

# 極座標ステージとエクストルーダの同期

青木 翔平

June 22, 2016

## 1 ステージ側の線速度について

極座標におけるステージ側の線速度を一定にできるかという問題について、理論的な検討を行う。

デカルト座標系での線速度  $v$  は極座標で以下のよう  
に表される。

$$\begin{aligned} |v| &= \sqrt{\dot{x}^2 + \dot{y}^2} \\ &= \sqrt{\left(\frac{d}{dt}r \cos \theta\right)^2 + \left(\frac{d}{dt}r \sin \theta\right)^2} \\ &= \sqrt{\dot{r}^2 + r^2\dot{\theta}^2} \end{aligned} \quad (1)$$

いま、角速度と半径方向の速度を制御して線速度一定にすることを考える。 $v = k$  (線速度一定) とすると、

$$\dot{r}^2 + r^2\dot{\theta}^2 = k^2 \quad (2)$$

$$\dot{r}^2 = k^2 - r^2\dot{\theta}^2 \quad (3)$$

微小区間で考えると

$$\begin{aligned} \left(\frac{\Delta r}{\Delta t}\right)^2 &= k^2 - r^2 \left(\frac{\Delta \theta}{\Delta t}\right)^2 \\ |\Delta r| &= \sqrt{(k\Delta t)^2 - (r\Delta \theta)^2} \end{aligned} \quad (4)$$

いっぽう、NC コードにおいて直線移動命令 G1 が与えられたとき、CNC コントローラではブレゼンハムのアルゴリズム (Algorithm 1) にしたがって補間位置  $\{x_1, x_2, \dots, x_n\}, \{y_1, y_2, \dots, y_n\}$  が計算される。また、計算された  $\{x_1, x_2, \dots, x_n\}, \{y_1, y_2, \dots, y_n\}$  は以下の関係式を用いて  $\{r_1, r_2, \dots, r_n\}, \{\theta_1, \theta_2, \dots, \theta_n\}$  へと座標変換される。

$$\begin{cases} r_i = \sqrt{x_i^2 + y_i^2} \\ \theta_i = \tan^{-1}(y_i/x_i) \end{cases}$$

すなわち、差分形式とすれば

$$\begin{cases} \Delta r = r_2 - r_1 = \sqrt{x_2^2 + y_2^2} - \sqrt{x_1^2 + y_1^2} \\ \Delta \theta = \theta_2 - \theta_1 = \tan^{-1}(y_2/x_2) - \tan^{-1}(y_1/x_1) \end{cases} \quad (5)$$

一般に、(5) で指定される  $\Delta r, \Delta \theta$  が (4) 式を満たすとは限らないため<sup>1</sup>、任意の目的地の座標が指定されたときに線速度を一定にすることはできない。

---

### Algorithm 1: Interpolation of Bresenham

---

```
dx ← x1 - x0
dy ← y1 - y0
D ← 2 * dy - dx
plot(x0, y0)
y ← y0
for x from x0 + 1 to x1 do
  if D > 0 then
    y ← y + 1
    plot(x, y)
    D ← D + (2 * dy - 2 * dx)
  else
    plot(x, y)
    D ← D + (2 * dy)
  end if
end for
```

---

## 2 エクストルーダの制御について

ステージの変位と射出量の関係について考える。

デカルト座標系において、移動時間  $\Delta t$  と線速度  $F$  [mm/min] の関係は、以下のように表される。

$$\Delta t = \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{F} \cdot 60 \cdot 1000 \quad (6)$$

エクストルーダ (スクリー) の射出量を  $\Delta E_{screw}$  , スクリューの回転数を  $SNW$  [rpm] とおけば、以下の式

<sup>1</sup> $r$  の自由度が残る。

が成り立つ．

$$\begin{aligned}\Delta E_{screw} &= \beta \cdot (SNW) \cdot \Delta t \\ \therefore SNW &= \frac{1}{\beta} \frac{\Delta E_{screw}}{\Delta t} = \beta' \frac{F \cdot \Delta E_{screw}}{\sqrt{(\Delta x)^2 + (\Delta y)^2}}\end{aligned}\quad (7)$$

前回，スライサーからの  $\Delta E = E_2 - E_1$  を  $\Delta E_{screw}$  と等しいとみなして計算を行ったが，これは誤りである．なぜならば，スライサーが出力した  $E$  には極座標の位置による速度変化が考慮されていないからである．したがって，スライサーからのエクストルーダの値は無

視する．（実際の実装ではエクストルーダの値は射出の On/Off 判定のために使用している）

極座標におけるステージの線速度を  $F_{polar}(t)$  とすれば，微分を後退差分表示することで以下の式を得る．

$$\begin{aligned}F_{polar}(t_k) &= \left. \frac{dS}{dt} \right|_{t=t_k} \\ &= \sqrt{\dot{r}^2 + r^2 \dot{\theta}^2} \Big|_{t=t_k} \\ &= \sqrt{\left( \frac{r_k - r_{k-1}}{\Delta t} \right)^2 + \left( \frac{r_k + r_{k-1}}{2} \right)^2 \left( \frac{\theta_k - \theta_{k-1}}{\Delta t} \right)^2}\end{aligned}$$

ステージの変位とスクリーの変位は比例するので

$$\begin{aligned}\int_{t_1}^{t_2} SNW(t) dt &= \alpha \int_{t_1}^{t_2} F_{polar}(t) dt \\ \therefore SNW \Big|_{t=t_k} &= \alpha F_{polar}(t_k) \\ &= \frac{\alpha}{\Delta t} \sqrt{(r_k - r_{k-1})^2 + \left( \frac{r_k + r_{k-1}}{2} \right)^2 (\theta_k - \theta_{k-1})^2} \\ &= \frac{\alpha \cdot 60 \cdot 1000 \cdot \sqrt{(\Delta x)^2 + (\Delta y)^2}}{F} \cdot \sqrt{(r_k - r_{k-1})^2 + \left( \frac{r_k + r_{k-1}}{2} \right)^2 (\theta_k - \theta_{k-1})^2} \\ &= \frac{\alpha'}{F} \sqrt{(r_k - r_{k-1})^2 + \left( \frac{r_k + r_{k-1}}{2} \right)^2 (\theta_k - \theta_{k-1})^2} \\ &\quad \cdot \sqrt{(r_k \cos \theta_k - r_{k-1} \cos \theta_{k-1})^2 + (r_k \sin \theta_k - r_{k-1} \sin \theta_{k-1})^2}\end{aligned}\quad (8)$$

(8) 式にしたがってスクリー回転数を制御することで，線速度が変化する状況においてもスクリーの射出量を線速度に追従させることができる．

### 3 予想される問題点

#### 3.1 半径の近似精度

(8) 式に表される  $r_k$  の計算において， $r_k$  と  $r_{k-1}$  の平均値を用いているが， $r$  が急激に増減する場合にこの部分の誤差の影響が大きくなる．一部の特殊な形状 例えば鋭利なヒトデ型 を造形する際に問題が懸念されるが，これは与えられる G-code の半径方向の刻み ( $= \Delta r$ ) が小さい場合，実際は問題にならないと考えられる．

### 4 試験結果をうけた制御式の変更

(8) 式にしたがって実際にプリント試験を行ったところ，角速度の変動が速すぎて射出量が追従できない，あるいは角速度が位置によってうまく変動しないなどの不具合があった．微分の誤差が効いているものと思われる．

制御式の変更について，以下の2つのアプローチを考える．

1. (8) 式の高周波成分を抑制する
2. モデルを変更する

以下でそれぞれについて検討する．

#### 4.1 高周波成分の抑制

プリント試験に用いたモデルの G-code を基に制御式のシミュレーションを行った結果を以下に示す．



Figure 1: Numerical computation of screw speed

(8) 式による制御結果 Figure 1 を見ると、下段  $v_2$  で示されるスクリー速度が振動していることが読み取れる。これを適当なローパスフィルターで除去することを考える。

## 4.2 デジタルフィルタ

### 4.2.1 移動平均フィルタ

$z$  変換

## 4.3 非線形カルマンフィルタ

### 4.3.1 拡張カルマンフィルタ (EKF)

(8) 式を離散時間非線形状態空間表現で表す。

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k)) + \mathbf{b}v(k) \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k)) + \mathbf{w}(k) \end{aligned}$$

$\mathbf{f}(\mathbf{x})$ ,  $\mathbf{h}(\mathbf{x})$  のヤコビ行列をそれぞれ  $\mathbf{A}(k)$ ,  $\mathbf{c}^T(k)$  とおけば、

$$\begin{aligned} \mathbf{A}(k) &= \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \\ \mathbf{c}^T(k) &= \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^-(k)} \end{aligned}$$

このとき、(9) 式は

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{b}v(k) + \mathbf{u}(k) \\ \mathbf{z}(k) &= \mathbf{c}^T(k)\mathbf{x}(k) \end{aligned}$$

拡張カルマンフィルタ (EKF) のアルゴリズムは以下のように表される。

### 拡張カルマンフィルタ

#### □ 時間更新式

$k = 1, 2, \dots, N$  に対して以下を実行

#### • 予測ステップ

事前状態推定値:

$$\hat{\mathbf{x}}^-(k) = \mathbf{A}\hat{\mathbf{x}}^-(k-1)$$

事前誤差共分散行列:

$$\mathbf{P}^-(k) = \mathbf{A}\mathbf{P}(k-1)\mathbf{A}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T$$

#### • フィルタリングステップ

カルマンゲイン行列:

$$\mathbf{G}(k) = \mathbf{P}^-(k)\mathbf{C}^T(\mathbf{C}\mathbf{P}^-(k)\mathbf{C}^T + \mathbf{R})^{-1}$$

状態推定値:

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{G}(k)(\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}^-(k))$$

事後誤差共分散行列:

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{G}(k)\mathbf{C}\mathbf{P}^{-1})$$

ただし共分散行列  $\mathbf{P}^{-1}(k)$ ,  $\mathbf{P}(k)$  は  $n \times n$  行列, カルマンゲイン  $\mathbf{G}(k)$  は  $n \times p$  行列である。

### 4.3.2 アンセンテッドカルマンフィルタ (UKF)

非線形関数  $\mathbf{f} : \mathcal{R}^n \rightarrow \mathcal{R}^n$  が  $n$  次元確率変数  $\mathbf{x}$  を  $n$  次元確率変数  $\mathbf{y}$  に変換するとき、 $\mathbf{x}$  を 1 次モーメント統計量と 2 次モーメント統計量で表現するためのシグマポイントは、スケーリングパラメータを  $\kappa$  とすると以下のように表される。

$$\begin{aligned} \mathcal{X}_0 &= \bar{\mathbf{x}} \\ \mathcal{X}_i &= \bar{\mathbf{x}} + \sqrt{n + \kappa}(\sqrt{\mathbf{P}_x})_i \\ \mathcal{X}_{n+1} &= \bar{\mathbf{x}} - \sqrt{n + \kappa}(\sqrt{\mathbf{P}_x})_i \end{aligned}$$

$\mathbf{P}_x$  は共分散行列の定義から正定値対称行列であり、したがって  $\sqrt{\mathbf{P}_x}$  はコレスキー分解によって求まる。

上のシグマポイントを用いて確率密度関数を  $\mathbf{U}$  変換する。具体的には、

$$\begin{aligned} \mathcal{Y}_i &= \mathbf{f}(\mathcal{X}_i) \\ \bar{\mathbf{y}} &= \sum_{i=0}^{2n} w_i \mathcal{Y}_i \\ \mathbf{P}_y &= \sum_{i=0}^{2n} w_i (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T \end{aligned}$$

アンセンテッドカルマンフィルタ (UKF) のアルゴリズムは以下のように表される。

## UKF

## □ 初期値

## □ 時間更新式

$k = 1, 2, \dots, N$  に対して以下を実行

## ● シグマポイントの計算

## ● 予測ステップ

シグマポイントの更新:

事前状態推定値:

$$\hat{x}^-(k) = A\hat{x}^-(k-1)$$

事前誤差共分散行列:

$$P^-(k) = AP^-(k-1)A^T + BQB^T$$

シグマポイントの再計算:

出力のシグマポイントの更新:

事前出力推定値:

事前出力誤差共分散行列:

事前状態・出力誤差共分散行列:

カルマンゲイン:

$$G(k) = P^-(k)C^T(CP^-(k)C^T + R)^{-1}$$

## ● フィルタリングステップ

状態推定値:

$$\hat{x}(k) = \hat{x}^-(k) + G(k)(y(k) - C\hat{x}^-(k))$$

事後誤差共分散行列:

$$P(k) = (I - G(k)CP^{-1})$$

めるというアドホックな方法であるが、直感的にもわかりやすく、実装も単純である。

(9) 式を用いて実験を行ったところ改善が見られたので、以降はこの方法を採用する。なぜこの方法がすぐれているのか、その原因については後述する。

## 5 議論

現時点で判明している問題点として、

1. そもそも吐出側を一定にしたい
2. 原点での切り返し問題
3. エクストルーダの On/Off の遅延

がある。以下でそれぞれについて考察する。

## 5.1 吐出側を一定にする

現在の計画では G-code に応じて吐出側、すなわちスクリーンの角速度を追従させることを目的としているが、これには以下の欠点がある。

- 回転数値が指令されてから実際にスクリーンの回転数に変更されるまでに遅延がある
- スクリーンの回転数に変更されたあと、射出量が追従するまでに遅延がある
- 吐出量の変動に応じて冷却を調節する必要があるが、冷却はリニアに効くわけではない

したがって、ステージの線速度を制御したほうがよいという結論になるが、前章で述べたように極座標において線速度を一定にすることは理論的に不可能である。

では、逆にエクストルーダが一定速度で回転しているとして、これにステージ側の線速度を追従させることは可能であろうか。極座標系での速度は  $r$  と  $\theta$  を用いて (2) 式で与えられるが、 $v$  を与えた場合に  $\Delta r$  と  $\Delta \theta$  は一意に決まるわけではない。加えて前章での経験では、平方根および乗算演算に起因する数値誤差が懸念されるので、直接的に  $r$  と  $\theta$  を決定することは避けたい。

ここで発想を変えて、射出量に応じた線速度から直接  $r$  と  $\theta$  の制御量を導くのではなく、なるべくこれに近い線速度でステージが動くように本体を制御することを考える。具体的には、以下のように問題を単純化する。

仮定 1: 半径方向の速度変化は比較的小さく、回転方向の運動が主である

## 4.4 パーティクルフィルタ

## 4.5 モデルの変更

Figure 1 の計算結果で示されるような高周波成分を含む速度指令をスクリーンに与えても、スクリーン内部での圧力変化にもとづく樹脂の流速の応答が遅いために樹脂の出力は追従できない。ならばいっそのことモデルを変更することを考える。

樹脂の追従速度が遅いため、速度指令は時間発展に応じてただちにドリフトするようなモデルが好ましい。また、定性的には動径方向の変位が大きいときに射出速度を速めるような処理が必要となる。

ここではヒューリスティックな方法として、単純に半径に比例するような成分を (7) 式に加えてみる。修正後の関係式を以下に示す。

$$SNW = \gamma \frac{F \cdot r \cdot \Delta E}{\sqrt{(\Delta x)^2 + (\Delta y)^2}} \quad (9)$$

(9) 式は原点より遠い場所ほどスクリーン回転数を高

この仮定をおけば (2) 式は  $v = r\omega$  と簡素化できるから、半径に反比例するようにステージ側の回転テーブルの回転数を調節すれば、エクストルーダのスクリーウ回転数に追従することができる。

仮定 1 の妥当性として、かりに  $r$  の変動量  $\dot{r}$  が大きかったとしてもそれは瞬間的なものであり、スクリーウ回転数上昇後の樹脂吐出量の応答が低速であることを踏まえれば、この変動成分は無視できると可能性が高い。 $r$  の絶対量は仮定後の式でも吸収できるから、問題はないと考えられる。

じつのところ、この制御方法は (9) 式でスクリーウ回転数を制御する方法と双対関係にある。したがって、ここで与えた検討は (9) 式の妥当性を補強している。

## 5.2 原点での切り返し問題

現在のファームウェアの実装では原点で進行方向を切返すために、原点近傍での出力に継ぎ目が見られる。これを避けるためにはファームウェアのロジックを変更する必要がある。

原点での切り返しが問題になるかどうかは造形物の形状に依存する。例えばコルセットをステージ中央に配置したときのプリントでは問題になっていない。まっすぐな円筒なども問題は発生しないが、折れ曲がった指先が原点にかかるような配置を行うと原点での切り返しが発生する。

予想では、いまの実装ではアークタンジェントの計算関数によって  $\Delta\theta$  の解の存在範囲が  $-2\pi \leq \theta \leq 2\pi$  に限定されているため、 $\theta$  が巻き戻されている可能性がある。したがって、この計算部分で連続的に  $\theta$  を推移させれば問題は解決する見込みである。

## 5.3 エクストルーダの On/Off の遅延

スクリーウ回転数にゼロを指定するとピンが作動して射出が止められるが、このピンが再度開いた時に樹脂が流れ出るまでに遅延が発生する。これを解決する一つの方法は G-code 指令を先読みすることでこの遅延分を見込んだ開閉を行うことである。

これの実現方法には (1)G-code を先読みする方法と、(2)エクストルーダのピン開閉時にステージ側のモーターを停めてしまう方法の 2 通りが考えられる。組み込みの実装としては後者のほうが簡易であったので、これを採用した。

## 6 Jerk の制御について

### 6.1 極座標における Jerk

前章で述べたように、極座標系において一般に線速度は一定にすることができない。 $\dot{r}$  と  $\dot{\theta}$  の振る舞いはファームウェア内で加速度を設定することで操作できるが、この際の瞬発的な加速度の増加 (加加速度  $\dot{a}$ ) を Jerk と呼んでいる。

いま、極座標で加速度制御を行ったときに線速度のプロファイルがどのような様子を描くかに興味がある。極座標系での線速度は以下の式 (10) から計算できる。半径方向の速度  $\dot{r}$  と角速度  $\dot{\theta}$  を適当な台形プロファイルで仮定して数値計算した結果を Figure 2 に示す。図中の右で示される  $v$  が極座標系における線速度である。

$$|v| = \sqrt{\dot{r}^2 + r^2\dot{\theta}^2} \quad (10)$$

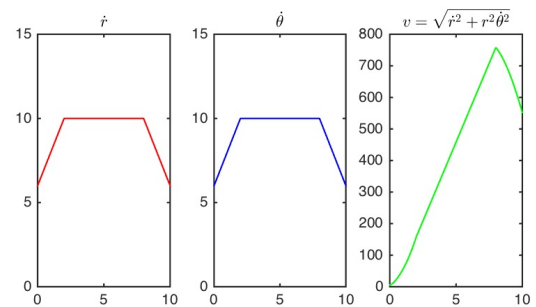


Figure 2: Velocity profile

Figure 2 から、 $\dot{r}$  と  $\dot{\theta}$  を台形型のプロファイルで制御したとしても、線速度は台形型とならないことがわかる。これはデカルト座標系との違いである。

### 6.2 Jerk の実装

現在使用している R360 用のファームウェア<sup>2</sup>においては、前章で述べたような台形型の加減速プロファイルが実装されており、Jerk の値は初期速度 (回転数) の固定値として以下のように与える。

```
#define DEFAULT_XYJERK 20.0 // (mm/sec)
#define DEFAULT_ZJERK 0.4 // (mm/sec)
#define DEFAULT_EJERK 5.0 // (mm/sec)
```

### 6.3 ステージ側 Jerk 計算の修正

現在の危機構成ではサーボアンプに対する指令をパルスで与えるのではなく、RS-232C シリアル通信上の ASCII

<sup>2</sup><https://github.com/kory75/Marlin.360>

文字列で指定しているため、エクストルーダのスクリュー回転数の加減速を細やかに制御するのは難しい。また、そもそもエクストルーダからの樹脂の射出の応答は遅い。

では、いっそのことステージ側の加減速計算を台形型ではなく、エクストルーダの応答に沿う形にしてはどうかという案が考えられる。しかし、そもそも Jerk の制御が最終出力物の精度にどの程度影響をあたえるかという疑問が浮かぶ。

いま、問題は以下の 2 つに分解される。

1. スライサーからのエクストルーダの値を使った場合、ステージ側の加速度はどのように制御すればよいのか
2. ステージ側の Jerk を制御できたとして、それが出力物の精度に与える影響はいかほどか

以下で、上記の 2 点について考察する。

### 6.3.1 ステージ側の加速度制御

スライサーが吐き出した G-code に記述されたエクストルーダの値から、スクリューの速度プロファイルを逆算する。いま、(10) 式に従って  $v$  が推移するとき、 $\dot{r}$  と  $\dot{\theta}$  は一意に決定できない。ここで独立変数を減らすために、 $\theta$  は今までと同じように線形加速制御を行うとする。いま、この  $\theta$  の計算に乗算器と加算器のみでステッピングモーターのパルス遅延時間を高速に計算するアルゴリズムである *leib ramp*[7] をすると仮定する<sup>3</sup>。

$\theta$  の計算方法について詳しく述べる。総加速ステップを  $S$ 、ステッピングモーターの初期速度を  $v_0$ 、目標速度を  $v$ 、1 秒ごとの加速ステップ数を  $a$ 、加速時間を  $t$  とすれば、

$$\begin{aligned} S &= v_0 t + \frac{1}{2} a t^2 \\ v &= v_0 + a t \end{aligned} \quad (11)$$

(11) における両式より、以下を得る。

$$\begin{aligned} S &= \frac{v^2 - v_0^2}{2a} \\ v &= \sqrt{v_0^2 + 2aS} \end{aligned} \quad (12)$$

(12) における両式を再帰的に適用して漸化式を導けば、

$$v_i = \sqrt{v_{i-1}^2 + 2a} \quad (13)$$

<sup>3</sup>Marlin でも実装されている。

$i$  ステップにおけるステッピングモーターの遅延時間を  $p_i$ 、プロセッサのタイマー割り込み周波数を  $F$  とするとき、

$$p_i = \frac{F}{v_i}$$

であるから、(13) 式を用いることで、 $p$  に関する漸化式は以下のように変形できる。

$$p_i = \frac{F}{\sqrt{\left(\frac{F}{p_{i-1}}\right)^2 + 2a}} \quad (14)$$

目標速度における遅延時間を  $p_s$ 、定数  $R = a/F^2$  とおき、(14) 式をテイラー展開して 1 次の項で打ち切れば、

$$p_{i+1} = p_i \cdot (1 + m \cdot p_i \cdot p_i) \quad (15)$$

ただし、 $m$  は以下で定義する。

$$\begin{cases} m = -R: & \text{during acceleration} \\ m = 0: & \text{slew speed} \\ m = R: & \text{during deceleration} \end{cases} \quad (16)$$

(16) 式をアルゴリズムとして整理したものを Algorithm 2 に示す。

---

#### Algorithm 2: Leib ramp acceleration

---

```

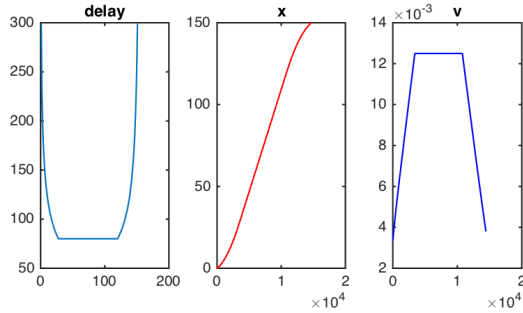
 $v_0, v, a, F$  given.
 $S \leftarrow (v^2 - v_0^2)/2a$ 
 $p_1 \leftarrow F/(v_0^2 + 2a)^{-\frac{1}{2}}$ 
 $p_s \leftarrow F/v$ 
 $R \leftarrow a/F^2$ 
 $T_s$  given.
for  $i$  from 1 to  $T_s$  do
  if  $i \leq S$  then
     $m \leftarrow -R$ 
  else if  $i \leq T_s - S$  then
     $m \leftarrow 0$ 
  else
     $m \leftarrow R$ 
  end if
   $p_{i+1} \leftarrow p_i \cdot (1 + m \cdot p \cdot p)$ 
end for

```

---

これをもとに数値計算を行った例を Figure 3 に示す。いま、ある時刻における  $\theta_i$  は *leib ramp* で求めた。次に、 $\dot{r}$  は (2) 式を用いて

$$\left| \frac{dr}{dt} \right| = \sqrt{\frac{\Delta E}{\Delta t} - r^2 \cdot \dot{\theta}} \quad (17)$$

Figure 3: Compute  $\dot{\theta}$  with leib ramp**Algorithm 3:**  $\dot{r}$  solver

---


$$\Delta E \leftarrow E_2 - E_1$$

$$\Delta S \leftarrow ((x_2 - x_1)^2 + (y_2 - y_1)^2)^{\frac{1}{2}}$$

$$\Delta \theta \leftarrow \tan^{-1}(y_2/x_2) - \tan^{-1}(y_1/x_1)$$

$$T_s = \alpha \cdot \Delta \theta$$

compute  $p_i$  with Leib Ramp.

$$\dot{\theta} \leftarrow 1/p_i$$

$$dr/dt \leftarrow (\Delta E/\Delta t - r^2 \cdot \dot{\theta})^{\frac{1}{2}}$$


---

と表されるから,  $\dot{r}$  を求めるアルゴリズムをまとめると, Algorithm 3 のようになる.

数値計算の結果を Figure 4 に示す.

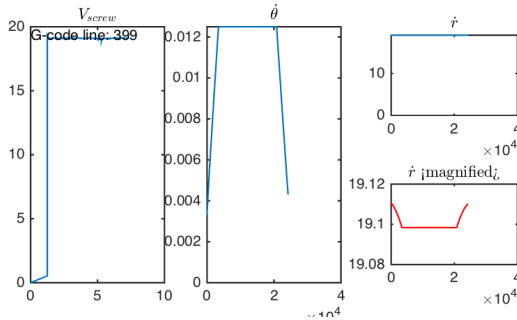
Figure 4: Computed  $\dot{r}$ **6.4 出力物の精度に与える影響の見積もり**

Figure 4 によれば,  $\dot{\theta}$  を線形制御した時の  $\dot{r}$  の変動は非常に小さいものである. ( $\sim 1.0 \times 10^{-4}$ )

- TODO:  $\dot{r}$  の加速部分の加速時間と, 変動量から理想制御における x-y 変位量のオーダーを計算し, それが小さいことを述べる

**7 樹脂射出応答の遅延について**

制御ボード上のプロセッサが PLC に回転数指令を与えてから, じっさいにエクストルーダから樹脂が射出されるまでには遅延がある. これについて定量的な考察を行う.

遅延が発生する箇所を再掲すると,

1. 回転数値が指令されてから実際にスクリーウの回転数に変更されるまでの遅延
2. スクリューの回転数に変更されたあと, 射出量が追従するまでの遅延

まず, 1. については回転数は実時間で追従していることが確認できている. したがって問題は 2. の部分である.

以下ではスクリーウ内部の圧力変動による速度変化をナビエ=ストークス方程式で計算する. 本項では速度-圧力法, 特にフラクショナル・ステップ法 [9, 10] を用いて計算する.

いま, 連続の式は

$$\nabla \cdot \mathbf{v} = 0 \quad (18)$$

ナビエ=ストークス方程式は,

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{v} \quad (19)$$

(19) 式の時間微分に対して前進差分を適用すると,

$$\frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} + (\mathbf{v}^n \cdot \nabla) \mathbf{v}^n = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{v}^n \quad (20)$$

(20) 式を次の 2 式に分解して段階的に計算することを考える.

$$\frac{\mathbf{v}^* - \mathbf{v}^n}{\Delta t} + (\mathbf{v}^n \cdot \nabla) \mathbf{v}^n = \frac{1}{Re} \nabla^2 \mathbf{v}^n \quad (21)$$

$$\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} = -\nabla p \quad (22)$$

ここで (21) 式は速度輸送方程式である.

(21) 式によって求めた  $\mathbf{v}^*$  と (22) 式を用いれば, 次の時間における  $\mathbf{v}^{n+1}$  が

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \Delta t \nabla p \quad (24)$$

として求まるが, まだ圧力が決定されていない. (24) 式の両辺のダイバージェンスをとり, 連続の式 (18) を用いれば, 圧力  $p$  は以下のポアソン方程式を満たすように決定できる.

$$\begin{aligned}
u_{i-1/2,j}^* &= u_{i-1/2,j}^n - \Delta t \left\{ u_{i-1/2,j}^n \frac{u_{i+1/2,j}^n - u_{i-3/2,j}^n}{2\Delta x} + v_{i-1/2,j}^n \frac{u_{i-1/2,j}^n - u_{i-1/2,j-1}^n}{2\Delta y} \right. \\
&\quad \left. + \frac{1}{Re} \left( \frac{u_{i+1/2,j}^n - 2u_{i-1/2,j}^n + u_{i-3/2,j}^n}{(\Delta x)^2} + \frac{u_{i-1/2,j}^n - 2u_{i-1/2,j-1}^n + u_{i-1/2,j-2}^n}{(\Delta y)^2} \right) \right\} \quad (26)
\end{aligned}$$

$$\begin{aligned}
v_{i,j-1/2}^* &= u_{i,j-1/2}^n - \Delta t \left\{ u_{i,j-1/2}^n \frac{v_{i+1,j-1/2}^n - v_{i-1,j-1/2}^n}{2\Delta x} + v_{i,j-1/2}^n \frac{v_{i,j+1/2}^n - v_{i,j-3/2}^n}{2\Delta y} \right. \\
&\quad \left. + \frac{1}{Re} \left( \frac{v_{i+1,j-1/2}^n - 2v_{i,j-1/2}^n + v_{i-1,j-1/2}^n}{(\Delta x)^2} + \frac{v_{i,j+1/2}^n - 2v_{i,j-1/2}^n + v_{i,j-3/2}^n}{(\Delta y)^2} \right) \right\} \quad (27)
\end{aligned}$$

## 2次元移流拡散方程式

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} + v \frac{\partial f}{\partial y} = D \left( \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)$$

## 2次元 CIP 法

$$\xi = -u\Delta t$$

$$\eta = -v\Delta t$$

## 3次元エルミート補間

$$F(\xi, \eta) = \sum_{l=0}^3 \sum_{m=0}^3 C_{l,m} \xi^l \eta^m$$

## A型 CIP

$$\begin{aligned}
F(\xi, \eta) &= \{(C_{3,0}\xi + C_{2,1}\eta + C_{2,0})\xi + C_{1,1}\eta + C_{1,0}\}\xi \\
&\quad + \{C_{0,3}\eta + C_{1,2}\xi + C_{0,2}\}\eta + C_{0,0}
\end{aligned}$$

## 格子点の関数の計算値

$$\begin{cases}
C_{3,0} = \{(g_{ip,j} + g_{i,j})\delta_x - 2(f_{i,j} - f_{ip,j})\}/\delta_x^3 \\
C_{0,3} = \{(h_{i,jp} + h_{i,j})\delta_y - 2(f_{i,j} - f_{i,jp})\}/\delta_y^3 \\
C_{2,0} = \{3(f_{ip,j} - f_{i,j}) + 2(g_{ip,j} + 2g_{i,j})\delta_x\}/\delta_x^2 \\
C_{0,2} = \{3(f_{i,jp} - f_{i,j}) + 2(h_{i,jp} + 2h_{i,j})\delta_y\}/\delta_y^2 \\
a = f_{i,j} - f_{i,jp} - f_{ip,j} + f_{ip,jp} \\
b = h_{ip,j} - h_{i,j} \\
C_{1,2} = (-a - b\delta_y)/(\delta_x\delta_y^2) \\
C_{2,1} = \{-a - (g_{i,jp} - g_{i,j})\delta_x/\delta_x^2\delta_y\} \\
C_{1,1} = (-b + C_{2,1}\delta_x^2)/\delta_x \\
C_{1,0} = g_{i,j} \\
C_{0,1} = h_{i,j} \\
C_{0,0} = f(i, j)
\end{cases}$$

## References

- [1] 水島二郎, 柳瀬眞一郎, 理工学のための数値計算法, 数理工学社 (2009)
- [2] 伊理正夫, 藤野和建, 『数値計算の常識』, 共立出版 (1985)
- [3] 永井清, 土橋宏規, 『ロボット機構学』, コロナ社 (2015)
- [4] 松日楽信人, 大明準治, 『わかりやすいロボットシステム入門』, オーム社 (2010)
- [5] 松原厚, 『精密位置決め・送り系設計のための制御工学』, 森北出版 (2008)
- [6] 足立修一, 丸田一郎, 『カルマンフィルタの基礎』, 東京電機大学出版局 (2012)
- [7] Aryeh Eiderman, "Real Time Stepper Motor Linear Ramping Just by Addition and Multiplication," <http://hwml.com/LeibRamp.pdf>
- [8] 酒井幸市, 『WebGL による流れと波のシミュレーション』, 工学社 (2014)
- [9] 河村哲也, 『流れのシミュレーションの基礎』, インデックス出版 (2007)
- [10] 梶島岳夫, 『乱流の数値シミュレーション』, 養賢堂 (2014)
- [11] 川又政征, MATLAB で学ぶデジタル信号処理の基礎, 信号処理学会誌 (2001)
- [12] 三谷政昭, Scilab で学ぶデジタル信号処理, CQ 出版 (2006)