s(i)$は事前計算済みテーブルなので、$s(i) = 0$になるところは事前に既知、すなわち繰り返し回数Nは事前に既知となります。これが CODRIC の計算、うごきなのです。

なお$z(0)$として対応可能な範囲は$-\pi/2 \le z(0) \le \pi/2$になるので、他の象限部分で計算をさせたい場合は、初期化時におまじないが必要です。詳細は参考文献[2]の p. 3 左コラム中央あたりをご覧ください。

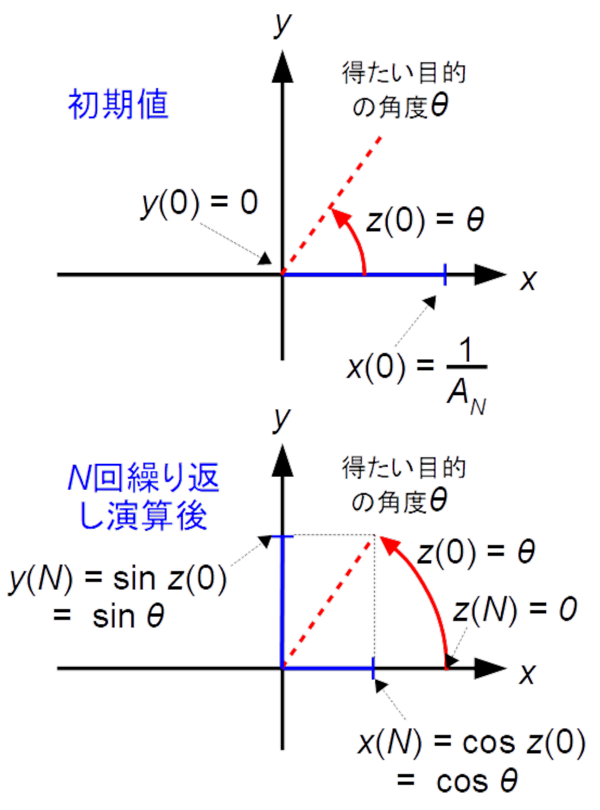


図 6. CORDIC で$x(0) = 1/A_N, y(0) = 0, z(0) = \theta$を初期値として$N$回演算の結果として得られるこたえ

スケーリング係数A_Nを考える

さきに出てきた定数A_Nは図 7 に示すように、CORDIC の漸化式を逐次的にステップごとに計算していくうえで生じる、図 7 の緑色破線の長さが増加する係数$A(i)$の繰り返し積で、

$$A_N = \prod_{i=0}^{N} \sqrt{1+2^{-2i}} = \prod_{i=0}^{N} A(i) \qquad (9)$$

Nが十分大きければ 1.64676⋯に収束します。ここでΠという記号は、要素ごとをすべて掛け算する「総乗」という意味です[3]。総和Σの掛け算版といえます。このA_Nはスケーリング係数と呼ばれます。これは次回詳しく説明します。

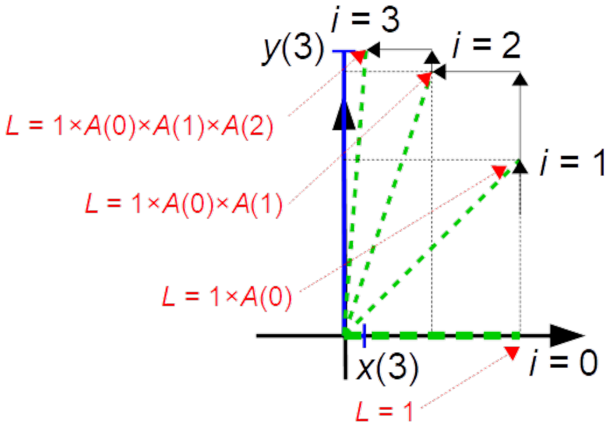


図 7. スケーリング係数A_Nの意味合い。Lは長さを意味する記号（ただし**それぞれの演算で$z(i)$がプラス、つまり$D(i)$がプラスの条件とした**もの。図 4 は$D(2)$がマイナスとなる条件）

実際にエクセルで計算してみる

演算論理ブロックをお見せするまえに（これは次回にご説明します）、この計算がホントに正しく実現できるかエクセルを使ってやってみましょう。

図 8 はエクセルの表計算上で CORDIC を計算してみたようすです。この演算では目的の角度θを 57° として、$z(0) = 57^\circ$とします（図中の 2 行目を黄色でハイライトしています）。またここまでの説明のように$x(0) = 1/A_N, y(0) = 0$とします。

なお$A_N = 1.64676 \cdots$なので、$x(0) = 1/1.64676 = 0.607253$となります。

「計算済みテーブル$s(i)$」は、ここではエクセルで

$$s(i) = \tan^{-1}\left(\frac{1}{2^i}\right) \qquad (6)\text{ 再掲}$$

をそのまま計算させています。

$N = 16$で 16 回の漸化式演算（ステップごとの演算）を行った結果が表中の一番下、$i = 16$のところです（黄色でハイライトしています）。表中の結果$x(16)$、$y(16)$が、cos(57°)、sin(57°)に近くなります。それぞれのセルの欄外下にエクセルで計算した cos(57°)、sin(57°)の結果を示してありますが、かなり近くなっていることが分かります。

これで CORDIC の計算が、説明した方法で正しくできることを確認できました。

i	$x(i)$	$y(i)$	$z(i)$	$D(i)$	$s(i)$
0	0.607253	0	57	1	45
1	0.607253	0.607253	12	1	26.565051
2	0.3036265	0.9108795	-14.56505	-1	14.036243
3	0.5313464	0.8349729	-0.528808	-1	7.1250163
4	0.635718	0.7685546	6.5962086	1	3.5763344
5	0.5876834	0.808287	3.0198743	1	1.7899106
6	0.5624244	0.8266521	1.2299637	1	0.8951737
7	0.5495079	0.83544	0.3347899	1	0.4476142
8	0.5429811	0.839733	-0.112824	-1	0.2238105
9	0.5462613	0.837612	0.1109863	1	0.1119057
10	0.5446253	0.8386789	-0.000919	-1	0.0559529
11	0.5454443	0.838147	0.0550335	1	0.0279765
12	0.5450351	0.8384134	0.027057	1	0.0139882
13	0.5448304	0.8385464	0.0130688	1	0.0069941
14	0.544728	0.8386129	0.0060747	1	0.0034971
15	0.5446768	0.8386462	0.0025776	1	0.0017485
16	0.5446513	0.8386628	0.0008291	1	0.0008743

↓ ↓

0.544639 0.8386706

cos $z(0)$ sin $z(0)$

図 8. CORDIC をエクセルで計算してみた（$N = 16$）

CORDIC の理論解説をしてみよう

ネットには各種の理論解説があるが

ネットで CORDIC の理論解説の記事を検索してみると、たくさん出てきます。多くが三角関数の回転行列（これは以降でも使っていきたいと思っていますが）や直角三角形をつなぎ合わせた説明になっています。読む気がない私が悪いのでしょうが（笑）、「簡単にわかる」とか「誰でもわかる」とか書いてある記事のわりには、どれもすっと腑に落ちてきません。

そこでこの WEB ラボでは、独自の切り口（？）と表現方法で、この理論解説をしてみたいと思います。

糸口「その 1」は$z(i)$を残留角度誤差量と考えること

ものごとを理解していくには、何かを「糸口」として見つけることがポイントです。この CORDIC の理解も同じでしょう。そういう私もこの技術ノートを書き始めながら、まず、そこを探っていきました。

一番重要と思われる糸口が、$z(i)$は、計算していくうえでの「角度の残留角度誤差量」だと考えることです。$z(i)$は

$$\left.\begin{aligned} z(i+1) &= z(i) - D(i) \cdot s(i) \\ s(i) &= \tan^{-1}\left(\frac{1}{2^i}\right) \end{aligned}\right\} \quad (3, 6)\text{ 再掲}$$

であり、引き算される$s(i)$が 1 演算ステップごとで小さくなり、$z(i)$から引き算されていきます。図 8 の最初の 3 ステップの演算を例にして、$z(i), s(i)$の変化のようすを図示したものを図 9 に示します。目的とする角度は 57° つまり、$z(i) = 57^\circ$です。図 8 においても、演算ステップを追うごとで残留角度誤差量である$z(i)$が低減していくことが分かります。

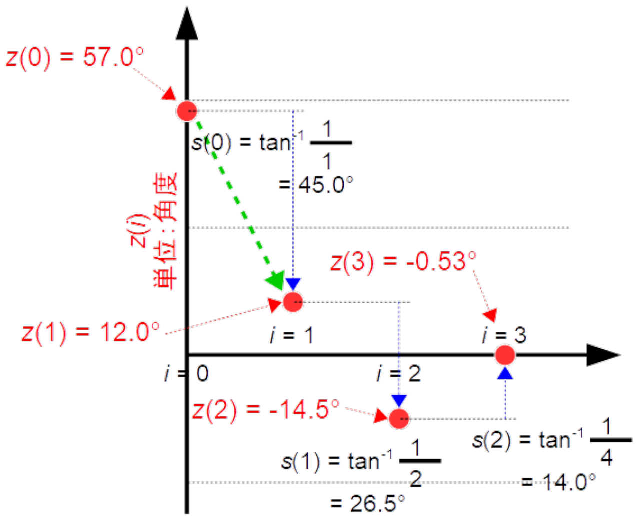


図 9. 図 8 の最初の 3 ステップの演算を例にして、$z(i)$が低減していくようすを示してみた（数値は有効数字 3 桁で表示）

$z(i)$自体の極性がプラスかマイナスかで、$s(i)$が$z(i)$から引かれるか足されるか（極性）が$D(i)$として決まります。この「引かれるか／足されるか」という表現も「残留角度誤差量を低下させていく操作」とみることもできます。

また演算繰り返しを十分な回数Nにすると

$$z(N) = 0 \qquad (8) \text{ 再掲}$$

に収束すると説明しました。ここからもさらに「$z(i)$は計算していくうえでの残留角度誤差量」ということが理解できます。

糸口「その 2」変化角度$s(i)$は直角三角形として決まり、その底辺が 1 演算ごとで 2 倍づつになっていく

つづいて減算量$s(i)$です。$s(i)$は 1 演算ステップごとで小さくなっていきます。$s(i)$自体の最初の 3 ステップでの値が、どのようにグラフィカルに表されるかを図 10 に示します。$s(i)$は

$$s(i) = \tan^{-1}\left(\frac{1}{2^i}\right) \qquad (6) \text{ 再掲}$$

ですから、

- 1 回目の演算ステップ、$i+1=1$（$i=0$）では、カッコ内は 1/1 となり、$s(0)$は図中の直角二等辺三角形（赤紫色／マゼンタ色）内に示している角の角度（45°）になる
- 2 回目の演算ステップ、$i+1=2$（$i=1$）では、カッコ内は 1/2 となり、$s(1)$は図中の底辺が 2 倍となる直角三角形内に示している角の角度（26.565°）になる
- 3 回目の演算ステップ、$i+1=3$（$i=2$）では、カッコ内は 1/4 となり、$s(2)$は図中の底辺が 4 倍となる直角三角形内に示している角の角度（14.036°）になる

このように$s(i)$は、1 演算ごとで底辺が 2 倍づつになっていく直角三角形の鋭角の角度なのです。

なぜこのように$s(i)$を「2 倍づつになっていく直角三角形」の鋭角の角度とするのでしょうか。それこそ 45°、(45/2)°、(45/4)°…というようにしていけばよいのでは？と頭をよぎるかと思います。

このようにする理由は、CORDIC の 1 演算ステップごとで、iビットのシフトにより演算を実現する（実際は$1/2^i$にする）ために必要な角度設定だからです。

詳しくは次回に示していきますが、このことは「ホントCORDIC、よくできてるな！」と思わせるポイントでもありました。次回の説明をご期待ください！

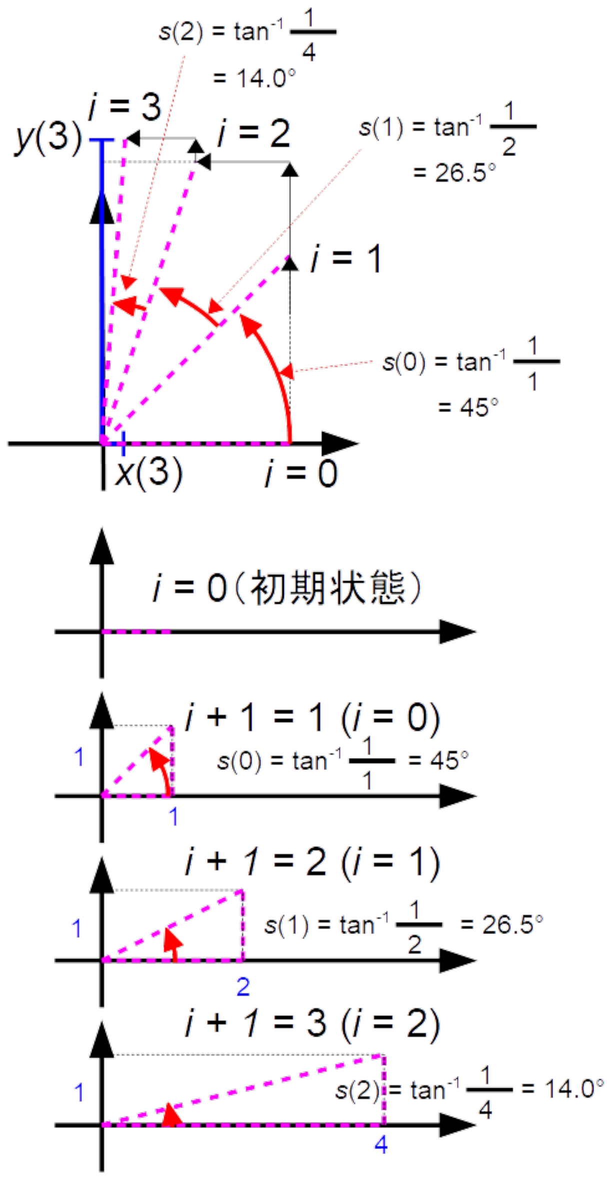


図 10. $s(i)$は底辺が 1 演算ごとで 2 倍になっていく赤紫色／マゼンタ色の直角三角形の鋭角の角度（**それぞれの演算で$z(i)$がプラス、つまり$D(i)$がプラスの条件とした**もの。数値は有効数字 3 桁で表示。図 4、図 9 は$D(2)$がマイナスとなる条件）

糸口「その 3」なんとこの計算は逐次比較（SAR）型 ADC の動作と同じではないか！

糸口「その 1」のように、残留角度誤差量$z(i)$と減算量となるテーブル値$s(i)$は、

$$\left.\begin{aligned} z(i+1) &= z(i) - D(i)\cdot s(i) \\ s(i) &= \tan^{-1}\left(\frac{1}{2^i}\right) \end{aligned}\right\} \text{(3, 6) 再掲}$$

であり、減算量$s(i)$は 1 演算ステップごとで小さく、そしてその減算極性は$z(i)$自体の極性がプラスかマイナスかで決まります。

これはまるで図 11 のような天秤と分銅による重量計測（重さが 1/2 の分銅を載せたり下したりする）と同じということに気がつくかと思います。さらにこれは図 12 のような逐次比較（Successive Approximation Register; SAR）型 ADC の動作と同じという気づきに至ることになると思います。

このように考えていけば、「CORDIC もそんなにややこしいものでもないな」と、気持ちの敷居を下げられるのではないでしょうか。

図 11. CORDIC の演算手順は天秤と分銅による重量計測と同じ（Old two pan balance, Nikodem Nijaki, Jan. 29 2011, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Balance_scale_IMGP9728.jpg?uselang=ja）

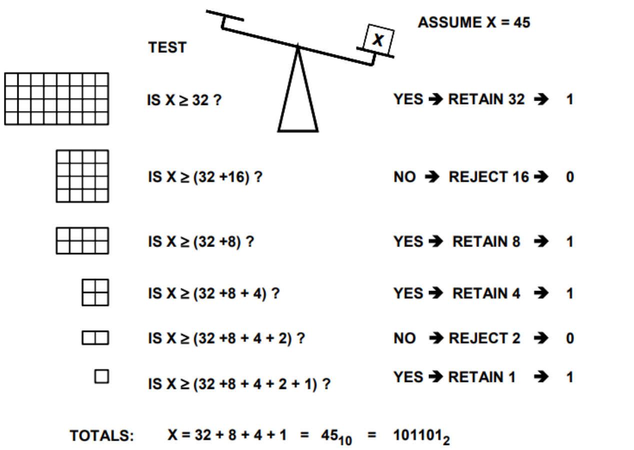


***Figure 3.59:** Successive Approximation ADC Algorithm*

図 12. CODRIC の演算手順や天秤と分銅による重量計測は SAR ADC の変換手順（Analog Devices, The Data Conversion Handbook, 2005 [4]の Figure 3.59 より）

糸口「その 4」は別に見出しを立てて

この糸口「その 4」の説明は少し長くなりますので、いったん別に見出しを立てて別の節として取り扱ってみましょう。

糸口「その 4」高校のときに習った三角関数の加法定理も糸口

高校のときに「三角関数の加法定理 [5]」というのを学習した記憶があるのではないでしょうか。

$$\left.\begin{aligned} \sin(\alpha+\beta) &= \sin\alpha\cos\beta + \cos\alpha\sin\beta \\ \cos(\alpha+\beta) &= \cos\alpha\cos\beta - \sin\alpha\sin\beta \end{aligned}\right\} \text{(10)}$$

私が高校生のときに習った数学の先生は、これを「サイン・コス・コス・サイン、コスコス・サイン・サインと覚えなさい」とおっしゃっていました。

これがなんと！このまま CORDIC の繰り返し演算の 1 回分の操作、またN回の繰り返し演算（つまり$i+1=N$）で得られる計算結果に相当するのです！

なおこの「三角関数の加法定理」については（一般的な高校数学であることから）本技術ノートではとくに説明や詳細な解説は行いません。証明なども含めて、ネット上に多数の解説がありますので、そちらをご覧いただければと思います。[5]にも解説が載っています。

N回繰り返し演算で得られる計算結果と加法定理との関係性を考える

ここではまず CORDIC でのN回の繰り返し演算（つまり$i+1=N$）で得られる計算結果$x(N), y(N)$について、加法定理との関係を説明し、次回の技術ノートで CORDIC の演算ステップ 1 回分について加法定理との関係を説明していきます。

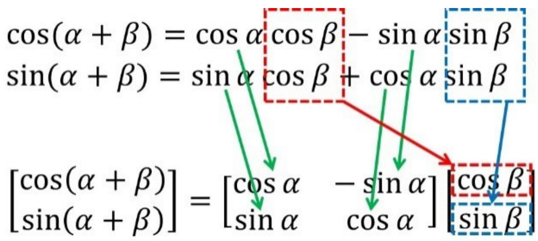


図 13. 加法定理を行列的に表記しなおしてみる

加法定理を行列計算に変形する

さきの式(10)の sin と cos（上下の式）を逆転してみます。

$$\left.\begin{aligned} \cos(\alpha+\beta) &= \cos\alpha\cos\beta - \sin\alpha\sin\beta \\ \sin(\alpha+\beta) &= \sin\alpha\cos\beta + \cos\alpha\sin\beta \end{aligned}\right\} \text{(10) 再掲}$$

これを行列的に表記しなおしてみると（図 13 がその説明）

$$\begin{bmatrix}\cos(\alpha+\beta)\\ \sin(\alpha+\beta)\end{bmatrix} = \begin{bmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{bmatrix}\begin{bmatrix}\cos\beta\\ \sin\beta\end{bmatrix} \tag{11}$$

これを「回転行列」というようですね。なおここでの行列計算とは

$$\begin{bmatrix}AX+BY\\ CX+DY\end{bmatrix} = \begin{bmatrix}A & B\\ C & D\end{bmatrix}\begin{bmatrix}X\\ Y\end{bmatrix} \tag{12}$$

に相当し、図 13 もこの式と同じ変換になります。

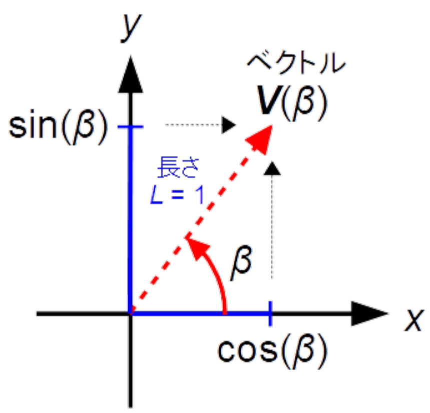


図 14. ふたつの要素を長さが 1 のベクトルの X 成分、Y 成分と考える

加法定理を行列に変形したものをベクトル演算と考える

ここでベクトル

$$\boldsymbol{V}(\beta)=\begin{bmatrix}\cos\beta\\ \sin\beta\end{bmatrix} \tag{13}$$

として、これまでの式のβに関連する部分を、「長さ（ノルム）が 1 で、角度βのベクトル$\boldsymbol{V}(\beta)$」の X 成分（$\cos\beta$）と Y 成分（$\sin\beta$）として、図 14 のように考えます。

また

$$\boldsymbol{R}(\alpha)=\begin{bmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{bmatrix} \tag{14}$$

として回転行列を、ひとつの記号$\boldsymbol{R}(\alpha)$で表します。ここでは回転行列の回転量パラメータを回転角度αとします。

回転行列でベクトルの回転ができる

この行列的表記により、図 15 のように

$$\boldsymbol{V}(\beta+\alpha)\leftarrow\boldsymbol{R}(\alpha)\boldsymbol{V}(\beta) \tag{15}$$

と表現すれば、ベクトル$\boldsymbol{V}(\beta)$（青の破線矢）をベクトル$\boldsymbol{V}(\beta+\alpha)$（赤の破線矢）に変換する、ベクトルの角度を回転できる演算になっていることが分かります。

要素ごとに分解して表記しても、回転行列で要素ごとの角度変化ができている

同じことを繰り返しやってみるわけですが…。またこれをもとに戻して、X 成分（$\cos\beta$）と Y 成分（$\sin\beta$）として要素ごとに分解してみます（数学的にはこれも「ベクトル」と表現されますが…）。

$$\begin{bmatrix}\cos(\beta+\alpha)\\ \sin(\beta+\alpha)\end{bmatrix}\leftarrow\boldsymbol{R}(\alpha)\begin{bmatrix}\cos\beta\\ \sin\beta\end{bmatrix} \tag{16}$$

これは長さ（ノルム）が 1 で角度βのベクトルの、X 成分である$\cos\beta$と、Y 成分である$\sin\beta$を、回転行列$\boldsymbol{R}(\alpha)$により、変化角度αで、「角度$\beta+\alpha$の cos と sin」までその角度を変化できるということです。

この説明をすっ飛ばして回転行列だけとか、複素数とかで CORDIC を解説している記事も多い（読者に理解してもらいたい意思があるのか？？）

英語の解説ページが多いのですが、上記にした説明をすっ飛ばして、いきなり回転行列

$$\boldsymbol{R}(\alpha)=\begin{bmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{bmatrix} \tag{14 再掲}$$

の式のみを示していて、「CORDIC はこうですよー」とか「CORDIC はこうなりますよー」とか書いてあったり、複素数極座標表記

$$Ae^{j\theta}$$

で CORDIC の回転が表現できますとか、式の関係性の導出なしに書いてあったりする解説があります。

結局、これでは何がなんだか、わけがわかりません（涙）。「読者に理解してもらいたい意思があるのか？」とか思います…。それとも本質が分かってないのかな…、とかも思います…。

ともあれこのように cos と sin（ベクトル）の角度を、高校で習った加法定理での「回転行列」を用いることで変化させることができるのです。これで CORDIC の演算を説明できるのです。

回転行列が CORDIC 演算そのもの

この「回転行列」が CORDIC の演算そのものなのです。CORDIC で得られる計算結果の一般式は、式(7)のように

$$\left.\begin{aligned}x(N)&=A_N[x(0)\cos z(0)-y(0)\sin z(0)]\\ y(N)&=A_N[y(0)\cos z(0)+x(0)\sin z(0)]\end{aligned}\right\}\ \text{(7) 再掲}$$

A_Nは図 7 に示したように、CORDIC の漸化式を逐次的にステップごとに計算していくうえで生じる、ベクトルの長さが増加する増加係数の繰り返し積で、Nが十分大きければ 1.64676…に収束します。

CORDIC 漸化式（一般式）を行列に変形する

この$y(N)$の式において、項の順序を変更し、

$$x(N)=A_N[x(0)\cos z(0)-y(0)\sin z(0)]$$
$$y(N)=A_N[x(0)\sin z(0)+y(0)\cos z(0)]$$

つづいて各項の表記順序を逆にして

$$x(N)=A_N[\cos z(0)\,x(0)-\sin z(0)\,y(0)]$$
$$y(N)=A_N[\sin z(0)\,x(0)+\cos z(0)\,y(0)]$$

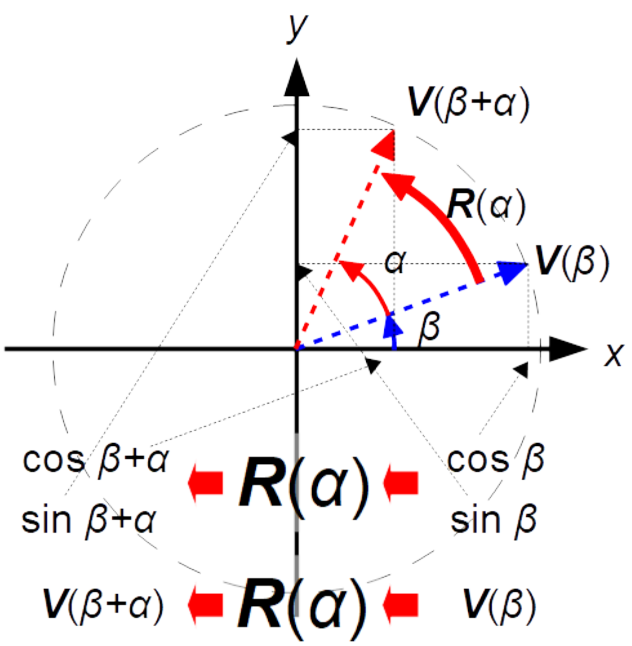


図 15. 加法定理の行列部分を「回転行列」としてみれば、角度βからの回転変化量αとなる角度$\beta+\alpha$への回転変換になる

そして行列的に

$$\begin{bmatrix} AX+BY \\ CX+DY \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \quad \text{(12) 再掲}$$

を使って表してみます。そうすると

$$\begin{bmatrix} x(N) \\ y(N) \end{bmatrix} = A_N \begin{bmatrix} \cos z(0) & -\sin z(0) \\ \sin z(0) & \cos z(0) \end{bmatrix} \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} \quad (17)$$

これは式(16)と全く同じかたちになり、

$$\boldsymbol{R}[z(0)] = \begin{bmatrix} \cos z(0) & -\sin z(0) \\ \sin z(0) & \cos z(0) \end{bmatrix} \quad (18)$$

とすれば、図 16 のようにベクトルの成分$x(0), y(0)$が

$$\begin{bmatrix} x(N) \\ y(N) \end{bmatrix} \leftarrow A_N \boldsymbol{R}[z(0)] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} \quad (19)$$

として、$\boldsymbol{R}[z(0)]$で図 15 と全く同じように、$x(N), y(N)$に回転変換されることが分かります。「ベクトル$\boldsymbol{V}$」の表記で、

$$\boldsymbol{V}(i) = \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (20)$$

とすれば、

$$\boldsymbol{V}(N) \leftarrow A_N \boldsymbol{R}[z(0)] \boldsymbol{V}(0) \quad (21)$$

とも表記できます。ここでも$\boldsymbol{R}[z(0)]$で、$\boldsymbol{V}(0)$ が$\boldsymbol{V}(N)$に回転変換されることが分かります。

CORDIC 漸化式行列（一般式）に初期値を設定する

さらに式(1, 2)で示したように、CORDIC で sin/cos を生成する目的であれば初期値と答えは、

$$\left.\begin{aligned} x(0) &= \frac{1}{A_N} \\ y(0) &= 0 \\ z(0) &= \theta \\ x(N) &= \cos z(0) = \cos\theta \\ y(N) &= \sin z(0) = \sin\theta \end{aligned}\right\} \quad \text{(1, 2) 再掲}$$

なので、式(17)はもっと視覚的に分かりやすい形で

$$\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = A_N \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \frac{1}{A_N} \\ 0 \end{bmatrix} \quad (22)$$

と表されます。

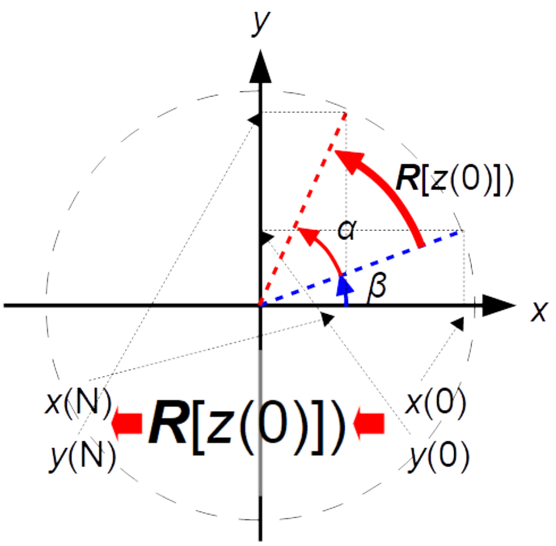


図 16. CORDIC の演算に回転行列を適用してみる

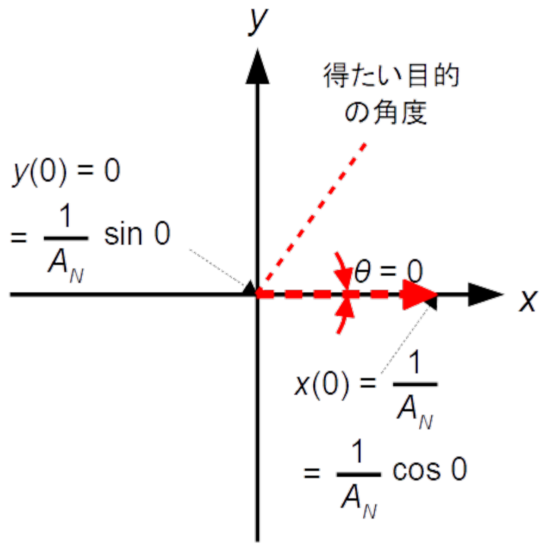


図 17. $x(0)$と$y(0)$を角度 0° ベクトルの X = cos 成分、Y = sin 成分と理解しておく

初期値を角度ゼロのベクトルと考える

さらに図 17 のように、初期値を角度 0° のベクトルだとして

$$\left.\begin{aligned} x(0) &= \tfrac{1}{A_N} = \tfrac{1}{A_N}\cos 0 \\ y(0) &= 0 = \frac{1}{A_N}\sin 0 \end{aligned}\right\} \quad (23)$$

として$x(0)$と$y(0)$を設定すれば、式(17)や式(20)は

$$\left.\begin{aligned} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} &= \frac{A_N}{A_N} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos 0 \\ \sin 0 \end{bmatrix} \end{aligned}\right\} \quad (24)$$

となります。

角度ゼロから回転行列で角度θのベクトルの成分（CORDIC の答え）が得られる

この CORDIC の式は、0° から角度θへの変換を、回転行列

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (25)$$

をもちいて

$$\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \leftarrow \boldsymbol{R}(\theta) \begin{bmatrix} \cos 0 \\ \sin 0 \end{bmatrix} \quad (26)$$

として、回転角度量をθにして計算する演算とまったく同じということが分かります。

まとめ

書き始めた段階では、まだトンネルの出口には到達していない、手探り状態というスタートでしたが、探求していくことで満足できるところまで理解を深めることができました。

CORDIC とは初期値$z(0)$を目的の角度、$z(i)$を残留角度誤差量として、加法定理を基礎とした計算アルゴリズムで、ベクトルの sin/cos 値を求めていくものだと分かりました。またそれは SAR ADC の動作とかなり近いしくみだとご説明しました。

次回はこの理解から、CORDIC の 1 ステップ演算がどのように構成されるかなどをより深く考えていき、実際に CORDIC アルゴリズムを FPGA に DDS（Direct Digital Synthesizer）として実装

し、DAC IC を使って正弦波を出力してみた例をご紹介したいと思います。

しかし「よく出来てるなぁー」としきりに感心する、CORDIC アルゴリズムでありました！

参考文献

[1] ショット雑音; Wikipedia, https://ja.wikipedia.org/wiki/ショット雑音

[2] Ray Andraka; A survey of CORDIC algorithms for FPGA based computers, http://www.fpga-guru.com/files/crdcsrvy.pdf

[3] 総乗; Wikipedia, https://ja.wikipedia.org/wiki/総乗

[4] The Data Conversion Handbook 2005, Analog Devices, https://www.analog.com/en/education/education-library/data-conversion-handbook.html

[5] 三角関数の加法定理, https://ja.wikipedia.org/wiki/三角関数#加法定理