



リニアな曲率・捩率を持つセグメントによる軌道生成*

蘭 豊礼** 玉井博文** 三浦憲二郎*** 牧野 洋†

Path Generation Using Linear Curvature and Torsion Segments

Fengli LAN, Hirofumi TAMAI, Kenjiro T. MIURA and Hiroshi MAKINO

In two-dimensional(2D) space, the clothoid is a preferred trajectory curve because its curvature varies linearly with its curve length. However, in three-dimensional(3D) space, both curvature and torsion must be considered. This paper deals with path generation using linear curvature and torsion segments which can be considered a 3D extension of the 2D clothoid. In our study, the path segments are generated by solving the Frenet-Serret equation. In every path segment, its curvature and torsion varies linearly with its curve length. In order to obtain more free parameters, plural curve segments are connected in series to make a compound curve. The curve is used to connect two given points which may have given Frenet-Frame, curvature and torsion constraints. These curves are also used to construct a smooth transition passing through an arbitrary point sequence. The resultant path possesses C^2 as well as torsion continuity and matches all given Frenet-frame, curvature and torsion constraints at the given points.

Key words: path generation, clothoid, Frenet-Serret equation, Frenet-Frame, curvature, torsion, C^2 , torsion continuity.

1. 緒 言

工作機械や産業用ロボットなどの分野では工具の軌道設計が必要である。二次元（以下、2D）の場合にクロソイド曲線を補間曲線として用いることが提案されている^{1)①}。クロソイド曲線の曲率は曲線長のリニア関数であり、優位性のある軌道曲線として認められている。ただし、接続端点における接線方向と曲率を同時に指定される場合にパラメータが不足する。筆者らは、このため複数のクロソイドを直列に接続し、特に3連クロソイド^②によってこの問題を解いた。

さらに三次元（以下、3D）空間においても、接線のピッチ角とヨー角がそれぞれクロソイド的に変化する曲線を定義して、これによる補間を行なったが^{③④}、この方法では、リニア曲率を持つ3D曲線を生成することは困難である。

3D曲線補間法としては、Bスプライン曲線やエルミート曲線を用いたもの^{⑤-⑩}があるが、これらの曲線の曲率は曲線長の複雑な関数である。また、3D離散クロソイドを使う方法^⑪もある。この方法では区分リニアな離散曲率が得られるが、各中間通過点での接線方向を指定できない。これらに対して、著者らの1人は単位4元数積分曲線^⑫を用いて3D曲線を取り扱うことを提案している。単位4元数積分曲線の曲率は曲線長の比較的簡単な関数であるが、やはりリニアな曲率を持つ空間曲線を生成することは困難である。また、リニアな曲率を持つ空間曲線については、対数型美的空間曲線の分野でも研究している^⑬。

本報においては、これらの解析を参考にして、リニアな曲率とリニアな捩率関数を定義して、これをフレネー・セレの式を用いて積分することによって一本の曲線セグメン

トを得、これが端末条件を満足するようにパラメータを決める。リニアな曲率とリニアな捩率を持つ曲線は2Dクロソイドの3D拡張とも言えるので、十分「3Dクロソイド曲線」と呼ぶに足る形状と性質を持っている。さらに、端末条件を指定することのためにパラメータが不足する場合には曲率、捩率のそれぞれを多連のリニア関数で構成することによって、必要な数のフリーパラメータを得ることになる。このようにして得られた曲線は多様な拘束条件に適用できる。

以下、この曲線（群）の求め方を示し、求めた曲線の形状を図示する。

2. 3D軌道曲線の持つべき特性

本論文では曲線上の位置ベクトルを $\mathbf{p}(s)$ と記す。ここで、 s は曲線始点からの曲線長（弧長）を表わす。この場合、曲線全体にわたって、 $d^n \mathbf{p}(s) / ds^n$ ($n=0$, または正整数) が s の連続関数であれば、 C^n 連続な曲線と言う。

質点がある速度で曲線上を運動する場合、その法線方向の加速度は曲線の曲率に比例する。滑らかな曲線運動を実現するには、曲線軌道の全体にわたって、曲率と法線方向の両方が連続でなければならない。即ち、曲線の全体にわたって、少なくとも C^2 連続である。 C^2 連続な曲線はゼロ曲率点を除いて、その曲率とフレネーフレームの両方を連続にする必要がある。捩率の連続性は必ずしも要求されない。

3. 3D曲線の表現方法

3.1 3D曲線の微分形式

原点を始点とする全長 h の3D曲線セグメントを式(1)で与える。ここで、 s は始点からの曲線長、 \mathbf{p} は曲線上の位置ベクトル、 \mathbf{t} 、 \mathbf{n} および \mathbf{b} はそれぞれ曲線上の単位接線(tangent)ベクトル、単位主法線(normal)ベクトルおよび単位従法線(binormal)ベクトルである。また、互いに直交する $[\mathbf{t} \ \mathbf{n} \ \mathbf{b}]$ の組をフレネーフレームと言う。

* 原稿受付 平成23年10月4日

** 正会員 マッスル株式会社 (大阪市淀川区木川東2-5-35)

*** 正会員 静岡大学 (静岡県浜松市城北3-5-1)

† 名誉会員 牧野オートメーション研究所

(東京都杉並区高円寺南1-24-14-703)

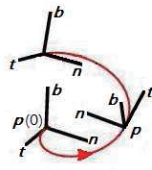


Fig.1 A 3D curve and its Frenet frame

$$\begin{cases} dp/ds = t \\ dt/ds = \kappa n \\ dn/ds = -\kappa t + \tau b \\ db/ds = -\tau n \\ p(0) = [0, 0, 0] = \theta \\ [t(0) \ n(0) \ b(0)] = [t_0 \ n_0 \ b_0] \\ 0 \leq s \leq h \end{cases} \quad (1)$$

τ と κ はそれぞれ曲線の曲率, 振率である. これらの変数を s の関数として定義する. $p(0) = [0, 0, 0] = \theta$ は曲線の始点を表わす. θ 以外の点を始点とする曲線の場合, 曲線全体をその位置ベクトル方向にオフセットするだけで良いので, この定式化の一般性は失われない. $[t_0 \ n_0 \ b_0]$ は 始点のフレネーフレームを表わす.

式 (1) の第 2~4 式はフレネー・セレの公式である. 始点のフレネーフレーム $[t_0 \ n_0 \ b_0]$, 曲線の全長 h , 曲率 κ と振率 τ を s の連続関数として指定すれば, それに対応する 3D 曲線が一意に定まる¹³⁾. このような 3D 曲線とフレネーフレームの様子を図 1 に示した.

主法線ベクトル n の向きが dt の向きと同じになるように規定されているので, 曲率 κ の数値はマイナスになることがない. $\tau = 0$ の区間では, その曲線は平面曲線になる. また, 本論文では $dt/ds = \kappa n$ を曲率ベクトル (curvature vector) と言う.

3.2 計算時における $\kappa = 0$ と $\kappa < 0$ の曲率の導入

従来, n と b の決まらない曲率 $\kappa = 0$ の点の特異点として式 (1) から除外されていたが, 実用的には $\kappa = 0$ の点を含む必要がある. さらに, 平面曲線において符号付曲率が用いられるように, S 形状の曲線セグメントを表すために曲率 κ はマイナスの数値を含むことが望ましい. この際, ゼロ曲率点の法線と従法線の向きとして前ステップの向きをそのまま保持すれば良い. $\kappa < 0$ の区間では 式 (1) による計算された n と b の向きが $\kappa > 0$ として定義されたフレネーフレームの法線と従法線の向きと逆になるが, 計算上では問題にならない. また, t , n と b の間には以下の関係が成り立つ.

$$b = t \times n \quad (2)$$

本論文では計算の途中でマイナスの曲率も用いるが, その計算結果は伝統的なフレネーフレームの定義に従って, マイナスの曲率の部分プラスの曲率に再変換している.

3.3 3D 曲線微分形式の無次元化

曲線長 s をその全長 h で割った無次元曲線長 S を定義し, これを用いて長さに関係のある量を次式のように無次元化する.

$$\begin{cases} S = s/h \\ P = p/h \\ K = h\kappa \\ T = h\tau \\ 0 \leq S \leq 1, \quad 0 \leq s \leq h \end{cases} \quad (3)$$

これらの変数を用いると, 式 (1) は式 (4) に示した無次元式に置き換えることができる.

$$\begin{cases} dP/dS = t \\ dt/dS = K n \\ dn/dS = -K t + T b \\ db/dS = -T n \\ P(0) = [0, 0, 0] = \theta \\ [t(0) \ n(0) \ b(0)] = [t_0 \ n_0 \ b_0] \\ 0 \leq S \leq 1 \end{cases} \quad (4)$$

無次元の式 (4) を解いてから, 式 (3) により, 実際の曲線長 s , 位置 p , 曲率 κ と振率 τ を計算すればよい. 式 (4) の計算では, 弧長が 1 に正規化された曲線を扱うことに相当する. 任意弧長 h の曲線は, この正規化された曲線を h によってスケールすることによって得られるので, 計算上のメリットが大きい. 以下では, P を無次元の位置ベクトル, K を無次元の曲率, T を無次元の振率と言う.

3.4 3D 曲線微分方程式の数値解

一般に式 (4) の解は初等関数で表せないため, 数値法で解かなければならない. このような連立常微分方程式の数値解を得るには幾つかの方法がある. 筆者らは 4 次の Runge-Kutta 法 (以下 RK4) を採用した. RK4 は差分ステップ長の 4 次精度を持つことが証明されており, 安定性も優れている¹⁷⁾. 曲線始点でのフレネーフレーム $[t_0 \ n_0 \ b_0]$, 無次元の曲率関数 $K(S)$, 無次元の振率関数 $T(S)$ と曲線の全長 h を指定すれば, RK4 により式 (4) を積分すると, それに対応する曲線セグメントが計算できる. 本論文ではこれを曲線セグメントの順解と言う.

4. 指定の拘束条件を満たす軌道曲線の導出方法

4.1 軌道曲線の拘束条件と必要な $K(S)$, $T(S)$

次に p_0 と p_1 の 2 端点を通る軌道設計を議論する. 部品を配置或いは加工する場合, ロボットアームの先端が指定の位置を指定の方向で通過し, 指定の方向へ旋回させる場合, 接線方向と法線方向の指定が必要である. このような拘束条件の一部は以下に示した.

- R1. p_0 での接線方向と p_1 での接線方向.
- R2. p_0 でのフレネーフレームと p_1 での接線方向.
- R3. p_0 での接線方向と p_1 でのフレネーフレーム.
- R4. p_0 でのフレネーフレームと p_1 でのフレネーフレーム.
- R5. R4 の条件に加えて, p_0 での曲率と p_1 での曲率.
- R6. R5 の条件に加えて, p_0 での振率と p_1 での振率.

本論文では, これらの条件を満たす $K(S)$, $T(S)$ と h を求めるのを曲線セグメントの逆解と言う. 上述の各拘束条件は以下の意味を持つ. R1 の場合, 端点での接線方向を指定するが, 端点での法線方向は指定しない. これは端点での曲線進行方向を指定するが, その湾曲方向を指定しないことを意味する. R2 と R3 の場合, 片端点の湾曲方向も指定している. R4 の場合, 端点での曲線進行方向とその湾曲方向の両方を指定している. R5 と R6 の場合, 端点での曲率や振率も指定しているため, ある指定した曲線との円滑な接続に必要となる.

筆者らの検証では, 曲率と振率を連続な多連区分線形関数として構成する場合, 以上の各拘束条件 R1~R6 に対応できる $K(S)$ と $T(S)$ 関数は以下の通りである.

- R1: リニアな $K(S)$ と定常な $T(S)$.

- R2, R3: リニアな $K(S)$ とリニアな $T(S)$.
 - R4: リニアな $K(S)$ と 2 連リニアな $T(S)$.
 - R5: 3 連リニアな $K(S)$ と 2 連リニアな $T(S)$.
 - R6: 3 連リニアな $K(S)$ と 4 連リニアな $T(S)$.
- 以下は適用性の高い R4, R5 と R6 の逆解を詳述する.

4.2 端点のフレネーフレームを指定する場合の逆解

次に 曲線の始点 p_0 と終点 p_l , 曲線始点と終点のフレネーフレーム $[t_0 \ n_0 \ b_0]$ と $[t_l \ n_l \ b_l]$ を指定した場合, その条件を満たす曲線セグメントを導出する. 式を簡単にするため, p_0 を原点とする座標系で議論する.

p_0 を始点とする曲線は終点 p_l でのフレネーフレームに合わせるのに 3 つの独立パラメータが必要であり, また曲線の終点は p_l に到着しなければならないので, 更に 3 つの独立パラメータが必要である. これらを合計して 6 個の独立パラメータが必要である. 曲線の全長 h は 1 個のパラメータに配分されるので, 残りの 5 個独立パラメータは式 (5) のように $K(S)$, $T(S)$ に配分する.

$$\begin{cases} K(S) = K_0 + (K_1 - K_0)S & 0 \leq S \leq 1 \\ T(S) = \begin{cases} T_0 + (T_m - T_0)S/S_m & 0 \leq S \leq S_m \\ T_m + (T_l - T_m)(S - S_m)/(1 - S_m) & S_m < S \leq 1 \end{cases} \end{cases} \quad (5)$$

ここで, $K(S)$ は無次元曲線長のリニア関数であり, $T(S)$ は無次元曲線長の 2 連リニア関数である. 定数 S_m ($0 < S_m < 1$) は関数 $T(S)$ 内部のつなぎ目の位置を表す. 特別な理由が無ければ $S_m = 1/2$ を取ればよい.

式 (5) 中の 5 個パラメータ K_0 , K_1 , T_0 , T_m と T_l はそれぞれ曲線始点の無次元曲率, 曲線終点の無次元曲率, 曲線始点の無次元振率, 曲線内部つなぎ目位置の無次元振率と曲線終点の無次元振率である.

具体的なアルゴリズムを以下に示す. まず, K_0 , K_1 , T_0 , T_m と T_l のある数値組を選定し, 式 (5) から計算された $K(S)$ と $T(S)$ を式 (4) に代入して, RK4 によりその曲線を計算する. このように得られた無次元曲線の終点を $P(1)$ とし, 得られた曲線の終点でのフレネーフレームは $[t(1) \ n(1) \ b(1)]$ とする. 続いて, 以下の目的誤差 e_{ptn} を計算する.

$$\begin{cases} e(p) = |P(1) / P(1) - p_l / p_l|^2 \\ e(t) = |t(1) - t_l|^2 \\ e(n) = |n(1) - n_l|^2 \\ e_{ptn} = e(p) + e(t) + e(n) \end{cases} \quad (6)$$

ここで, $P(1) = |P(1)|$, $p_l = |p_l|$ である. $e(p)$ は $P(1)$ 方向の単位ベクトルと p_l 方向の単位ベクトルとの誤差二乗を表し, $e(t)$ は曲線終点での単位接線ベクトルと設定された終点 p_l での単位接線ベクトルとの誤差二乗を表し, $e(n)$ は曲線終点の単位主法線ベクトルと設定された終点 p_l での単位主法線ベクトルとの誤差二乗を表す. また, $e(p)$, $e(t)$, $e(n)$ のどれも 4 以下の数値しか取らないので, 標準化した誤差成分と言える.

式 (6) の誤差 e_{ptn} はゼロに近づくと, 曲線の終点とその弦ベクトル p_l に到着し, しかも曲線終点のフレネーフレームも指定された終点 p_l でのフレネーフレームに一致することを意味する. この際, 曲線の全長 h は以下の式で計算できる.

$$h = |p_l| / |P(1)| \quad (7)$$

式 (7) は無次元曲線の終点を設定された終点 p_l に伸ばすため

の拡大或いは縮小率である. このように拡大或いは縮小しても, 曲線端点のフレネーフレームが変わらないことに注意する. 式 (6) の誤差関数 e_{ptn} は K_0 , K_1 , T_0 , T_m と T_l の非線形関数であり, その最小数値を得るため, 非線形最適化法を利用できる. 具体的には最急降下法や共役勾配法¹⁸⁾などの方法があるが, 目的誤差関数の偏導数を計算しなければならないため, 計算負荷と精度の面から避けたい. 筆者らは Nelder-Mead の シンプレックス法¹⁹⁾ (以下 NM 法) を用いた. NM 法は目的関数の偏導数 (勾配) を全く使わないので, 計算上の利点がある. 目的誤差 e_{ptn} は K_0 , K_1 , T_0 , T_m と T_l の 5 変数関数であるので, NM 法を使うには これらの変数を成分とする 5 次元ベクトルを 6 個作る必要がある. また, NM 法の収束はその初期シンプレックスに依存するので, その初期値を適切に選定する必要がある. 初期シンプレックスの原点 $[SK_0, SK_1, ST_0, ST_m, ST_l]$ は K_0 , K_1 , T_0 , T_m と T_l の各係数を限定な区間に適切な間隔で走査し, その目的誤差 e_{ptn} を最小になるような数値組を取ればよい. 以上のように初期シンプレックスの原点 $[SK_0, SK_1, ST_0, ST_m, ST_l]$ を求めれば, 式 (8) のように初期シンプレックスを作成できる.

$$\begin{cases} [SK_0 \ SK_1 \ ST_0 \ ST_m \ ST_l] \\ [SK_0 + d \ SK_1 \ ST_0 \ ST_m \ ST_l] \\ [SK_0 \ SK_1 + d \ ST_0 \ ST_m \ ST_l] \\ [SK_0 \ SK_1 \ ST_0 + d \ ST_m \ ST_l] \\ [SK_0 \ SK_1 \ ST_0 \ ST_m + d \ ST_l] \\ [SK_0 \ SK_1 \ ST_0 \ ST_m \ ST_l + d] \end{cases} \quad (8)$$

式 (8) の d はシンプレックスの初期サイズを決定する. 筆者らの数値実験では $d = \pi/8$ が妥当である. 上述の初期シンプレックス原点を探索する段階では, 大まかな近似曲線を得れば良いので, 式 (4) の解を得るのに 10 分割の RK4 を用いれば十分である. 探索間隔もやや大きく (例えば $\pi/8$) 取るほうが適切である. 特に始点の法線方向を保持する必要があるから, $K_0 \geq 0$ に限定する必要があるが, K_1 はマイナスの数値を取っても構わない (S 字カーブ曲線の場合). また, その逆解曲線が一回転以上廻らないように, 以下の式を満たす必要がある.

$$\begin{cases} \theta_K = \int_0^1 K(S) dS = (K_0 + K_1) / 2 < 2\pi \\ \theta_T = \int_0^1 T(S) dS = (T_0 + 2T_m + T_l) / 4 < 2\pi \end{cases} \quad (9)$$

ここで, θ_K と θ_T はそれぞれ曲線セグメントの湾曲角と振り角を意味する. 一回転以上に廻るような逆解曲線は確かに存在するが, 適切な軌道曲線ではないので, 避けなければならない.

図 2 にこのように計算された曲線例を示す. ここで, 曲線始点と終点のフレネーフレーム $[t_0 \ n_0 \ b_0]$ と $[t_l \ n_l \ b_l]$ は赤色, 緑色, と青色の棒で表わす. 赤色ポイントは曲線の始点を示し, 青色ポイントは曲線の終点を示す. 曲線の曲率ベクトル $\kappa \mathbf{n}$ を緑色のラインで示している. 始点からの曲線長を横軸としての曲率 (青) と振率 (赤) の分布図をその下部のグラフに示している. ここで, 曲率がゼロになる点 (変曲点) の前後では, 曲率ベクトルの向きが逆になることに注意する.

アルゴリズム全体のフロー図を図 3 に示す. 無次元曲線の数値計算において, RK4 は差分ステップ長の 4 次精度を持つので, 0.01 の差分ステップ長を採用しても, 実用に耐える精度の良い

結果が得られる。より高い精度が望ましい場合、差分ステップ長を更に小さくすれば達成できる。

4.3 指定した点列を通過する[$t \ n \ b$] 連続軌道設計

指定した点列を通過する軌道設計問題を考える。その点列(点数= $N+1$)を以下のように表わす。

$$P_0, P_1, P_2, \dots, P_N \tag{10}$$

ここで、 P_k から P_{k+1} に向かう単位ベクトルをその間の弦方向単位ベクトルと言う。各通過点でのフレネーフレームを実際の制約状況に応じて、指定するのは普通である。本論文は一つの計算例として、以下の規則を用いる。

ある通過点前後の弦方向単位ベクトルを CV_1 と CV_2 とすると、この通過点でのフレネーフレームを以下の式で指定する。

$$\begin{cases} t = (CV_1 + CV_2) / |CV_1 + CV_2| \\ b = t \times CV_2 \\ n = b \times t \end{cases} \tag{11}$$

式(11)は接線ベクトル t の方向が通過点前後の弦ベクトル方向の中間方向(合成方向)に向いて、通過点での曲線の接平面が接線ベクトル t と弦ベクトル CV_2 を含む平面と一致することを意味する。全ての通過点において、式(11)のようにその通過点でのフレネーフレームを指定して、区間毎に4.2節のような目標曲線を求めると、その自由点列を通過する軌道を得る。得られた軌道曲線はそのフレネーフレームが連続であり、曲線の各部分は式(5)のリニアな曲率と振率セグメントによって構成される。図4はこのような曲線例である。

この図例の点列は立方体(辺長=100)の8個の頂点に位置する。曲線の曲率ベクトル(緑)、曲率(青)および振率(赤)の分布図を示している。各点の曲線長に対する位置を横軸の丸い点で表わす。曲線全体の曲り方が緩やかであるが、一般に指定した各通過点においては、曲線の曲率と振率は連続ではない。また、曲率図を常にゼロ以上という伝統に従って描画したこと注意到。

4.4 指定した点列を通過する C^2 連続軌道設計

上述の[$t \ n \ b$]連続曲線では各通過点において、そのフレネーフレームが連続になるが、一般にその曲率は不連続である。これを言い換えると、その曲線は C^2 連続ではない。

各通過点での接線と法線方向指定を変えずに、[$t \ n \ b$]連続曲線から C^2 連続曲線を得るのに、各通過点での曲率を指定できれば実現できる。そのためには隣接通過点間の曲線セグメントの自由度を増やす必要がある。これは各曲線セグメントの $K(S)$ を三連リニア関数に換えれば、可能である(4.1節参照)。この場合の $K(S)$ と $T(S)$ 関数は以下の式(12)になる。

ここで、 $K(S)$ は無次元曲線長の三連リニア関数であり、 $T(S)$ は無次元曲線長の二連リニア関数である。定数 S_1, S_2 と S_m は曲率と振率関数内部のつなぎ目の位置を表す。特別な理由が無ければ $S_1=1/4, S_m=1/2, S_2=3/4$ を取れば良い。

式(12)のパラメータ $K_0, K_{S_1}, K_{S_2}, K_1$ は以下の意味を持つ。 K_0 と K_1 は曲線始点と終点での無次元曲率、 K_{S_1} と K_{S_2} は曲線内部つなぎ目 S_1 と S_2 での無次元曲率を表わす。

式(5)と比べると、式(12)の無次元曲率関数 $K(S)$ は単一のリニア関数から三連のリニア関数になるが、曲線の導出方法は同じである。この場合、独立なパラメータの数が2個増えるので、曲線セグメントの始点と終点での曲率を指定できる。

各通過点において、曲率連続な曲線を求める前にまず式(5)の無次元曲率関数 $K(S)$ と $T(S)$ を用いて、4.3節に述べた

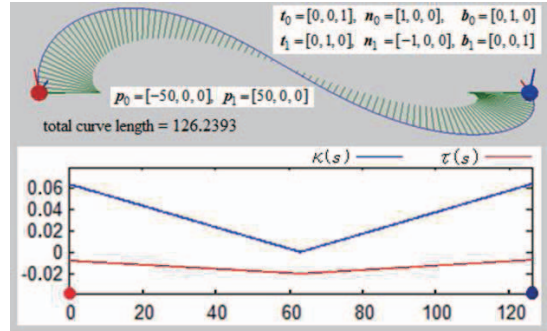


Fig.2 Resultant curves matching the Frenet-frame constraints of $[t_0 \ n_0 \ b_0]$ and $[t_1 \ n_1 \ b_1]$ at its terminals. The curvature vectors (green), curvature (blue) and torsion (red) distribution are also plotted

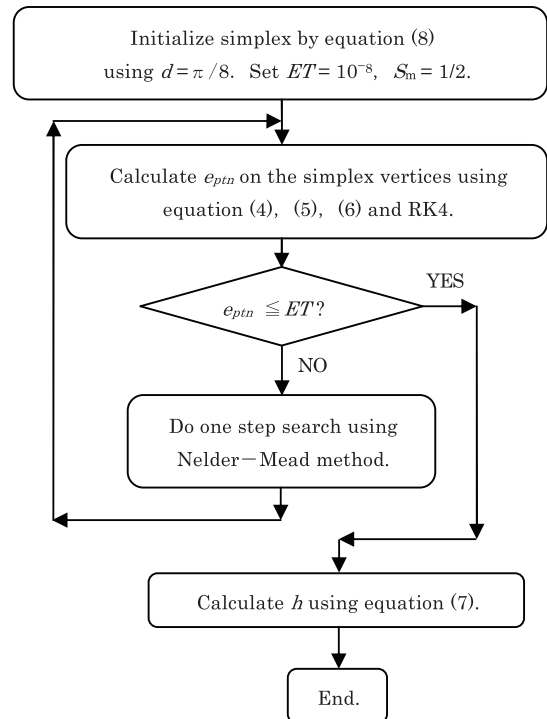


Fig.3 Algorithm flow chart for the Frenet frame matching

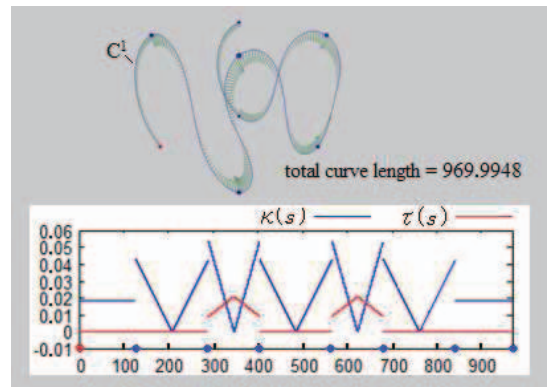


Fig.4 Path generation with Frenet frame continuity. The points are taken from a cube's vertices. The curvature vectors (green), curvature (blue) and torsion (red) distribution are also plotted

$$K(S) = \begin{cases} K_0 + (K_{S_1} - K_0)S/S_1 & 0 \leq S \leq S_1 \\ K_{S_1} + (K_{S_2} - K_{S_1})(S - S_1)/(S_2 - S_1) & S_1 < S \leq S_2 \\ K_{S_2} + (K_1 - K_{S_2})(S - S_2)/(1 - S_2) & S_2 < S \leq 1 \end{cases} \quad (12)$$

$$T(S) = \begin{cases} T_0 + (T_m - T_0)S/S_m & 0 \leq S \leq S_m \\ T_m + (T_1 - T_m)(S - S_m)/(1 - S_m) & S_m < S \leq 1 \end{cases}$$

ような[*t n b*]連続曲線を計算する。この際、各通過点での曲率が連続しないが、各通過点の前段補間曲線の終点曲率とその後段補間曲線の始点曲率を求めて、曲率が低いほうの曲率をこの通過点での目標曲率とする。補間したい区間において、このように求めた始点と終点での曲率をそれぞれ κ_s , κ_e とすると、始点と終点での目標曲率に合わせるのに以下の式 (13) を満たす必要がある。

$$\begin{cases} K_0 = h\kappa_s \\ K_1 = h\kappa_e \end{cases} \quad (13)$$

ここで、次元のある曲率を合わせるのに得られた補間曲線セグメントの長さ *h* を用いる。この場合、8 個のパラメータ K_0 , K_{S_1} , K_{S_2} , K_1 , T_0 , T_m , T_1 および *h* を求める必要があるが、式 (13) の関係が成り立つので、 K_{S_1} , K_{S_2} , T_0 , T_m , T_1 と *h* の 6 個パラメータを NM 法に組めればよい。即ち、 C^2 連続曲線を得るための初期シンプレックスは式 (8) の代わりに式 (14) のように与える。

$$\begin{cases} [SK_{S_1} SK_{S_2} ST_0 ST_m ST_1 Sh] \\ [SK_{S_1} + d SK_{S_2} ST_0 ST_m ST_1 Sh] \\ [SK_{S_1} SK_{S_2} + d ST_0 ST_m ST_1 Sh] \\ [SK_{S_1} SK_{S_2} ST_0 + d ST_m ST_1 Sh] \\ [SK_{S_1} SK_{S_2} ST_0 ST_m + d ST_1 Sh] \\ [SK_{S_1} SK_{S_2} ST_0 ST_m ST_1 + d Sh] \\ [SK_{S_1} SK_{S_2} ST_0 ST_m ST_1 Sh + dh] \end{cases} \quad (14)$$

ここで、 SK_{S_1} , SK_{S_2} , ST_0 , ST_m および ST_1 は既に計算した[*t n b*]連続曲線の式 (5) から導出できる。*Sh* はその曲線長 *h* を用いる。 $d = \pi/8$ と $dh = Sh/8$ が妥当である。また、NM 法計算の過程で *h* を更新するので、式 (13) により K_0 と K_1 を更新しなければならない。NM 法の目的誤差関数は式 (6) の代わりに式 (15) を用いる。

$$\begin{cases} e(p) = |h\mathbf{P}(1) - \Delta\mathbf{p}|^2 / |\Delta\mathbf{p}|^2 \\ e(t) = |t(1) - t_1|^2 \\ e(n) = |n(1) - n_1|^2 \\ e_{pin} = e(p) + e(t) + e(n) \end{cases} \quad (15)$$

ここで、 $\Delta\mathbf{p}$ は補間する区間終点と始点のベクトル差である。 $h\mathbf{P}(1)$ は計算された曲線セグメントの弦長ベクトルなので、この両者の差が 0 であれば、計算された曲線の終点とその補間する区間の終点と同じになる。

パラメータの数は異なるが、そのアルゴリズムは 4.2 節のそれと類似する。図 5 はこのように計算した曲線例である。点列データは図 4 と同じものを使っている。比較のため、4.3 節のフレネーフレーム連続曲線を左上グラフの緑色曲線 (C^1) で示す。 C^2 連続な曲線の曲率ベクトル (緑)、曲率 (青) および振率 (赤) の分布図も示している。 C^2 連続な曲線は各指定した通過点において、そのフレネーフレームも曲率も連続である。ただし、各通過点での振率は連続ではない。

4.5 指定した点列を通過する $C^{2+\tau}$ 連続な軌道設計

前節の C^2 連続曲線は とても実用的な軌道曲線と言えるが、

振率の連続性を追加すればさらに改良できる。各通過点での[*t n b*]指定を変えずに、[*t n b*]連続曲線から $C^{2+\tau}$ 連続曲線を得るのに、各通過点での曲率と振率の両方を指定できれば実現できる。そのためには隣接通過点間の曲線セグメントの自由度を更に増やす必要がある。これは式 (12) の $T(S)$ 関数は以下の式 (16) にすれば良い。

$$T(S) = \begin{cases} T_0 + (T_{S_1} - T_0)S/S_1 & 0 \leq S \leq S_1 \\ T_{S_1} + (T_m - T_{S_1})(S - S_1)/(S_m - S_1) & S_1 < S \leq S_m \\ T_m + (T_{S_2} - T_m)(S - S_m)/(S_2 - S_m) & S_m < S \leq S_2 \\ T_{S_2} + (T_1 - T_{S_2})(S - S_2)/(1 - S_2) & S_2 < S \leq 1 \end{cases} \quad (16)$$

4.4 節の式 (12) と比べて、振率関数 $T(S)$ だけは二連リニアな関数から四連リニアな関数になっている。新しく増やしたパラメータ T_{S_1} , T_{S_2} は曲線内部つなぎ目 S_1 と S_2 での無次元振率を表す。その以外のパラメータは式 (12) の通りである。

各通過点での曲率と振率の指定方法も 4.4 節と類似する。即ち、まず 4.3 節に述べたような[*t n b*]連続曲線を計算する。それから、各通過点の前段補間曲線の終点曲率・振率とその後段補間曲線の始点曲率・振率を求めて、曲率が低いほうの曲率及び振率が低いほうの振率をこの通過点での目標曲率及び振率とする。補間したい区間において、求めた始点と終点での曲率及び振率をそれぞれ κ_s , κ_e , τ_s , τ_e とする。この場合、式 (13) に続いて以下の式 (17) を追加すればよい。目的誤差関数は式 (15) と変わらず、そのアルゴリズムも 4.4 節と類似する。

$$\begin{cases} T_0 = h\tau_s \\ T_1 = h\tau_e \end{cases} \quad (17)$$

図 6 はこのように計算した曲線例である。同じく図 4 の点列データを使っている。図から分かるように、曲線全体に渡って、その曲率と振率の両方が連続である。図 5 の C^2 連続曲線と比べて、図 6 の $C^{2+\tau}$ 連続曲線は僅かに長い。

図 7 は閉じた点列の $C^{2+\tau}$ 連続な接続例を示す。この点列は正四面体 (辺長 = 1 0 0) の頂点であるが、点列の終点 (5 点目) はその始点と同じとしている。他の例と同様、曲率ベクトル (緑)、曲率 (青) と振率 (赤) の分布図をそのグラフに示している。

以上の各図例では距離単位の指定が任意である。例えば距離単位を cm とすると、曲率と振率の単位が rad/cm となる。また式 (3) の無次元パラメータは点列図の形状だけに依存し、各点間の間隔に依存しない。例えば、点列の各点間の間隔を 2 倍に拡大すると、得られる補間曲線はその各区間の曲線長 *h* だけが 2 倍になって、その他の無次元パラメータが変わらない。これは無次元化計算のメリットである。

4.6 異なる平面にある二直線間の三次元接続

ここでは、異なる平面上に存在する二直線間の接続について簡単に述べる。このような接続は端点の接続方向を接続したい直線の方向に一致させればよい。 $\kappa_s = \kappa_e = 0$ を指定して、4.4 節のアルゴリズムを採用すれば、 C^2 連続な接続ができる。更に $\kappa_s = \kappa_e = \tau_s = \tau_e = 0$ を指定して、4.5 節のアルゴリズムを採用すれば、 $C^{2+\tau}$ 連続な接続になる。特に曲線の湾曲方向を指定する必要がなく、単純な振率が望ましい場合、接点の法線方向を自由度として使えるので、三連リニアな曲率関数と定常な振率関数で接続可能である。

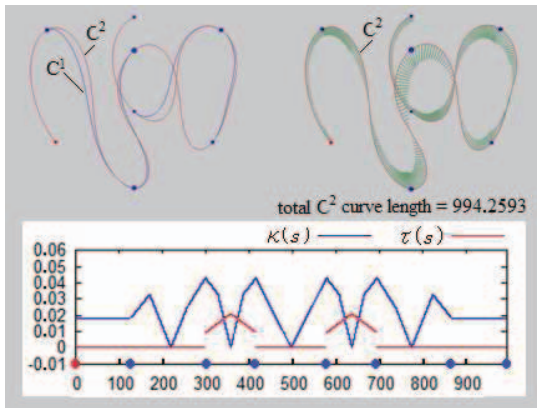


Fig.5 Path generation with C^2 continuity. The points are taken from a cube's vertices. The curvature vectors(green), curvature(blue) and torsion(red) distribution are also plotted

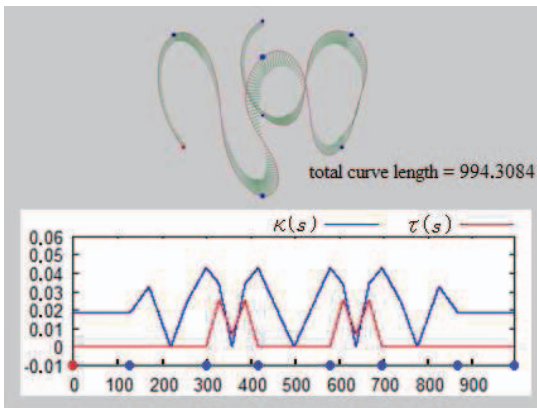


Fig.6 Path generation with $C^{2+\tau}$ continuity. The points are taken from a cube's vertices. The curvature vectors(green), curvature(blue) and torsion(red) distribution are also plotted

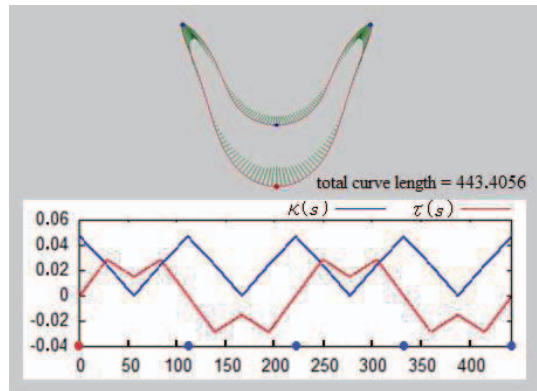


Fig.7 Path generation with $C^{2+\tau}$ continuity. The points are taken from a regular tetrahedron's vertices. Its final point (the 5th point) equals its first point. This makes a closed curve. The curvature vectors(green), curvature(blue) and torsion(red) distribution are also plotted

5. 結 言

本研究では 区分リニアな曲率と振率関数を定義して、無次元な 3D 曲線の微分式からその曲線を求めた。与えられたすべての

通過点において、そのフレネーフレーム、曲率と振率を指定できる運動軌道の設計法を示した。このように設計された軌道曲線はリニアな曲率と振率セグメントにより構成され、動力学的に優れた特性を持つ。得られた曲線軌道はその任意点の曲線長、フレネーフレーム、曲率と振率を正確に計算できるので、軌道上の運動体について、その到達位置、運動姿勢と加速度を正確に予測できる。このように計算された C^2 連続な 3D 曲線は与えられた通過点でのフレネーフレームを自由に指定できるので、実用的に柔軟性の高い軌道曲線と言える。フレネーフレームを基準にした姿勢変化を更に滑らかに制御したい場合、振率も連続な $C^{2+\tau}$ 連続な軌道曲線を用いれば良い。

計算時間については、2.13GHz の i3 CPU、4GB RAM の Windows 7 PC で確認した。この中に、100 分割の 4 次 Runge-Kutta(RK4)法を使って、Nelder-Mead のシンプレックス法(NM 法)の収束許容誤差は $ET=10^{-8}$ を用いた。この場合、3.4 節の RK4 による式(4)の順解計算は約 0.5ms かかり、4.5 節の RK4 と NM 法による 8 点を通る $C^{2+\tau}$ 連続な補間曲線の逆解 (Fig.6) は約 3900ms かかった。

点列の各区間に本論文で示すような一回転以内の逆解曲線のほかに、多重回転のような解も存在するので、一般に解の一意性がないことに注意したい。また、逆解のないケースを見つけなかったが、逆解の存在性についてはまだ解明していない。今後の課題として、アルゴリズムの改良および C^3 連続以上の軌道曲線構成法を検討する。また各拘束条件において、その逆解曲線の存在性についても解明したい。

参 考 文 献

- 1) 牧野洋, 接線法を用いた自由点列のクロソイド補間, 精密工学会誌 **60**, 1 (1994)80.
- 2) 仇時雨, 牧野洋, 須田大春, 横山恭男, クロソイドによる自由曲線補間法, 日本ロボット学会誌, **8**, 1(1990)680.
- 3) 黒田満, 斉藤剛, 渡辺由美子, 東正毅, 曲率連続な双クロソイド/第2クロソイドスプライン補間曲線, 精密工学会誌 **63**, 6(1997)801.
- 4) Meek D., Walton D., Clothoid spline transition spirals. Mathematics of Computation **59**, 7(1992)117
- 5) Josef Stoer. Curve fitting with clothoidal splines. J. Res. Nat. Bur. Standards, **87**, 4(1982)317.
- 6) 蘭豊礼, 玉井博文, 牧野洋, 三連クロソイドによる自由点列補間, 精密工学会誌 **76**, 10 (2010)1194.
- 7) Li Guiqing, Li Xianmin, Li Hua, 3D Discrete Clothoid Splines, Computer Graphics International (2001 Proceedings), 321.
- 8) D. R. Forsy and R. H. Bartels. Hierarchical B-spline refinement. Computer Graphics, **22**, 4(1988)205.
- 9) H. S. Goghary and S. Abbasbandy. Interpolation of fuzzy data by Hermite polynomial. Int. J. Comput. Math., **82**, 5(2005)595.
- 10) M. Gasca and J. I. Maeztu. On Lagrange and Hermite interpolation in R^k . Numer. Math., **39**, 1(1982)1.
- 11) 三浦憲二郎, 単位 4 元数積分曲線, 情報処理学会論文誌, **38**, 11(1997)2227.
- 12) 小林左千夫, 木村文彦, 牧野洋, 3次元クロソイド曲線を用いた補間手法, 精密工学会秋季大会講演会講演論文集(2004).
- 13) Erwin Kreyszig. Differential Geometry. Dover Publications, (1991).
- 14) Mehlum, E.: Nonlinear splines. Computer Aided Geometric Design, (1974)173.
- 15) Henry Moreton. Minimum Curvature Variation Curves, Networks and Surfaces for Fair Free-Form Shape Design. PhD thesis, University of California, Berkeley, (1992).
- 16) N. Yoshida, R. Fukuda, T. Saito, Log-Aesthetic Space Curve Segments, SIAM/ACM Joint Conference on Geometric and Physical Modeling (GDSPM2009), 35.
- 17) Timothy Sauer, Numerical Analysis, Addison Wesley, (2005).
- 18) N, Jorge and W, Stephen, Numerical Optimization, Springer, (2006)
- 19) J. A. Nelder and R. Mead, A simplex method for function minimization, Comput. J., 7(1965):308.
- 20) 牧野洋, 3次元機構学, 日刊工業新聞社(1998).