RECONSTRUCȚIA 3D A IMAGINII PICĂTURII ÎN PROCESUL DE TURNARE A MICROFIRULUI

Autori: Petre Plotnic, Sergiu Zaporojan, Constantin Plotnic
Universitatea Tehnică a Moldovei

Abstract: *Forma picăturii în procesul de turnare a microfirului este un indice cheie în luarea deciziilor de reglare de către operatorul uman antrenat în această activitate. Deoarece sistemul inteligent de monotorizare al acestui proces[4] presupune acumularea unei baze de cunoştinţe ce conţine date numerice despre caracteristicile principale din procesul de turnare, inclusiv despre forma picăturii, este important ca printr-un număr limitat de date să fie posibilă reconstrucţia imaginii virtuale a picăturii în spaţiul 3D. Pentru a putea realiza mai uşor şi mai rapid modelul virtual al picăturii, datele iniţiale obţinute sub forma unei tabele de puncte, cu ajutorul unor softuri sunt transformate în curbe B-splines sau suprafeţe B-splines, aceste softuri având în spate o intreagă bibliotecă cu formule matematice (ca exemplu, sistemul MATLAB).*

Cuvinte cheie: *Microfir, picătură, model virtual, B-splines, interpolare liniară, interpolare spline cubică.*

Pentru modelarea virtuală a picăturii, în procesul de turnare a microfirului, se utilizează o funcţie de interpolare asociată unei tabele de puncte date de pe conturul real al picăturii.

Din punct de vedere matematic, o curbă generată pornind de la vîrfurile care definesc un poligon depinde de o anumită metodă de interpolare sau aproximare, care stabileşte legătura dintre curbă şi poligon.

În lucrarea [5] s-a folosit interpolarea liniară cu ajutorul polinomului Lagrange.

Reamintim că aproximarea Lagrange foloseşte următoarele funcţii de bază:

$$L_j(x) = \frac{(x-x_0)...(x-x_{j-1})(x-x_{j+1})...(x-x_p)}{(x_j-x_0)...(x_j-x_{j-1})(x_j-x_{j+1})...(x-x_p)}$$

şi polinomul Lagrange $P(x)$, asociat:

$$P(x) = \sum_{j=0}^{p} y_j L_j(x).$$

unde $x_0<x_1<x_2<...<x_p$ sînt abcisele punctelor de control, iar $y_0,y_1,y_2,...,y_n$ respectiv ordonatele lor.

Un avantaj al reprezentării funcţiei de aproximare sub forma Lagrange o constitue faptul că polinoamele L_j depind numai de alegerea nodurilor şi nu cer condiţii de restricţii auxiliare.

Două caracteristici ale acestora limitează flexibilitatea curbelor rezultate:

- numărul vîrfurilor poligonului determină gradul polinomului care defineşte curba;
- natura globală a polinoamelor Lagrange face ca o modificare a poziţiei unui singur vîrf să se facă simţită pe întreaga curbă; practic se elimină posibilitatea de a produce scimbări locale în curbe.

Interpolarea *spline cubică* înlătură aceste dezavantaje ale polinoamelor Lagrange. Comportamentul nonglobal al curbelor B-spline se datorează faptului că fiecărui vîrf i se asociază o singură funcţie de bază. În consecinţă, fiecare vîrf afectează forma curbei numai în jurul acestuia, şi anume pe domeniul în care funcţia de bază asociată este nenulă. De asemenea se poate modifica ordinul funcţiilor de bază fără a se modifica numărul vîrfurilor date.

O curbă B-spline este descrisă prin relaţia:

$$P(t) = \sum_{i=0}^{n} P_i N_{i,k}(t)$$

unde:

- P_i sînt cele $(n+1)$ vîrfuri date;
- $N_{i,k}$ sînt funcţiile de amestec, numite şi funcţii B-spline, de ordin k definite recursiv astfel:

$$N_{i,1}(t) = \begin{cases} 1, \text{ dacă } x_i \le t < x_{i+1} \\ 0, \text{ în caz contrar} \end{cases}$$

$$N_{i,j}(t) = \frac{t - x_i}{x_{i+k-1} - x_i} N_{i,j-1}(t) + \frac{x_{i+1} - t}{x_{i+k} - x_{i+1}} N_{i+1,j-1}(t) \ \ pentru \ \ 1 < j < k$$

- valorile x_i sînt elemente ale unui vector de noduri $(x_0, \ldots x_{n+k})$ care va fi descris mai jos;
- parametrul t ia valori în intervalul $[x_0, x_{n+k}]$.

Vectorul de noduri este elementul adițional care conferă curbelor B-spline flexibilitate. În majoritatea cazurilor se folosesc vectori de forma:

$$(\underbrace{0\,0\ldots0}_{k\ ori}\ 1\,2\ldots m-1\ \underbrace{m\,m\ldots m}_{k\ ori}) \qquad \text{unde } m = n\text{-}k\text{+}2.$$

În acest caz funcția *P(t)* este un polinom de grad *k-1* pe fiecare interval *[xi,xi+1]*.
P(t) şi derivatele sale de ordin *1, 2, ... k-2* sînt continue pe tot intervalul *[0,m]*.
Pentru *k=2* se obține un caz particular: linia poligonală însăşi.
Pentru *k=n* se obține un alt caz particular; în acest caz parametrul *t* ia valori în intervalul *[0,1]* şi se obține curba Bézier.

Pentru aproximarea numerică a funcției *f* de o variabilă reală prin *interpolare liniară*, *interpolare cu polinom Hermite* sau *interpolare cu funcții spline*, sistzemul MATLAB pune la dispoziția utilizatorului funcția *interp1* cu următoarea sintaxă de apel:

vy=interp1(x,y,vx,'metoda')

unde:
x - este vectorul punctelor *{xi}*;
y - este vectorul punctelor *{yi}*;
vx - este vectorul punctelor în care se doreşte aproximarea funcției *f*;
vy - este vectorul obținut prin aproximarea funcției *f* în punctele *vx*;
metoda - reprezintă un şir de caractere prin care se precizează metoda de interpolare dorită:
linear – pentru interpolare *liniară* (este metoda implicită);
cubic sau *pchip* – pentru interpolare cu polinom *Hermite cubic pe porțiuni*;
spline – pentru interpolare *spline cubică*.
Interpolarea cu funcții spline poate fi realizată şi cu funcția MATLAB *spline*, care are următoarea sintaxă de apel:

vy=spline(x,y,vx)

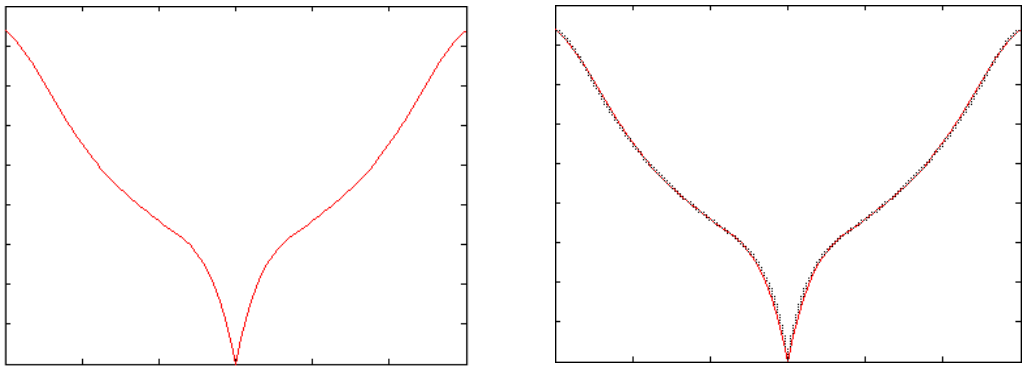
în care parametri *x,y,vx* şi *vy* au aceleaşi semnificații ca şi în cazul funcției *interp1*.

De fapt, methoda *'spline'* a funcției *interp1*, pentru realizarea interpolării, apelează fără implicarea utilizatorului funcția MATLAB *spline*.

Următorul cod:

```
x=[0.0, 0.8, 1.6, 2.4, 4.2,  5.0, 6.0];
y=[0.0, 2.5, 3.3, 3.9,  5.8,  7.1, 8.4];
xx=[0.0, -0.8, -1.6, -2.4, -4.2,  -5.0, -6.0];
xi=0:0.1:6;
xxi=-6:0.1:0;
yi=spline(x,y,xi);
yyi=spline(xx,y,xxi);
plot(xi,yi,'r',xxi,yyi,'r')
```

realizat în MATLAB, desenează profilul picăturii, inițial dat prin şapte puncte de control, utilizând interpolarea spline cubică, figura 1(a). În figura 1(b) este afişat rezultatul unui program Java, care realizează un algoritm de interpolare liniară prin polinoame Lagrange,suprapus pe graficul obținut în MATLAB.



a) b)

Figura 1. Reprezentarea picăturii prin şapte puncte : a), b) graficile suprapuse căpătate prin cele două metode.

La suprapunerea celor două grafice observăm că profilurile căpătate, prin două metode diferite, practic coincid, ceea ce demonstrează selectarea reuşită a numărului de puncte pentru metoda Lagrange. În cazul cînd numărul de puncte este diferit de şapte, graficul căpatat prin interpolare liniară diferă mult de profilul real al picăturii. În continuare pentru asigurare, în cazul modificării numărului punctelor de control, vom utiliza interpolarea *spline cubică*.

Ideia reconstrucţiei formei picăturii în spaţiul *3D* este: o suprafaţă oarecare de rotaţie este generată prin rotirea unei linii oarecare, dreaptă sau curbă, plană sau strâmbă, în jurul unei drepte numită axa suprafeţei. În cazul dat ca generatoare vom folosi curba de profil a picături căpătată prin interpolare spline cubică utilizînd şapte puncte de control, proiectată în planul *(x,z),* iar axa *Oz* se ia ca axă a suprafeţei.

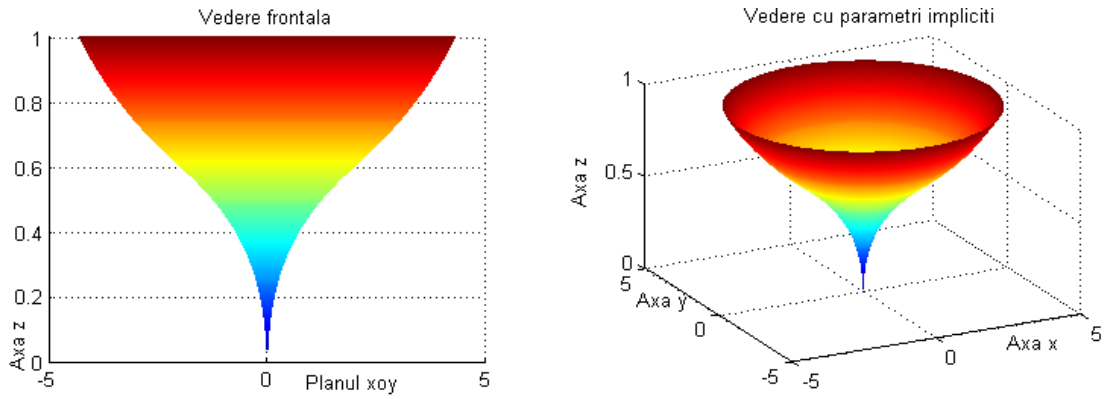
Funcţia *cylinder* din MATLAB, generează un cilindru de rază *R*, cu cercul bazei aproximat din *N* puncte echidistante. Se apelează cu sintaxa:

[x,y,z]=cylinder(R,N)

Funcţia returnează matricele cu *2 X (N+1)* elemente, care specifica vârfurile fiecărei suprafeţe rezultate din aproximarea cercurilor intermediare bazelor cu poligoane cu *N* laturi. Vectorul *R* are două elemente *[R1 R2]*, care precizeaza raza obiectului la partea inferioară *(R1)* şi superioară *(R2),* prin aceasta fiind posibilă construirea de conuri, trunchiuri de con, piramide, trunchiuri de piramide etc. Valoarea implicită este pentru *R=[1 1],* iar pentru *N=20*. În cazul cînd modificăm valoarea lui *R* de la *R1* pînă la *R2* cu punctele de pe curba generatoare obţinem circumferinţe ce aproximează suprafaţa cvazicilindrică a picăturii. Reprezentarea grafică a suprafeţei se face cu funcţia *surf(x,y,z).*

Executarea umătorului cod în sistemul MATLAB, afişază, pe ecran, modelul virtual al picăturii în spaţiul *3D*,figura 2.

```
clear all;
axis([-6 6 -6 6 0 9]);
axis equal;
xi=[0:.1:6];
zi=[0:.1:6];
x=[0.0 0.8 1.6 2.4 4.2 5.0 6.0];
y=[0.0 2.5 3.3 3.9 5.8 7.1 8.4];
yi= spline(y,x,zi);
subplot(121)
[X,Y,Z] = cylinder(yi,30);
surf(X,Y,Z);
view( 0,0);
title('Vedere frontala');
xlabel('Axa x');
ylabel('Axa y');
zlabel('Axa z');
shading interp;
subplot(122)
[X,Y,Z] = cylinder(yi,30);
surf(X,Y,Z);
view( -37.5,30);
title('Vedere cu parametri impliciti');
xlabel('Axa x');
ylabel('Axa y');
zlabel('Axa z');
shading interp;
```



a) b)

Figura 2. Modelul virtual al picăturii în spațiul *3D*.

Poziția observatorului față de un obiect *3D* se precizează prin unghiul pe orizontală (numit azimut) şi unghiul pe verticală (numit elevație). Funcția *view* este utilizată pentru vizualizarea unei reprezentări grafice spațiale din diverse poziții; se apelează cu sintaxa:

view(az,elv);

unde: *az* este azimutul (sau unghiul în plan orizontal), iar *elv* este unghiul pe verticala (ambele în grade). Azimutul poate lua valori pozitive sau negative, cea pozitivă fiind o rotire în jurul axei *Oz* în sens orar. Valorile pozitive ale unghiului pe verticală corespund unui punct de observare plasat deasupra planului *(x,y),* iar valorile negative corespund punctelor de sub plan.

view(3) - stabileşte reprezentarea grafica *3D* cu valorile implicite: *az=-37.5, elv=30.*

Pentru orientare:

elv=90 - vederea de deasupra obiectului.

*az=elv=*0 - vedere directa din planul zero.

az=180 - vedere din spate a obiectului.

Figura 2(a) corespunde vederii directe din planul zero, iar figura 2(b) corespunde vederii cu valori implicite. Schimbînd valoarea unghiului în plan orizontal şi respectiv, valoarea unghiului pe verticală vom obține imaginea virtuală a picăturii din diferite puncte de observare.

Bibliografie

[1] Marin Ghinea, Virgiliu Fireţeanu. MATLAB,Calcul numeric,Grafică,Aplicaţii.Editura Teora, Bucureşti,2012.

[2] Florica Moldoveanu. Grafica pe calculator . Editura Teora, Bucureti,. 1996.

[3] M. Postolache.Metode numerice. Editura Sirius, Bucureşti,1994.

[4] S. Zaporojan, C. Plotnic, I. Calmicov, V. Larin. A knowledge-based approach for microwire casting plant control. In: J. Jozefczyk and D. Orski (Eds). *Knowledge-Based Intelligent System Advancements:systemic and cybernetic approaches,* Information Science Reference, Hershey New York, 2010.

[5] S. Zaporojan, C. Plotnic, I.r Calmicov. Some of the aspects of decision design in development of the intelligent wire casting machine.*Second International Conference Modelling and Development of Intelligent Systems*,Sibiu - Romania, September 29 - October 02, 2011