

ASSISTING THE CHOICE OF DRILLING TOOLS BY PROCESSING STEP FILES

BĂTRÎNU Andrei-Alexandru

Facultatea: Inginerie Industrială și Robotică, Specializarea: Informatică aplicată în ingineria industrială,
Anul de studii: 4, e-mail: batrinuandrialexandru1298@gmail.com

Conducător științific: Prof.dr.ing. Tom SAVU

REZUMAT: Having a STEP file as input data, the developed algorithm will extract information about the holes in the part and thus it will process a solution for drilling them. Following the analysis of each hole it will display the start point of the hole, the distance traveled by the drill, and the dimensions of it.

CUVINTE CHEIE: step, găurire, fabricare

1. Introducere

În domeniul proiectării CAD exista multe tipuri de fișiere neutre precum IGES, STEP, DXF, STL, OBJ, AMF, 3MF; cele mai populare dintre acestea fiind IGES și STEP. Fișierele de tip STEP sunt destinate schimbului de date despre produs, iar cele IGES servesc schimbului de date geometrice. Formatul STEP este recunoscut pe plan mondial drept formatul neutru pentru aproape toate softurile CAD. Acronimul STEP provine de la “STandard for the Exchange of Product model data”.

În prezent, pentru a pregăti procesele de prelucrare prin aşchiere se utilizează software-uri de fabricare asistată (CAM), cum ar fi Fusion 360, SolidCAM, NX CAM etc. În acest software utilizatorul introduce reperul proiectat și, în baza suprafețelor acestuia, configurează prelucrările. La final acesta va exporta codul G dedicat mașinii declarată în software-ul CAM. Acronimul CAM provine de la “Computer-aided manufacturing”.

Codul G este cel mai utilizat limbaj de programare al mașinilor unelte cu comandă numerică (CNC). Acesta este utilizat pentru a ordona prelucrările ce vor fi făcute de mașina cu comandă numerică și are sintaxe diferite în funcție de controller-ul mașinii. În funcție de sintaxa citită din codul G, controller-ul acționează motoarele mașinii cu comandă numerică.

Pentru procesul de găurire, utilizatorul trebuie să aleagă mașina unealtă pentru care va genera codul G, să specifice locul unde va fi poziționată gaura, apoi acesta va configura parametri de prelucrare cum ar fi tipul sculei aşchietoare, dacă este burghiu sau freză, diametrul acesteia, distanța parcursă de sculă, și parametri de tehnologicitate.

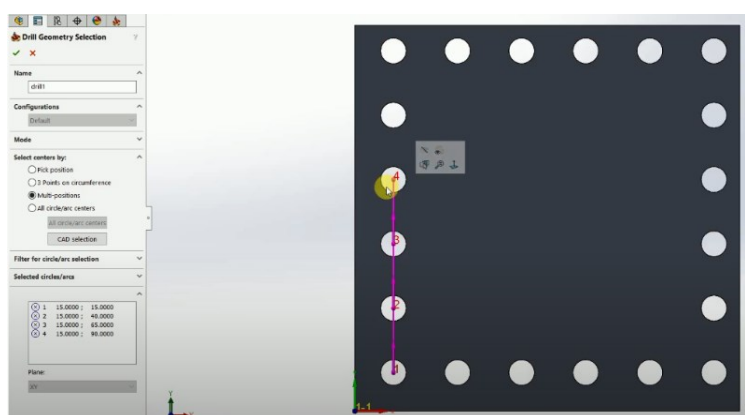


Fig. 1. Precizarea poziției găurilor de către utilizator în SolidCAM

2. Structura unui fișier STEP

Formatul STEP este divizat în două părți: HEADER și DATA. În HEADER este prezentată o descriere a reperului, numele acestuia și o schemă. În DATA se găsesc informații legate de geometria piesei, suprafețele acesteia, dimensiuni și poziționare. Liniile din DATA încep cu „#” urmat de un număr, reprezentând indexul aferent acestora. Informațiile reperului sunt obținute parcurgând liniile fișierului STEP, începând cu linia care poartă eticheta „CLOSED_SHELL” și urmărind în continuare fiecare index din componența sa. Datele sunt structurate precum în Fig.2.

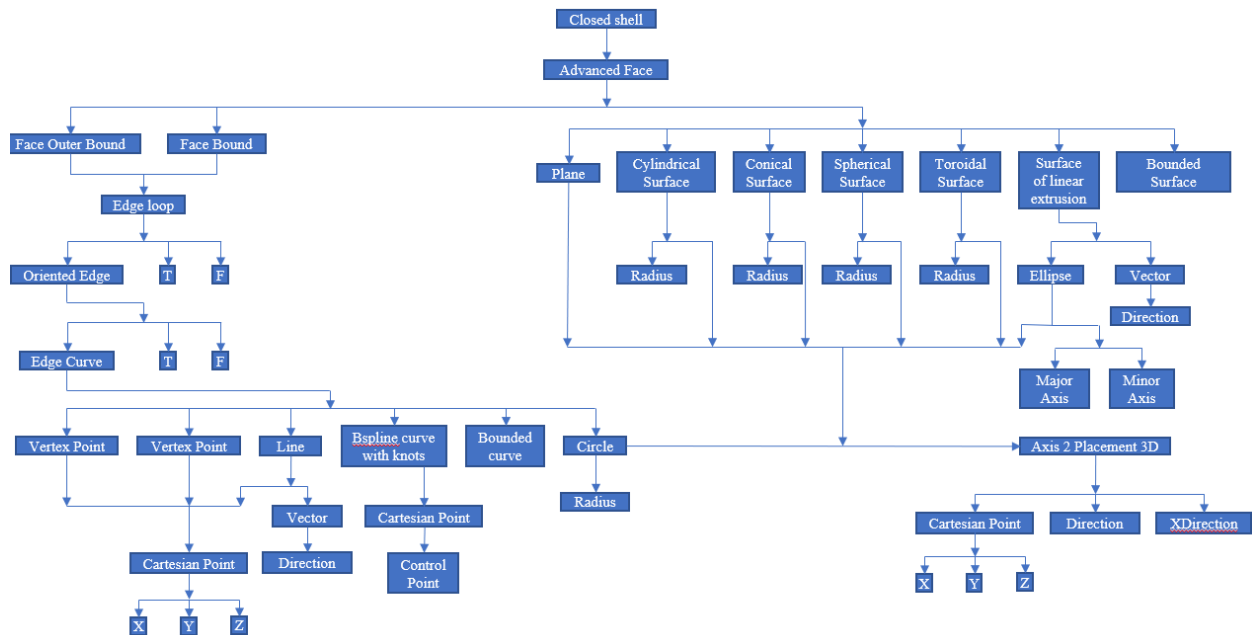


Fig. 2. Structura datelor în STEP

Entitățile care oferă informații despre găurile unui reper în format STEP sunt următoarele:

- `##d=CLOSED_SHELL(,"(șir de suprafețe));`

Această entitate reprezintă reperul și conține un șir de indexuri care fac trimitere la entitățile numite „ADVANCED_FACE”.

Exemplu: `##467=CLOSED_SHELL(",(#440,#441,#442,#443,#444,#445,#446,#447,#448,#449,#450,#451,#452,#453,#454,#455,#456,#457,#458,#459,#460,#461,#462,#463,#464,#465,#466));`

- `##d=ADVANCED_FACE(,"(șir de fețe atașate), geometria suprafeței, .boolean.);`

Entitatea „ADVANCED_FACE” reprezintă o suprafață a reperului. Aceasta este definită de fețe care formează suprafața respectivă, geometria suprafeței și un boolean.

Exemplu: `##463=ADVANCED_FACE(",(#423,#424),#438,.F.);`

- `##d=CYLINDRICAL_SURFACE(," axa, raza*10-3);`

Această entitate reprezintă geometria suprafeței, în cazul acesta cilindrică, și este definită de axa pe care este orientată suprafața și raza acesteia și raza înmulțită cu 10^{-3} .

Exemplu: `##438=CYLINDRICAL_SURFACE(",#526,0.0035);`

- `##d=CIRCLE(," axa, raza*10-3);`

Această entitate reprezintă geometria unei curbe care formează marginea unei fețe și este definită de o axă față de care este orientată în sistemul cartezian xyz și în cazul acesta deoarece este un cerc mai este definită și de rază care este înmulțită cu 10^{-3} .

Exemplu: #44=CIRCLE(",#529,0.0025);

- #%d=ELLIPSE(", axă, semi-axa majora, semi-axa minora);

Această entitate reprezintă geometria unei curbe care formează marginea unei fețe și este definită de o axă față de care este orientată în sistemul cartezian xyz și în cazul acesta deoarece este o elipsa mai este definită de două semi-axe înmulțite cu 10^{-3} .

Exemplu: #309=ELLIPSE(",#488,0.00362346663143529,0.0035);

3. Algoritm de identificare a găurilor

Procese ale algoritmului:

- P1. Citirea fișierului STEP

Pentru a parcurge eficient fișierul, acesta se va citi pe rânduri deoarece în DATA fiecare rând este indexat.

- P2. Identificarea tuturor suprafețelor care formează reperul

Parcurge fiecare linie a fișierului STEP până când o găsește pe cea etichetată „CLOSED_SHELL”. Această entitate este definită de toate suprafețele reperului, prin urmare, identifică indexurile tuturor suprafețelor.

- P3.1. Identificarea geometriei suprafețelor

Parcurge fiecare linie a fișierului STEP până când găsește indexurile extrase din „CLOSED_SHELL” cu eticheta „ADVANCED_FACE”. Această entitate este definită de mai multe fețe, geometria suprafeței și un boolean, prin urmare, pentru fiecare suprafață, identifică indexul cu geometria acesteia.

- P3.2. Identificarea fețelor de pe fiecare suprafață

Parcurge fiecare linie a fișierului STEP până când găsește indexurile extrase din „CLOSED_SHELL” cu eticheta „ADVANCED_FACE”. Această entitate este definită de mai multe fețe, geometria suprafeței și un boolean, prin urmare, pentru fiecare suprafață, identifică toate indexurile cu fețe care o formează.

- P4. Identificarea marginilor sau a colțurilor pentru fiecare față

Parcurge fiecare linie a fișierului STEP până în momentul în care găsește indexurile extrase din „ADVANCED_FACE” cu eticheta „FACE_BOUND”. Această entitate este definită de o margine sau un colț și un boolean, prin urmare, pentru fiecare față identifică indexul marginii sau al colțului.

- P5. Identificarea muchiilor orientate pentru fiecare margine

Parcurge fiecare linie a fișierului STEP până când găsește indexurile extrase din „FACE_BOUND” cu eticheta „EDGE_LOOP”. Această entitate înglobează un șir de muchii orientate, prin urmare, pentru fiecare margine, algoritmul identifică indexurile cu muchii orientate.

- P6. Identificarea curbei pentru fiecare muchie

Parcurge fiecare linie a fișierului STEP până când găsește indexurile extrase din „EDGE_LOOP” cu eticheta „ORIENTED_EDGE”. Această entitate este caracterizată de o curbă și un boolean, prin urmare, pentru fiecare muchie, identifică indexul curbei.

- P7.1. Identificarea geometriei pentru fiecare curbă

Parcurge fiecare linie a fișierului STEP până când găsește indexurile extrase din „ORIENTED_EDGE” cu eticheta „EDGE_CURVE”. Această entitate este definită de un punct de început, unul de final, geometria curbei și un boolean, prin urmare, pentru fiecare curbă, algoritmul identifică indexul geometriei.

- P7.2. Identificarea punctelor pentru fiecare suprafață

Parcurge fiecare linie a fișierului STEP până când găsește indexurile extrase din „FACE_BOUND” cu eticheta „VERTEX_LOOP”. Această entitate este caracterizată de un punct cartezian. Algoritmul

formează pentru fiecare suprafață un șir de puncte în urma concatenării șirului de puncte identificate în entitatea "EDGE_CURVE" cu șirul de puncte identificate în "VERTEX_LOOP".

- P8. Formarea unui șir de suprafețe cilindrice

Având date pentru fiecare suprafață, cum ar fi axa, curbe și puncte, algoritmul separă suprafețele cilindrice de celelalte suprafețe și formează un șir de structuri cu informații specifice.

- P9. Gaură întreruptă

Algoritmul compară axa și raza suprafețelor, iar dacă găsește două suprafețe cilindrice pe aceeași axă, având aceeași rază, calculează cea mai mare distanță dintre cele patru puncte și pe acelea două le declară capetele suprafeței cilindrice, în caz contrar, acestea nu se modifică.

Formula distanței dintre două puncte în coordonate xyz este:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

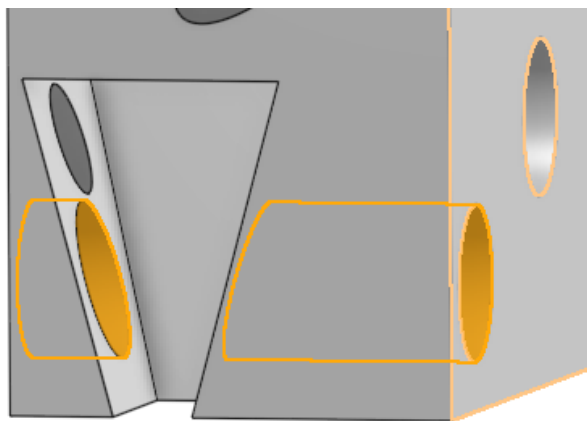


Fig. 3. Gaură întreruptă

- P10. Gaură înfundată sau străpunsă

Algoritmul caută suprafețe conice sau plane care conțin doar o curbă. Verifică dacă aceasta este un cerc de rază egală cu o suprafață cilindrică și dacă se află pe aceeași axă. Dacă centrul cercului corespunde cu unul din capetele suprafeței cilindrice, atunci algoritmul definește gaura ca fiind înfundată respectiv, dacă nu îndeplinește aceste condiții, o definește drept gaură străpunsă.

4. Implementarea în LabVIEW

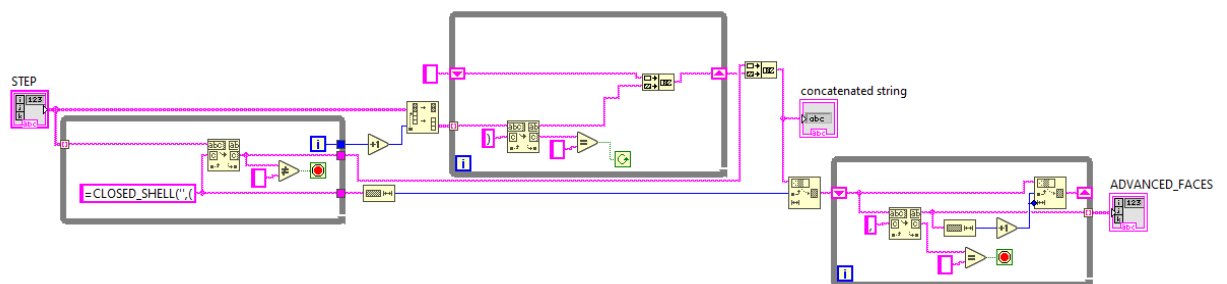


Fig. 4. SubVI pentru identificarea suprafețelor

SubVI-ul pentru identificarea suprafețelor (Fig.4) parcurge array-ul cu linii al fișierului STEP până când găsește entitatea „CLOSED_SHELL” și extrage din aceasta indexurile suprafețelor reperului.

Pașii parcurși de SubVI sunt:

- Găsește în STEP entitatea „CLOSED_SHELL”

Utilizând o buclă „While”, la fiecare iterație este căutată secvența „=CLOSED_SHELL(”, „(” în câte o linie a fișierului STEP. Bucla se oprește în momentul în care a găsit o linie cu acea secvență. Pentru

căutarea în linie a secvenței „=CLOSED_SHELL(”,(” este utilizată funcția „Search/Split String” care are ca intrări un string în care va căuta, un string de căutat și un integer pentru a ști unde să înceapă căutarea, iar ca ieșiri² are un string pe care îl găsește înaintea secvenței căutate, un string cu secvența găsită și ce urmează după aceasta și un integer cu indexul unde a găsit secvența.

Pentru că entitatea „CLOSED_SHELL” se află pe mai multe linii și bucla „While” se termină după ce a găsit linia cu secvența aferentă. SubVI-ul reține indexul la care a găsit secvența și împarte array-ul cu linii din STEP în două de la următorul index. În altă buclă „While” caută elementul „)” în array-ul care începe de la următorul index față de unde a găsit entitatea „CLOSED_SHELL” și până când îl găsește concatenează liniile parcurse, acestea făcând parte din entitatea „CLOSED_SHELL”.

Pentru a obține un singur string care conține entitatea „CLOSED_SHELL”, SubVI-ul concatenează string-ul pe care îl găsește la prima buclă „While” cu string-ul format în a doua buclă „While”.

- Creează un șir cu indexurile suprafețelor

În entitatea „CLOSED_SHELL” indexurile suprafețelor sunt despărțite de „, ”, astfel, printr-o buclă „While”, string-ul cu entitatea „CLOSED_SHELL” este trecut printr-un registru de transfer și la fiecare iterație este căutat în string elementul despărțitor. String-ul găsit înaintea elementului despărțitor este scos din buclă și eliminat din entitate, pregătind string-ul pentru următoarea iterație.

5. Testarea aplicației

Testarea aplicației constă în citirea unui reper cu găuri aleatorii în format STEP și verificarea rezultatelor obținute în urma rulării programului pentru a observa dacă acestea coincid cu datele reale despre gaură.

Exemplu gaură din reper:

Punct de început[mm]: (11, 0, 34)

Punct de final[mm]: (11, 5, 34)

Raza găurii[mm]: 3

Tip: Înfundată

Gaura formează unghi de 90° cu axa x, 180° cu axa y și 90° cu axa z.

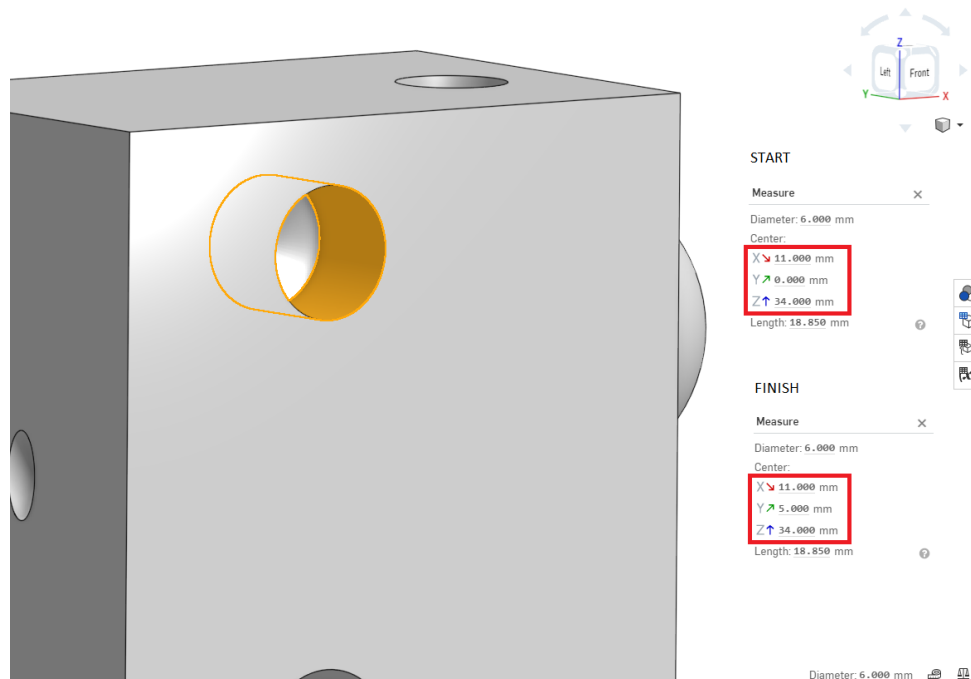


Fig. 5. Exemplu gaură

PUNCT	PUNCT	PUNCT	PUNCT	PUNCT	PUNCT	PUNCT	PUNCT	PUNCT
x	x	x	x	x	x	x	x	x
0.0402199	0.00370172	0.011	0.024	0.011	0.006	0.0246277	0.0247951	0.003
y	y	y	y	y	y	y	y	y
0.0290462	0.015	-0.0392076	0.009	0.00340129	0.016	0.008	-0.0392076	0.004
z	z	z	z	z	z	z	z	z
0.0379312	0.0279656	0.012	0.0352489	0.034	0.00510086	-0.0186564	0.0057018	0.003
DIRECTIE	DIRECTIE	DIRECTIE	DIRECTIE	DIRECTIE	DIRECTIE	DIRECTIE	DIRECTIE	DIRECTIE
x	x	x	x	x	x	x	x	x
113.47	180	90	90	90	90	120	90	90
y	y	y	y	y	y	y	y	y
131.771	90	0	90	180	90	90	0	180
z	z	z	z	z	z	z	z	z
129.093	90	90	0	90	180	30	90	90
START	START	START	START	START	START	START	START	START
x	x	x	x	x	x	x	x	x
0.03	-8.67362E-	0.011	0.024	0.011	0.006	0.0138564	0.0247951	0.003
y	y	y	y	y	y	y	y	y
0.0119519	0.015	0	0.009	0	0.016	0.008	0	0
z	z	z	z	z	z	z	z	z
0.0217497	0.0279656	0.012	0.04	0.034	0	0	0.0057018	0.003
FINISH	FINISH	FINISH	FINISH	FINISH	FINISH	FINISH	FINISH	FINISH
x	x	x	x	x	x	x	x	x
0.0228545	0.005	0.011	0.024	0.011	0.006	0	0.0247951	0.003
y	y	y	y	y	y	y	y	y
0	0.015	0.02	0.009	0.005	0.016	0.008	0.02	0.004
z	z	z	z	z	z	z	z	z
0.0104359	0.0279656	0.012	0.032	0.034	0.009	0.024	0.0057018	0.003
RAZA	RAZA	RAZA	RAZA	RAZA	RAZA	RAZA	RAZA	RAZA
0.00225	0.004	0.0035	0.0025	0.003	0.002	0.001	0.0035	0.0025
TIP GAURA	TIP GAURA	TIP GAURA	TIP GAURA	TIP GAURA	TIP GAURA	TIP GAURA	TIP GAURA	TIP GAURA
Patrunsa	Infundata	Patrunsa	Infundata	Infundata	Infundata	Patrunsa	Patrunsa	Infundata

Fig. 6. Rezultate obținute pentru gaura exemplificată

După cum se poate observa rezultatele obținute sunt înmulțite cu 10^{-3} deoarece așa sunt regăsite valorile în fișierul STEP. Structurile PUNCT și DIRECTIE fac parte din componența axei găurii. Reperul citit de aplicație conține nouă găuri.

6. Concluzii

Pentru aplicația de asistare a alegerii sculelor așchietoare de găurire prin procesarea fișierelor STEP s-a studiat formatul acestor fișiere, s-a elaborat un algoritm de extragere a găurilor dintr-un reper, s-a implementat algoritmul în mediul de programare LabVIEW și s-a testat aplicația pe un reper proiectat pe platforma Onshape care conține găuri aleatorii.

În continuare, pentru proiectul de diplomă se dorește crearea unei baze de date cu scule așchietoare și dezvoltarea aplicației de a alege pentru fiecare gaură o astfel de sculă în funcție de tipul găurii. De asemenea se dorește calcularea punctului de început și de sfârșit al prelucrării deoarece, pentru găurire, burghiul sau freza trebuie să înceapă mișcarea de rotație înainte să pătrundă în piesă. Pentru poziționarea semifabricatului pe masa de prelucrare se vor calcula unghiurile la care trebuie rotit.

7. Bibliografie

[1]. V. Naga Malleswari, L. Likhith Sai, P. Lohith Raj, and P. Tharun Kumar (2019), "Automatic recognition of geometrical dimensioning and tolerances from a STEP file", AIP Conference Proceedings 2200, nr. articol 020007

8. Notății

Următoarele simboluri sunt utilizate în cadrul lucrării:

d = distanța dintre două puncta în spațiu