

Program pentru comanda unui utilaj cu comanda numerica

Grupa: 30232

Laszlo Bogdan Gheorghe

Cuprins

Introducere	3
Context	3
Specificari	3
Obiective	3
Studiu bibliografic	3
Interpolare liniara	4
Interpolare circulara	4
Analiza	5
Implementare	7
Teste	9
Miscare rapida	9
Taiere liniara	11
Taiere circulara	12
Reset	13
Bibliografie	16

Introducere

Context

Scopul proiectului este de a implementa un simulator de taiere cu flama, primind semnale bazate pe un fisier continand traiectoria de taiere. Taierea se va simula pe ecranul calculatorului.

Orice om va putea folosi acest program cat timp se stie scrie un fisier conform conditiilor de computare pentru a folosi o masina de taiat cu flama automata.

Specificari

Programul va simula taierea pe ecranul calculatorului cu ajutorul IDE-ului IntelliJ. Va simula miscarea flamei pentru a efectua taierea in timp real.

Obiective

Scrierea unui fisier care continue traiectoria de taiere, preluarea informatiilor din fisier si proiectarea unui program care prelucreaza informatiile din fisier astfel incat acesta sa afle coordonatele punctele ce trebuie strabatute pe axa X si Y de flama masinii pentru taiere si proiectarea unui cod care simuleaza aceasta taiere pe ecranul calculatorului.

Studiu bibliografic

Codurile G sunt denumite și coduri/funcții pregătitoare. Acestea sunt orice cuvânt dintr-un program de CNC care începe cu litera G urmat de un număr. În general, este un cod care spune mașinii unelte ce tip de acțiune trebuie să efectueze, cum ar fi mișcarea rapidă (duce unealta cât mai repede posibil prin spațiu la locul unde va avea loc tăierea).

G01 – Interpolare liniara

G17 – Comanda de coordonate polare, selectarea planului XY

Etc.

Interpolare liniara

Interpolarea liniară este un proces care vă permite să deduceți o valoare între două valori bine definite, care pot fi într-un tabel sau într-un grafic liniar.

Dacă pentru o valoare x cunoscută, cuprinsă în intervalul x_1, x_2 trebuie să se determine valoarea y corespunzătoare, știind că aceasta este cuprinsă în intervalul y_1, y_2 (marginile intervalelor, x_1, x_2 , respectiv y_1, y_2 fiind de asemenea cunoscute) aceasta se poate calcula cu ajutorul uneia dintre relațiile:

$$y = y_1 + (x - x_1) \cdot (y_2 - y_1) / (x_2 - x_1) \quad \textbf{(relatia 1)}$$

$$y = y_2 + (x_2 - x) \cdot (y_2 - y_1) / (x_2 - x_1) \quad \textbf{(relatia 2)}$$

Interpolare circulara

Interpolarea circulară, cunoscută și sub numele de interpolare de-a lungul unei circulare, este o tehnică folosită pentru a estima valorile intermediare de-a lungul unui cerc sau unui arc de cerc, în funcție de un număr finit de puncte cunoscute de pe cerc.

Pentru a efectua interpolarea circulară trebuie urmăriți următorii pași:

- Avem nevoie de o serie de puncte cunoscute de-a lungul cercului sau arcului de cerc. Aceste puncte pot fi reprezentate sub forma coordonatelor polare (raza și unghiul) sau sub forma coordonatelor carteziane (x, y), dar este important să fie în relație cu centrul cercului.
- Calculăm centrul cercului, care este punctul de origine în sistemul de coordonate. Dacă avem coordonatele polare, centrul cercului este în $(0,0)$, iar dacă avem coordonatele carteziane, putem folosi metoda mijlocului segmentului pentru a găsi centrul.
- Pentru a estima valoarea pentru un punct intermediar pe cerc sau arc, utilizăm formule trigonometrice sau geometrice pentru a calcula coordonatele acestui punct.

De exemplu, pentru un cerc cu centrul în origine, raza R și unghiul θ care variază de la θ_1 la θ_2 , coordonatele pentru punctele de-a lungul arcului ar fi:

$$X = R * \cos(\theta) \quad Y = R * \sin(\theta)$$

pentru θ în intervalul $\theta_1 \leq \theta \leq \theta_2$.

Codurile G02 și G03 sunt proiectate pentru a efectua interpolare circulară. Codul G02 este folosit pentru a muta de-a lungul unui arc de cerc în sensul acelor de ceasornic, un G03 - invers acelor de ceasornic. Direcția de deplasare este determinată, atunci când ne uităm la partea axului instrument în direcția negativă a axei Z. Ca atunci când se efectuează o interpolare liniară în cadrul interpolare trebuie să specifice o circulară F. tăiere viteza de avans.

Analiza

Pentru rezolvarea acestui proiect vom împarti în mai multe parti problema:

1. Preluarea infomatiilor de la o interfata grafica
 - Se vor pune instructiunile într-un text field
 - Se verifica daca acestea sunt valabile prin folosirea unui cod REGEX
 "move/linear/circular" "X" "Y" ("I" "J" clockwise/counterclockwise)

The diagram illustrates a user interface for a CNC control system. It consists of a main container with a light gray background. On the left side, there is a control panel with a 'Text Field' at the top, followed by three buttons: 'Reset', 'Send', and 'Get Position'. Below these buttons is a label 'Current Position: X... Y...'. On the right side, there is a large rectangular area labeled 'GRID' in the center, representing the workspace for the tool path.

Figura 2 Interfata utilizatorului

- Se trimite comanda printr-un buton "Send" ce confirma in acelasi timp daca comanda transmisa este valida;
- Se trimite spre aplicatie si se genereaza secventa de G Code ce concide cu comanda data;
- Se vor calcula punctele folosind metodele de mai sus;
- Se va reprezanta taierea prin desenarea punctelor din grid;
- La sfarsit, utilizatorul poate da in continuare comenzi sau sa reseteze simularea prin apasarea butonului "Reset";
- Se poate apasa butonul "Get Position" pentru a afisa pozitia curenta a cnc-ului.

2. Implementarea interpolarilor si calcularea punctelor interpolate

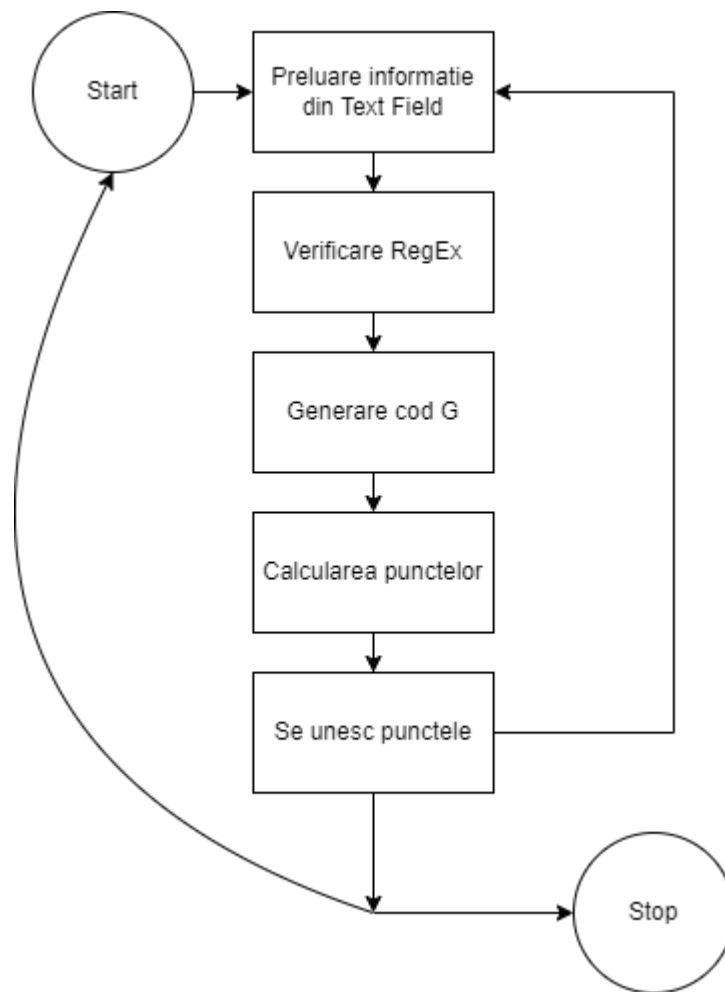


Figura 3 diagrama de functionare

Figura 4 diagrama de functionare

Implementare

Pentru miscarea rapida fara a face vreo taiere se foloseste functia din dreapta:

Pentru interpolarea liniara sa folosit logica explicata mai sus si este implementata astfel:

```
public void moveToGoal() {  
    if (goal.x > currentX) {  
        this.currentX += 1;  
    } else if (goal.x < currentX) {  
        this.currentX -= 1;  
    }  
    if (goal.y > currentY) {  
        this.currentY += 1;  
    } else if (goal.y < currentY) {  
        this.currentY -= 1;  
    }  
}
```

Figura 5 Cod miscare rapida

```
public void calculateLinearInterpolatedPoints(float startX, float startY, float endX, float endY, int numPoints) {  
    float interpolatedX = 0;  
    float interpolatedY = 0;  
    for (int i = 0; i ≤ numPoints; i++) {  
        float t = (float) i / numPoints;  
        interpolatedX = startX + t * (endX - startX);  
        interpolatedY = startY + t * (endY - startY);  
        setPixel((int) interpolatedX, (int) interpolatedY);  
  
        try {  
            Thread.sleep( millis: 10);  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
  
        SwingUtilities.invokeLater(this::repaint);  
    }  
  
    currentX = interpolatedX;  
    currentY = interpolatedY;  
}
```

Figura 6 Cod interpolare liniara

Variabilele startX si startY primesc coordonatele in care se afla cnc-ul in acel moment si endX si endY primesc punctul in care dorim sa ajungem prin taiere. Threadurile au fost folosite pentru simularea taierii. Dupa interpolare, pozitia cnc-ului ramane la ultimul punct interpolat.

```

public void calculateCircularInterpolationPoints(double endX, double endY, double centerX, double centerY, int clockwise) {
    double radius = Math.sqrt(Math.pow(centerX-currentX, 2) + Math.pow(centerY-currentY, 2));
    double startAngle = Math.atan2(currentY-centerY,currentX -centerX);
    double endAngle = Math.atan2(endY - currentY, endX - currentX);

    if (clockwise == 2) {
        // Ensure end angle is greater than start angle for counterclockwise interpolation
        if (endAngle <= startAngle) {
            endAngle += 2 * Math.PI;
        }
    } else {
        // Ensure end angle is less than start angle for clockwise interpolation
        if (endAngle >= startAngle) {
            endAngle -= 2 * Math.PI;
        }
    }

    double angleIncrement = (endAngle - startAngle) / numPoints;
    double currentAngle;
    for (int i = 0; i <= numPoints; i++) {
        currentAngle = startAngle + i * angleIncrement;
        currentX = (float) (centerX + radius * Math.cos(currentAngle));
        currentY = (float) (centerY + radius * Math.sin(currentAngle));
        setPixel((int) currentX, (int) currentY);
        System.out.println(currentX + " " + currentY);
        if (checkGoalReached())
            break;
        try {
            Thread.sleep( millis: 10);
        } catch (InterruptedException e) {
            e.printStackTrace();
            Thread.currentThread().interrupt();
        }
    }

    SwingUtilities.invokeLater(this::repaint);
}

```

Figura 7 Cod interpolare circulara

Pentru interpolarea circulara explicate mai [sus](#) se foloseste functia de mai jos:

Teste

Miscare rapida

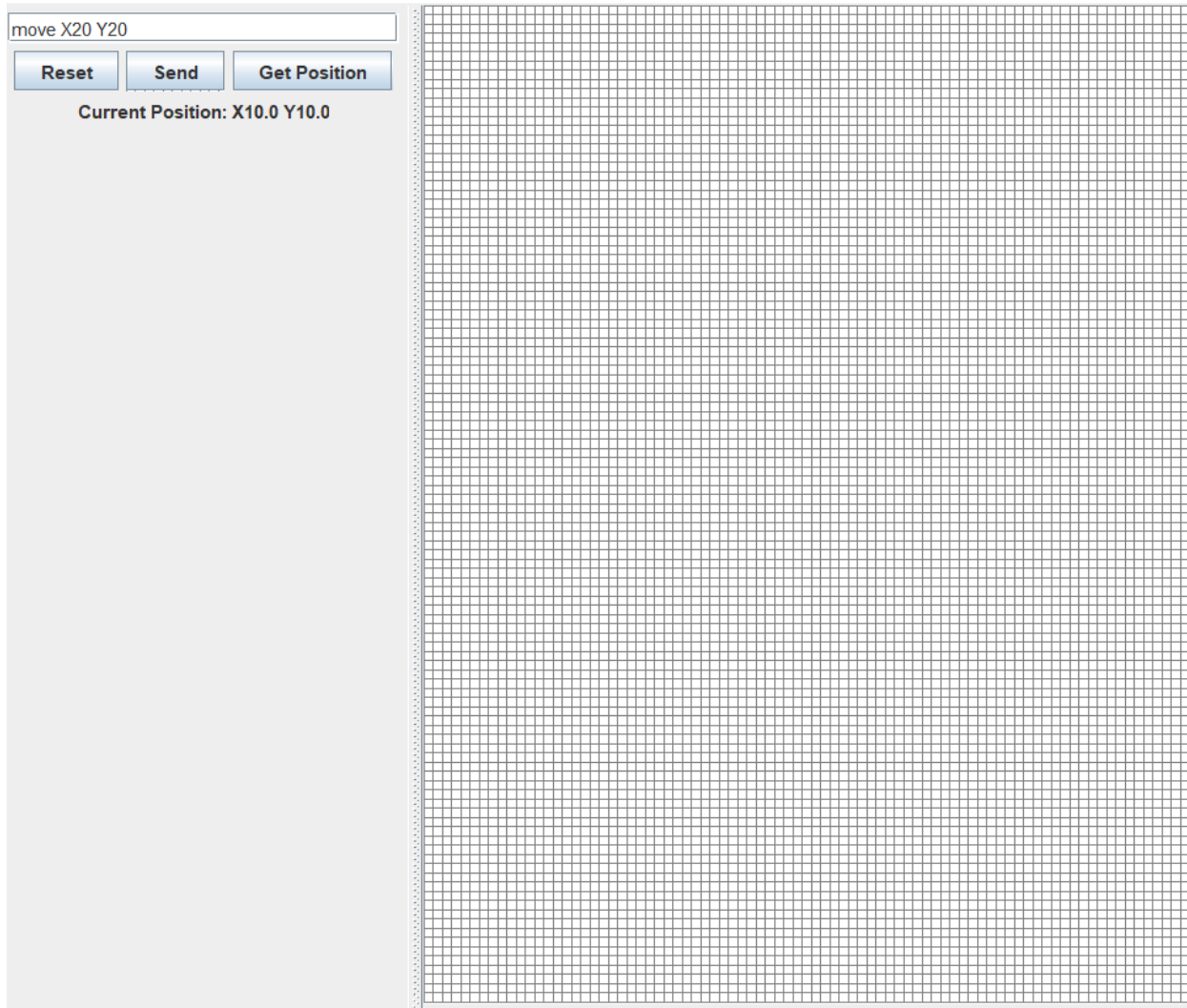


Figura 8 UI dupa introducerea in text-field miscarea rapida la X20 Y20

Dupa trimiterea comenzii move X20 Y20 vom avea pozitia cnc-ului mutata la pozitia X20 Y20 fara a se executa vreo taiere. Dupa apasarea butonului "Get Position", pozitia curenta va fi modificata si pe interfata, aceasta functionalitate este doar pentru convenientul utilizatorului, pentru a afla pozitia curenta fara alte complicatii.

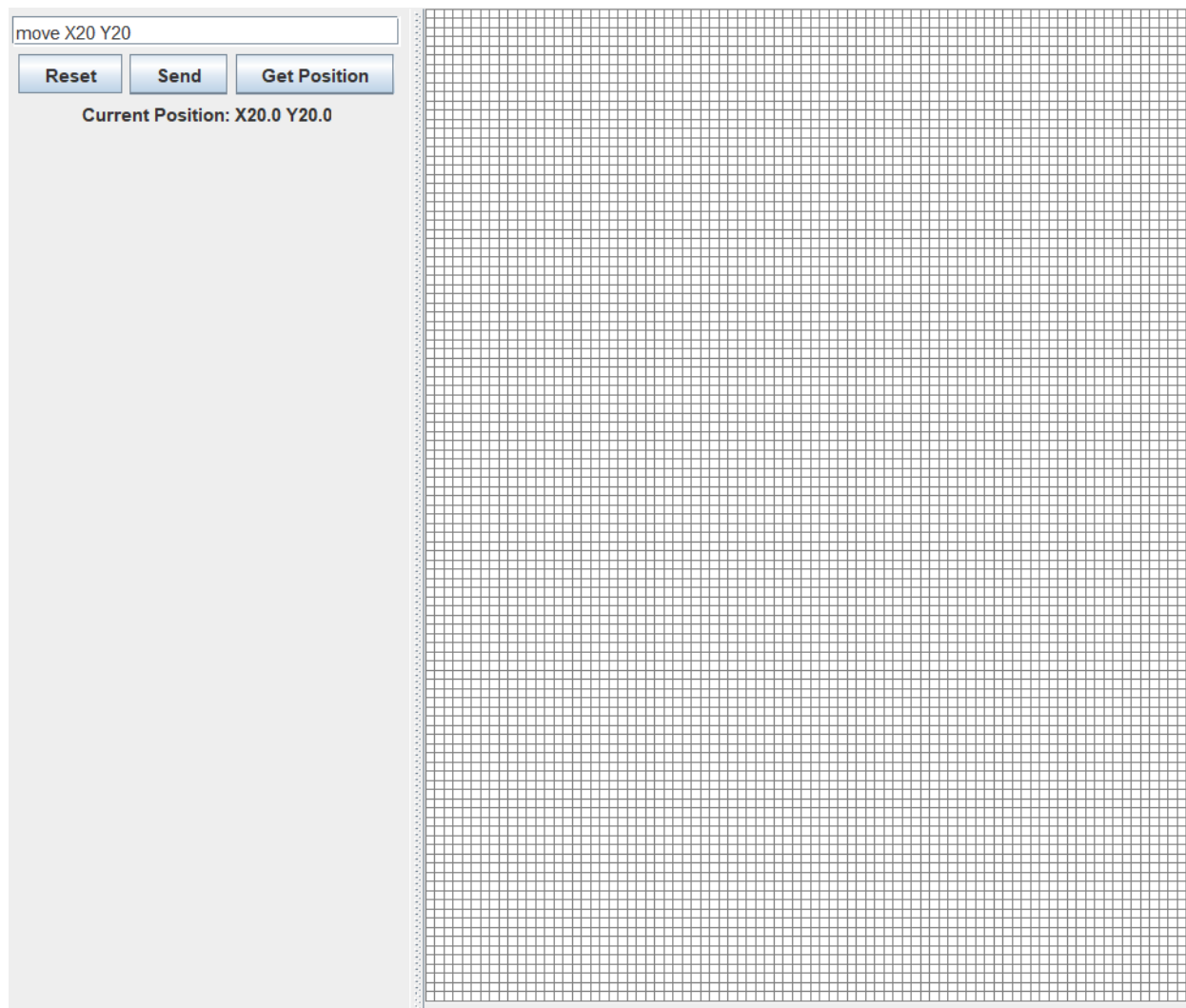


Figura 9 UI dupa apasarea butonului "Send" si "Get Position"

Taiere liniara

Continuand de la pozitia obtinuta mai sus se poate face orice taiere liniara pana la orice pozitie (ex X40 Y60):

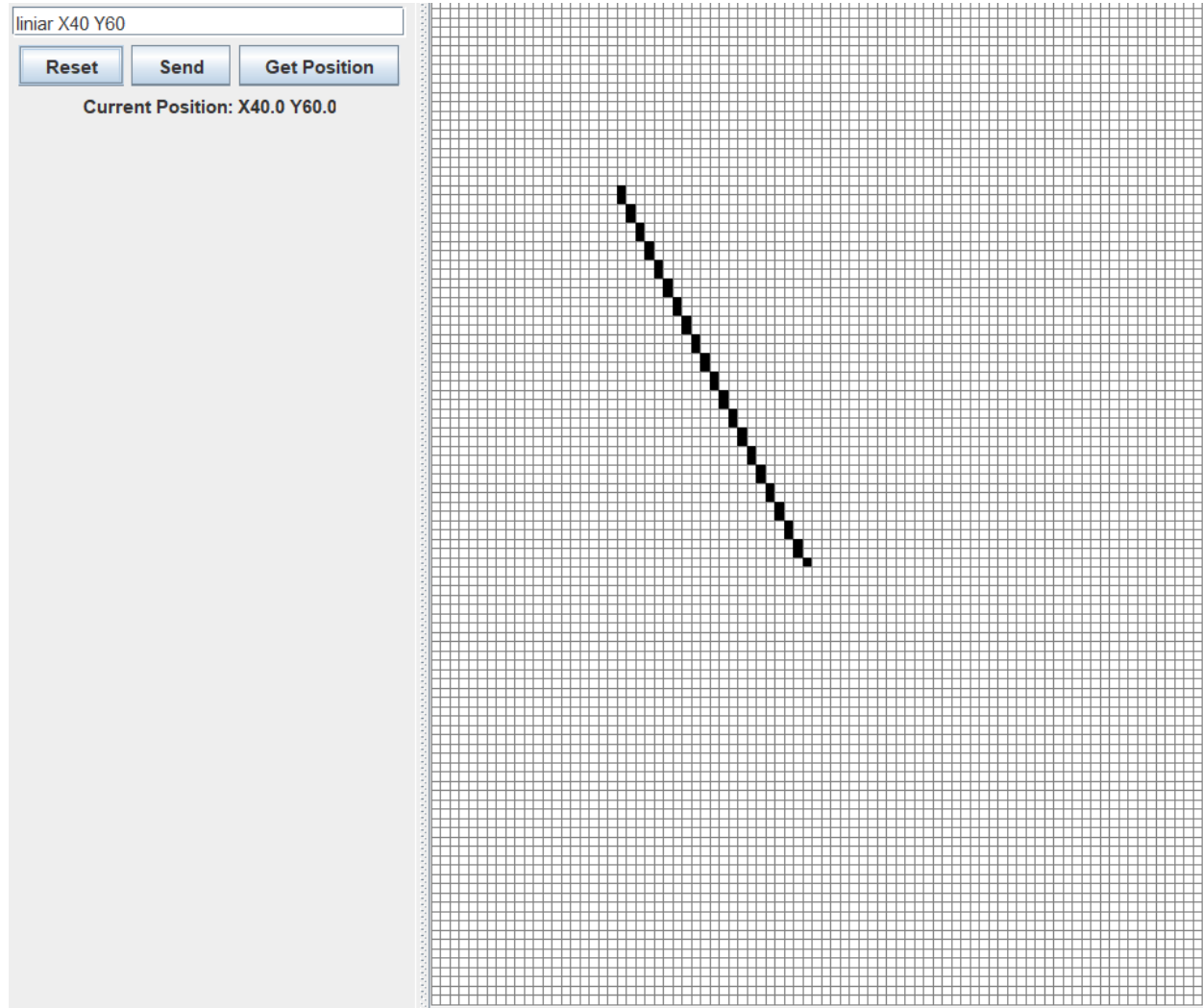


Figura 10 UI dupa trimiterea unei comenzi de taiere liniara si apasarea butonului "Get Position"

Dupa introducerea comenzii in text field s-a apasat pe butonul "Send" si pe "Get Position".

Taiere circulara

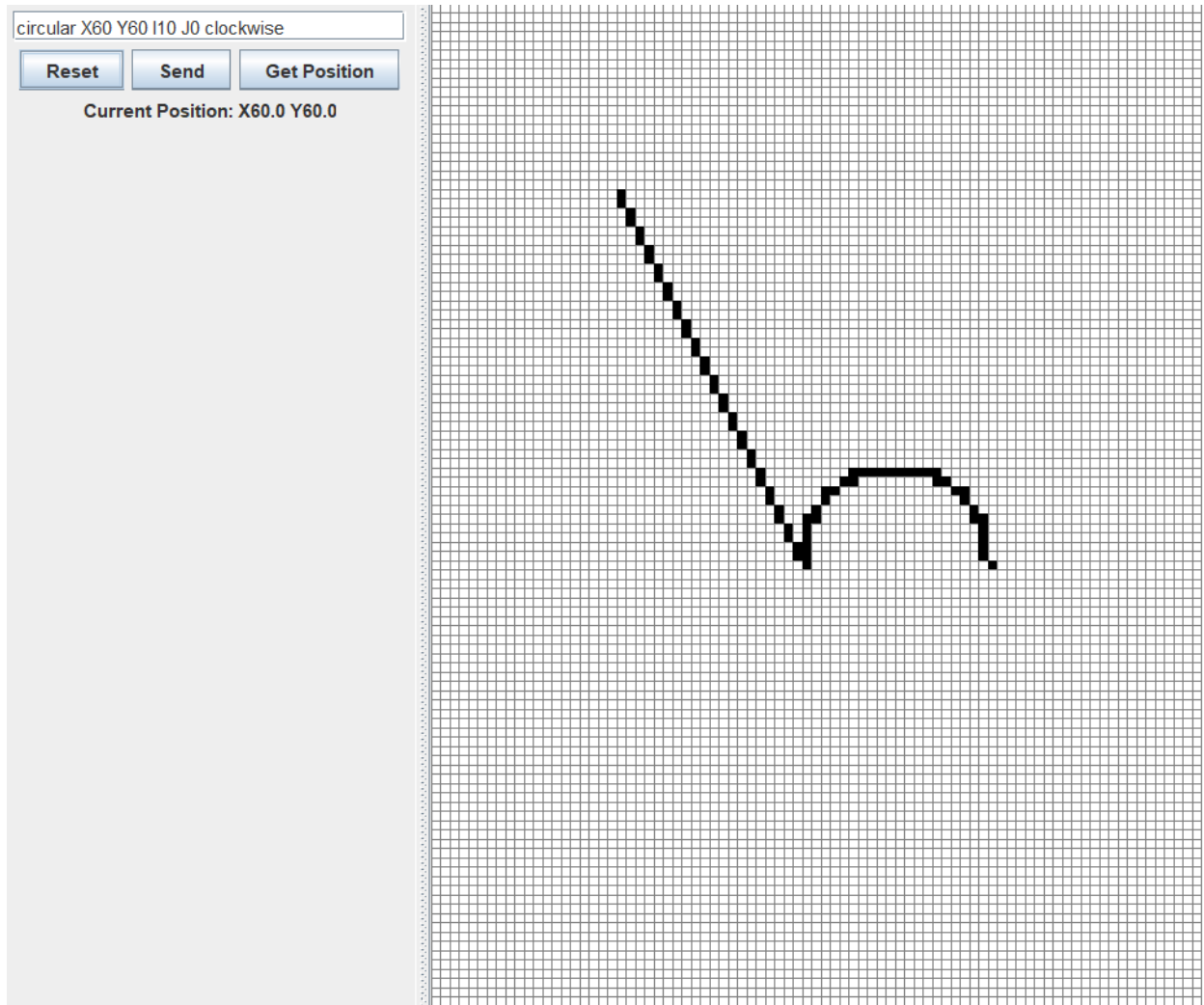


Figura 11 UI dupa trimiterea unei comenzi de taiere circulara clockwise si apasarea butonului "Get Position"

Se trimite comanda de taiere circular cu destinatia in X60 Y60 si centrul cercului la X50 Y60($X=X40+I10$, $Y=Y60+J0$) cu miscare in directia ceasului

Dupa introducerea comenzii in text field s-a apasat pe butonul "Send" si pe "Get Position".

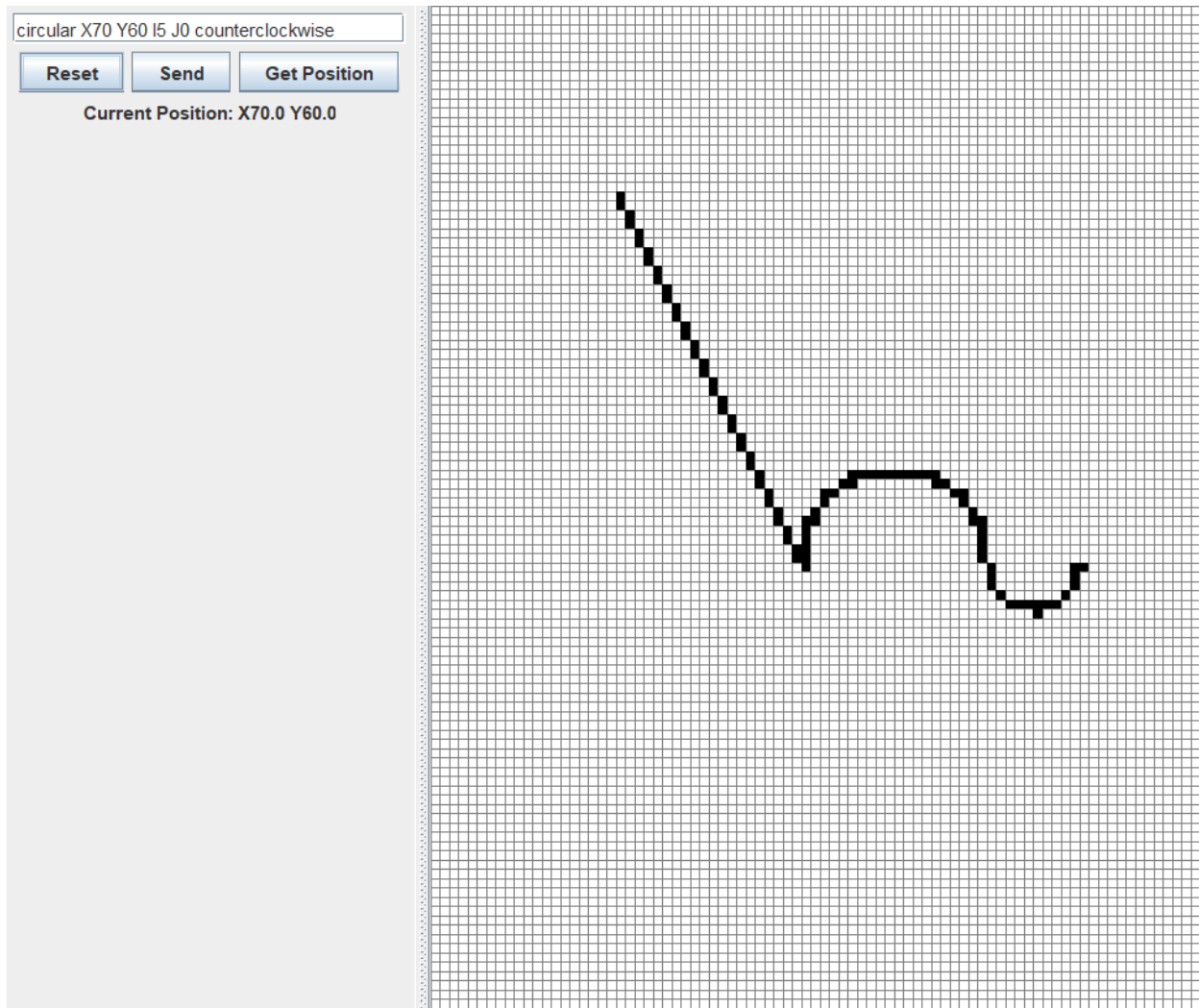


Figura 12 UI dupa trimiterea unei comenzi de taiere circulara counterclockwise si apasarea butonului "Get Position"

Se trimite comanda de taiere circular cu destinatia in X70 Y60 si centrul cercului la X65 Y60($X=X60+I5$, $Y=Y60+J0$) cu miscare in directia opusa ceasului.

Dupa introducerea comenzii in text field s-a apasat pe butonul "Send" si pe "Get Position".

Reset

Dupa apasarea butonului "Reset" se sterge tot si devine pozitia curenta inapoi pe X10 Y10.

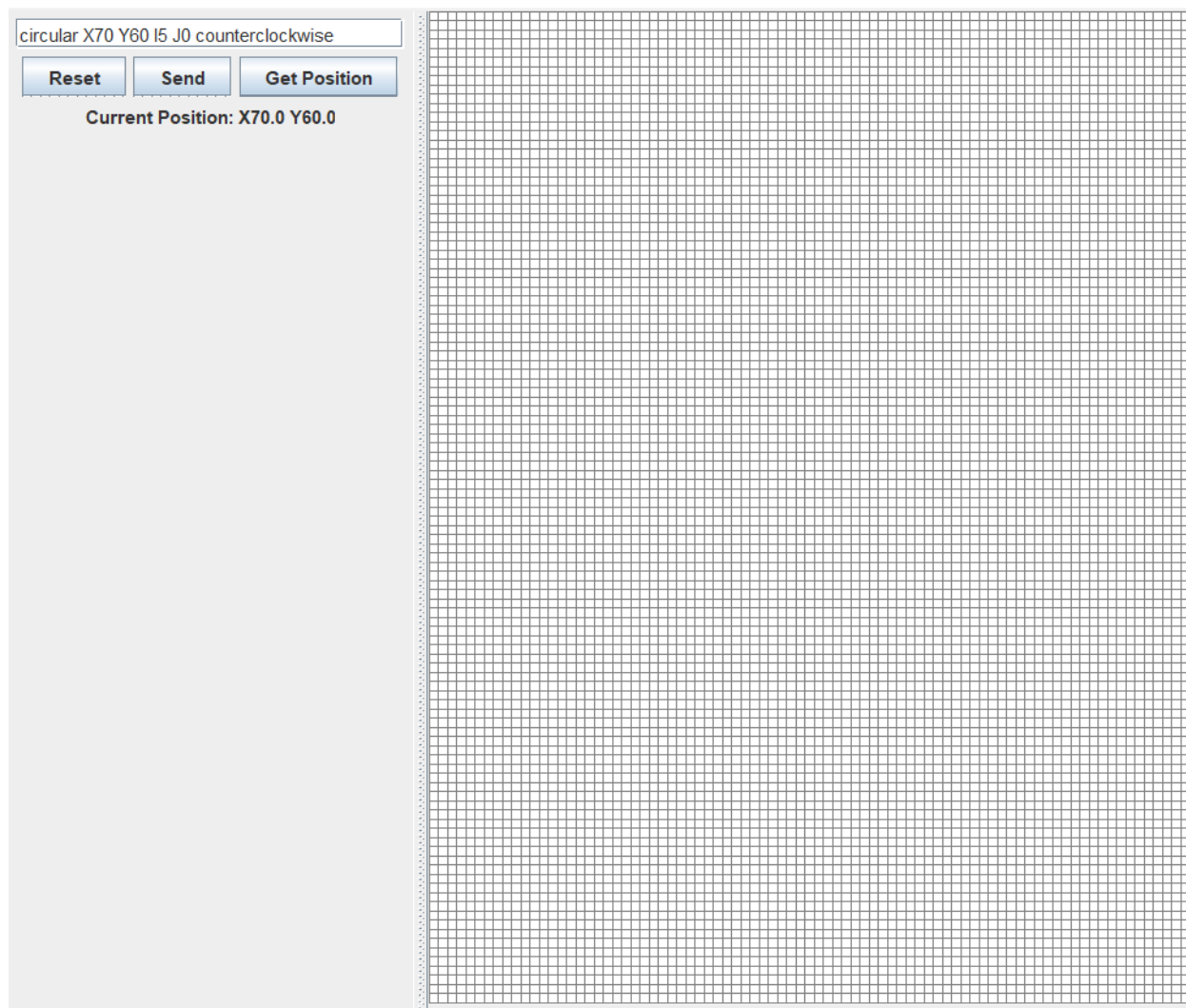


Figura 13 UI dupa apasarea butonului "Reset"

Dupa apasarea butonului "Reset" asa va arata interfata.

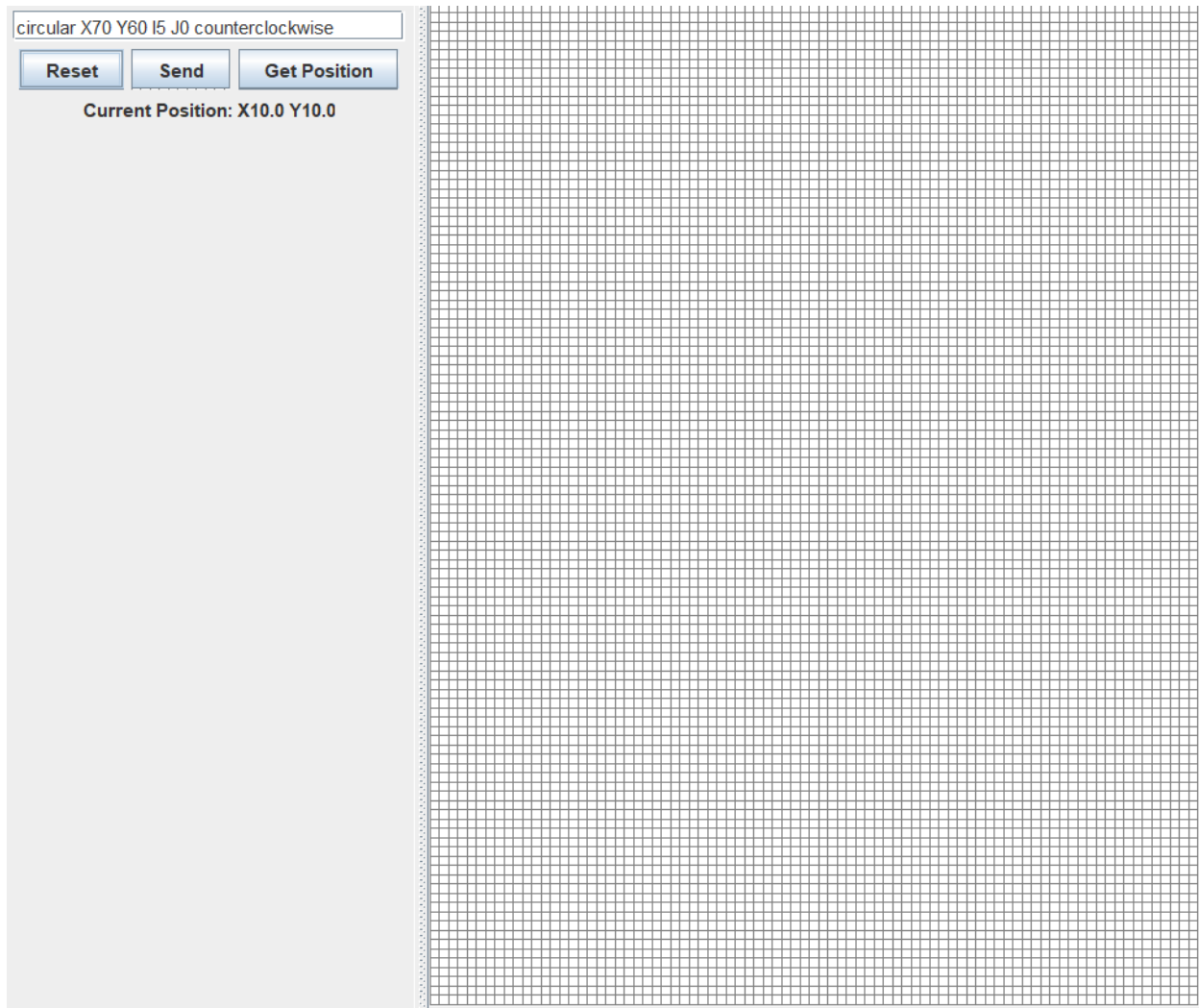


Figura 14 UI dupa apasarea butonului "Reset" si "Get Position"

Dupa apasarea butonului "Get Position" se va afisa pozitia curenta a cnc-ului, pozitia de inceput.

Concluzie

Scopul proiectului a fost de a simula pe calculator taierea efectuata de un utilaj cu comanda numerica.

Am invatat din acest proiect implementarea interpolarii liniare si circulare si reprezentarea unui grid pe o interfata, de asemenea folosirea threadurilor pentru simularea reala a taierii.

Bibliografie

https://ro.wikipedia.org/wiki/Programarea_mașinilor-unelte_cu_comandă_numerică

<https://reprap.org/wiki/G-code#G-commands>

https://www.rasfoiesc.com/educatie/c_himie/Metode-de-interpolare36.php

<https://materiale.pvgazeta.info/revista-34/interpolare-circulara-g02-si-g03.html>

Tabel de figuri

Figura 1 Interfata utilizatorului.....	5
Figura 1 Interfata utilizatorului.....	5
Figura 2 diagrama de functionare	6
Figura 2 diagrama de functionare	6
Figura 3 Cod miscare rapida	7
Figura 4 Cod interpolare liniara	7
Figura 5 Cod interpolare circulara	8
Figura 6 UI dupa introducere in text-field miscarea rapida la X20 Y20	9
Figura 7 UI dupa apasarea butonului "Send" si "Get Position"	10
Figura 8 UI dupa trimiterea unei comenzi de taiere liniara si apasarea butonului "Get Position"	11
Figura 9 UI dupa trimiterea unei comenzi de taiere circulara clockwise si apasarea butonului "Get Position"	12
Figura 10 UI dupa trimiterea unei comenzi de taiere circulara counterclockwise si apasarea butonului "Get Position"	13
Figura 11 UI dupa apasarea butonului "Reset"	14
Figura 12 UI dupa apasarea butonului "Reset" si "Get Position"	15