

Е. А. ГЛІБКО, І. П. ГРЕЧКА, А. П. МІНАКОВ

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ ПРОГРАМУВАННЯ ПРИСТРОЇВ З ЧПК

На підставі теорії комп'ютерного зору для автоматизації та прискорення програмування пристрів з числовим програмним керуванням розроблена програма, що автоматично генерує керуючий код, готовий для використання на фрезерувальному обробляючому комплексі. Розглянуто особливості створення коду керуючої програми для чорнового та чистового фрезерування деталей з подальшим аналізом створених геометричних моделей процесу різання для корегування траєкторії руху інструменту в напрямку висотної координати для запобігання небезпечного зрізу матеріалу в процесі обробки.

Ключові слова: автоматизація програмування, комп'ютерний зір, фрезерування, алгоритм роботи, керуюча програма, карта рельєфу, моделі деталі, колір та яскравість пікселів.

Вступ та аналіз останніх досліджень. Програмне забезпечення верстатів з числовим програмним керуванням (ЧПК) постійно вдосконалюється, але існує зворотній зв'язок між простотою керування та якістю обробки. Тому необхідно знайти компроміс між цими критеріями, що задовольняв би конкретну ситуацію на виробництві. Розроблено декілька варіантів програмного забезпечення для створення коду, що верстат сприймає як керуючу програму [1 – 4]. Але навіть використання подібних продуктів не забезпечує бажану швидкість досягнення кінцевого результату, і постає задача максимально спростити цей процес.

Робота присвячена використанню *комп'ютерного зору* для автоматизації та прискорення програмування пристрів з ЧПК. Ідея полягає в тому, щоб використовувати двомірне зображення для створення коду обробки тривимірної моделі. Третю координату замінено яскравістю окремих фрагментів зображення.

Мета роботи та постановка проблеми. Автоматизація розробки повноцінного керуючого коду з використанням теорії комп'ютерного зору для фрезерування на верстаті з ЧПК на підставі *растрової карти рельєфу моделі*. Кінцевий програмний продукт має відповідати наступним вимогам та задовольняти такі потреби: достатня точність обробки, висока швидкість обчислення траєкторій, оптимальна траєкторія руху ріжучого інструменту, відповідність режимів різання до нормативних, можливість редагування необхідних параметрів, а саме: вибір інструменту, вибір режимів різання, редагування параметрів існуючих та додавання нових інструментів та ін., простота використання програми.

Автоматизація програмування пристрою з ЧПК. Обробляючий фрезерувальний комплекс має координатну систему, що містить 5 осей. Точність позиціонування досягає 0,001 мм. Це дає можливість обробляти поверхні будь-якої складності на високій швидкості із максимальною точністю. Але процес програмування ЧПК досить складний, трудомісткий та відповідальний. Найменша помилка із боку програміста може призвести до псування коштовного інструменту та заготовки.

Розроблено декілька варіантів програмного забезпечення для створення коду, що верстат сприймає як керуючу програму. При користуванні ними створюється тривимірна модель необхідної конфігурації та на базі параметричного програмування задається траєкторія руху ріжучого інструменту і параметри різання. Постає задача максимально спростити цей процес. В основу роботи закладені положення теорії комп'ютерного зору.

Метод, що було запропоновано, дозволяє за лічені хвилини створити код обробки для фрезерування за умови існування карти рельєфу моделі. Карта рельєфу – це зображення виду моделі в напрямку нормалі до горизонтальної площини. Програму було написано мовою Pascal в середовищі Delphi 2009.

Функції програми. Алгоритм роботи. Було заплановано такі функціональні елементи програми:

- обробник графічного матеріалу;
- блок проектування чорнової обробки;
- блок проектування чистової обробки;
- база даних інструментів;
- інструмент для редагування інструментів.

Роботу над програмним додатком розпочато із створення блок-схеми, що ілюструє алгоритм роботи із програмою (рис. 1).

Програму можна умовно поділити на дві частини: редактор інструментів і генератор керуючого коду. Початок роботи полягає в підготовці графічного матеріалу та ініціалізації геометричних даних деталі. Окремими частинами створюється код для різних режимів обробки. Параметри різання обираються залежно від параметрів обраного інструменту. Дані про інструмент зберігаються в базі даних інструменту. Для повноцінного користування програмою необхідно мати можливість створювати новий інструмент, редагувати існуючий та видаляти непотрібний. Такий доступ до бази даних реалізовано окремим елементом програми.

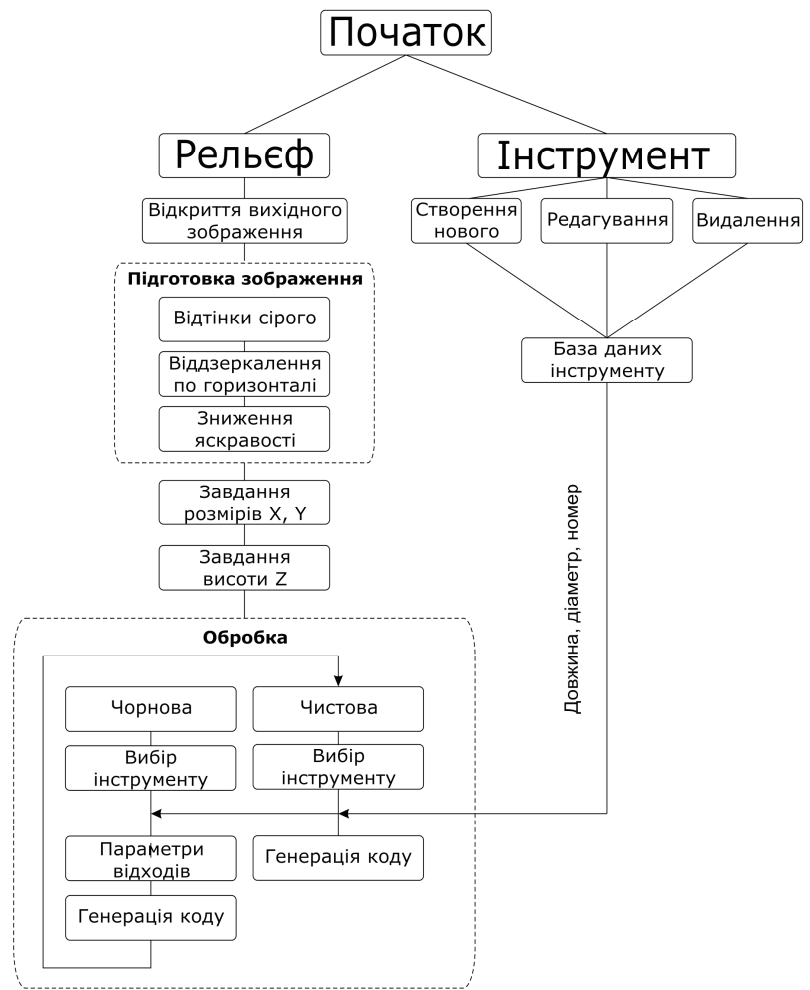


Рис. 1 – Алгоритм роботи програми.

База даних інструменту. Редагування інструменту. Процес створення керуючої програми включає використання даних про інструмент, які зберігаються в базі даних інструменту, що містить набір файлів, кожен з яких описує один із інструментів, та файл-ключ, з переліком назв усіх інструментів. Опис кожного інструменту включає в себе значення діаметру фрези, довжини її ріжучої кромки та номер інструменту. Якщо в базу даних вносяться зміни із використанням засобів програми, що описується в даному проекті, зміст файлу-ключа змінюється автоматично. Таким чином користувач завжди має актуальну інформацію про доступний інструмент. Нижче наведено реалізацію процесу редагування параметрів інструменту. Доступ до редагування бази даних інструменту відбувається через головне меню програми.

Завантаження карти рельєфу. Підготовка рельєфу. Відкриття робочого зображення здійснюється із головного меню програми (рис. 2). Перед використанням завантаженого зображення необхідно його підготувати: перетворити зображення у гаму відтінків сірого кольору; змінити напрямок осей координат, оскільки в середовищі Delphi використовується ліва система координат (вісь X – вправо, вісь Y – вниз), а верстат із ЧПК використовує праву систему коор-

динат; встановити потрібний рівень яскравості. Це відбувається при натисканні кнопки «Подготувати зображення». Для змінення кольорової гами відбувається приведення до середнього арифметичного значення яскравості кожного з трьох кольорів – червоного, зеленого та синього. Отримане значення надається кожному з трьох кольорів відповідного пікселю в контейнері, що містить робоче зображення. В циклі перебирається кожен піксель вихідного зображення (рис. 3).

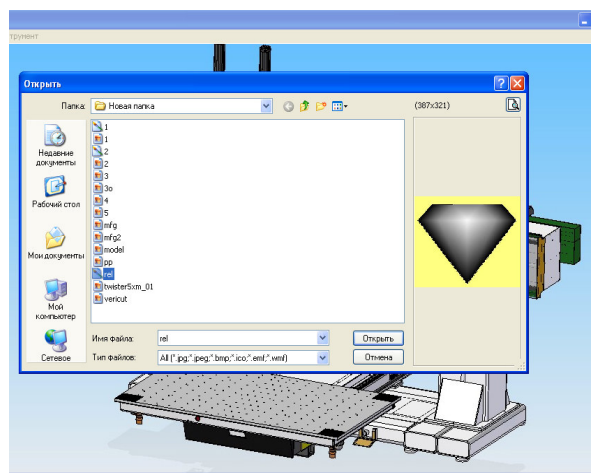


Рис. 2 – Завантаження зображення.

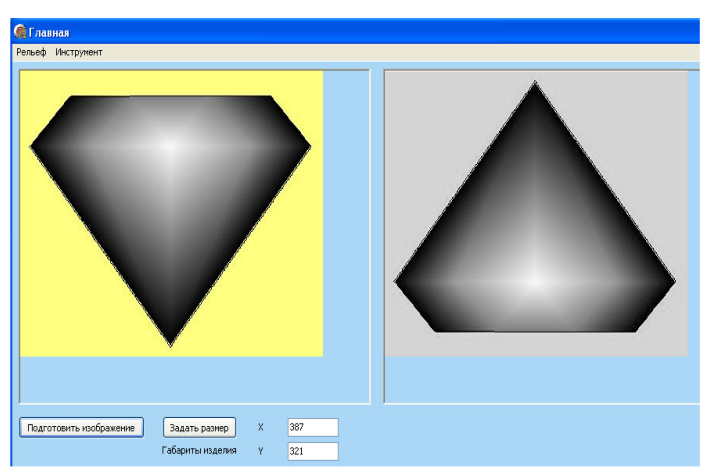


Рис. 3 – Підготовлене зображення.

Далі визначаються значення яскравості кожного з трьох кольорів окремо:
 $r := \text{GetRValue}(\text{col})$; $g := \text{GetGValue}(\text{col})$; $b := \text{GetBValue}(\text{col})$.

Отримуються значення яскравості для кожного кольору робочого пікселю $\text{Ch_B_col} := (\text{r} + \text{g} + \text{b}) \text{div } 3$.

Для зниження рівня яскравості зображення до мінімально можливого необхідно визначити існуючий міні-

мальний рівень яскравості:

```
min_color :=255; ... if Ch_B_col<min_color then min_color :=Ch_B_col ;
```

Готове зображення формується наступним чином (рис. 3):

```
Image2.Canvas.Pixels[i,Image2.Height-j]:=TColor(RGB(Ch_B_col-min_color, Ch_B_col-min_color, Ch_B_col-min_color)); end; end;
```

Фрагмент [i,Image2.Height-j] відповідає за заповнення підготовленого зображення в такому порядку, при якому отримуємо зображення, віддзеркалене по горизонталі відносно вихідного.

Кожному пікселю надається значення кольорів $Ch_B_col - min_color$, де min_color – це максимальна різниця, на яку можна зменшити рівень яскравості зображення. Після перетворення зображення на формі відображується група елементів, що дозволяють налаштувати геометричні параметри моделі.

За умовчанням розміри моделі приймаються за рівністю 1 піксель зображення дорівнює 1 мм моделі. Але на практиці є необхідність змінювати значення лінійних розмірів готової моделі за умови збереження її пропорцій. Це реалізовано наступним чином: нові значення розмірів моделі отримуються із полів Edit1 та Edit2. При зміні їх значень, нові значення зберігаються, після натискання кнопки «Задать размер». Без цього контролююча змінна «size» матиме значення «false». Такі міри захисту гарантують відповідність розмірів, що використовуються при обчисленні, до розмірів, які необхідні користувачеві.

Далі стає активним поле для завдання висоти деталі, або глибини обробки. Необхідно задати мінімальне і максимальне значення глибини відносно нуля (Z_{min} та Z_{max}). Z_{min} – це мінімальна висота обробки, нижче якої фреза не опуститься, відповідає рівню яскравості пікселю «0». Z_{max} – це максимальна висота обробки, відповідає рівню яскравості пікселю «255».

Реалізовано захист від ситуації, коли значення мінімальної та максимальної глибини завдані невірно. Якщо значення висот задані вірно, відбувається обчислення коефіцієнту вертикального переміщення фрези залежно від зміни рівня яскравості пікселю. За умовчанням розміри деталі приймаються за пропорцією: одиниця рівня яскравості пікселю зображення дорівнює 1 мм висоти моделі. На практиці є необхідність змінювати значення висоти готової моделі за умовою зберігання пропорційності змін рівня висот до змін рівня яскравості зображення, яка реалізована у програмі.

Програмування обробки. Форма «Обработка» містить елементи, що надають можливість перейти до проектування як чорнової, так і чистової обробки, попереджаючи про необхідність зберігати послідовність технологічного процесу. Специфічність роботи із графічним матеріалом полягає в тому, що найменший елемент, який аналізується – це один піксель. При змінених значеннях ширини і довжини, зображення має бути змінене відповідно до коефіцієнту зміни лінійних розмірів відносно вихідного значення (K_x). З урахуванням нових розмірів відбувається заповнення робочого зображення. Значення яскравості пікселю на полі головної форми використовується для призначення яскравості пікселям на формі «Обработка» в режимі циклу.

Форма «Обработка» містить елемент, який представляє собою перелік назв інструментів, що містяться в відповідній базі даних. Перелік інструментів розміщено на цій формі з метою прискорити доступ до програмування обробки.

Керуюча програма містить три складові частини: *ініціалізація*, *основна частина* та *завершення*. Для усіх керуючих програм чистової обробки *ініціалізація* відрізняється лише параметрами використаних інструментів, а *завершення* виглядає однаково. Тому завантажування заздалегідь підготовлених вступної та завершальної частини програми в код обробки здійснено автоматично.

Найважливішою частиною керуючого коду є фрагмент, що несе інформацію про переміщення ріжучого інструменту під час обробки. Кожний кадр програми містить координати наступної точки простору, в якій має прийти фреза. Координати X та Y отримуються автоматично під час сканування вихідного зображення. Координата Z , що відповідає за положення фрези по висоті, вираховується, виходячи із рівня яскравості кожного пікселю зображення. Під час обробки фреза пересувається, проходячи від нуля координат до максимального значення по осі X . Після цього інструмент повертається в нуль по осі X , положення по осі Y змінюється на значення ширини одного пікселю, і фреза знову переміщується до максимального значення по осі X . Таким чином, інструмент переміщується до максимального значення по осях X та Y (рис. 4).

На рис. 4 синіми лініями позначено рух інструменту на підході та на відводі інструменту, зеленим кольором – рух фрези під час різання матеріалу, червоним – рух при поверненні інструменту в нуль по осі X на безпечній висоті відходів. Для розрахунку висоти використовується рівень яскравості пікселю на робочому зображенні. Оскільки при підготовці зображення до обчислень відбулося прирівнювання рівнів кожного з основних кольорів, достатньо визначити рівень яскравості одного з каналів:

```
col :=Image1.Canvas.Pixels[X_local, Yo]; Z :=GetRValue(col)*unit1.Form1.Kz.
```

Для збереження пропорційності змін рівня висот до змін рівня яскравості зображення використовується коефіцієнт K_z , що було обчислено при роботі із головною формою. Таким чином, Z – це висота пікселя, який оброблюється на даний момент.

Оскільки при написанні керуючого коду для обробки на фрезерувальному верстаті завжди вказуються координати кінцевої точки інструменту, а для розрахунків використовуються координати центру сфери фрези, необхідно внести поправку висоти, що визначається радіусом фрези:

$$Z_o := Z + r.$$

На практиці дуже часто сусідні від робочого пікселі мають висоту більшу, за щойно обчислену. Тоді інструмент, приходячи на висоту робочого пікселю, може зрізати матеріал сусідніх пікселів. Для отримання найкращих результатів в подібних ситуаціях необхідно врахувати висоту, на яку буде достатньо підняти фрезу, щоб запобігти небажаному зрізання матеріалу.

Для цього необхідно перевірити висоти кожного пікселю, який може бути оброблений при даних координатах центру фрези. Для прикладу розглянемо ситуацію, коли фреза має діаметр 4мм. Необхідно перевірити всі пікселі, що лежать хоча б частково в колі діаметром 4мм (4 пікселя) (рис. 5).

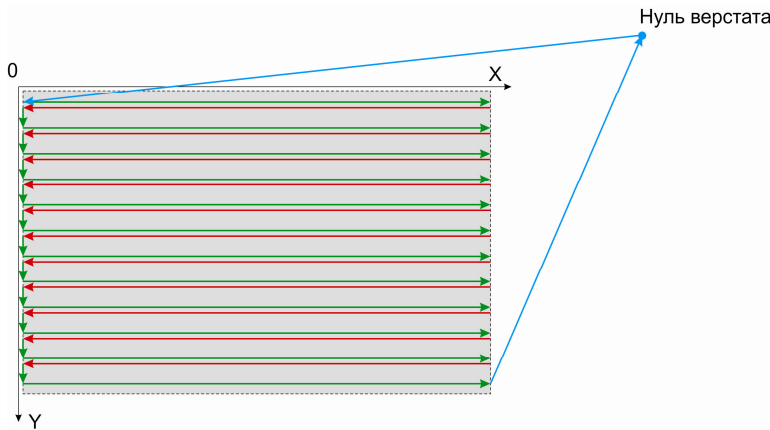


Рис. 4 – Схема руху інструменту при чистовій обробці.

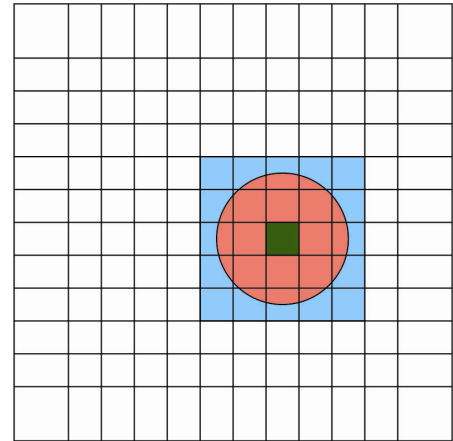


Рис. 5 – Виявлення можливих зарізів.

Зеленим кольором позначено робочий піксель, червоним – діаметр фрези, блакитним – пікселі, що перевіряються. Розглянемо область, яку покриває діаметр фрези (плюс-мінус значення радіусу фрези від координат робочого пікселю):

for Yt := Yo-R to Yo+R do begin ; for Xt := X_local-R to X_local+R do begin.

В цій області необхідно перевірити лише пікселі, що знаходяться «всередині» діаметру фрези. Для виявлення таких пікселів використано рівняння кола $R^2 = X^2 + Y^2$. Якщо пікселі знаходяться в середині кола, що розглядається, буде виконуватись нерівність $R^2 > X^2 + Y^2$ (рис. 6). Проаналізуємо кожен піксель в зазначеній області:

col := Image1.Canvas.Pixels[Xt, Yt] ; Zt := GetRValue(col)*unit1.Form1.Kz.

Якщо висота точки, що розглядається, не перевищує безпечну, значення висоти з урахуванням поправки радіусу буде використано як остаточну координату Z для даної точки: $Z_o := Z + r$. Таким чином, стає відома висота точки, що розглядається в даний момент циклу. Необхідно перевірити чи буде знаходитись ця точка в середині сфери, що описує фрезу, коли найнижча точка поверхні фрези буде знаходитись на висоті, визначеній для робочого пікселя. Для цього використано рівняння сфери $R^2 = X^2 + Y^2 + Z^2$. Якщо пікселі знаходяться всередині сфери, що розглядається, буде виконуватись нерівність $R^2 > X^2 + Y^2 + Z^2$. Проаналізуємо кожен піксель в зазначеній області:

if (R>sqrt(sqrt(Xo-Xt)+sqrt(Yo-Yt)+sqrt(Zo-Zt)*unit1.Form1.Kz))) then begin.

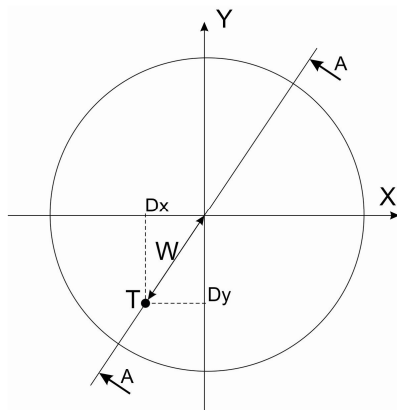


Рис. 6 – Аналіз точки. Горизонтальна проекція.

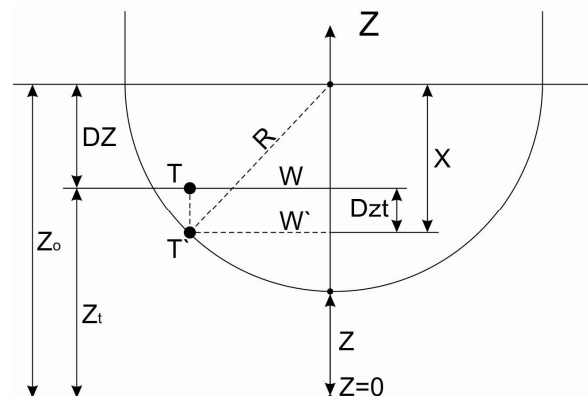


Рис. 7 – Аналіз точки. Фронтальна проекція.

У випадку, коли розглянутий піксель дійсно знаходиться в середині сфери фрези, необхідно визначити висоту, на яку необхідно підняти фрезу для запобігання зарізу. Для цього визначається відстань від точки до осі інструменту W (рис. 7):

$$W = \sqrt{Dx^2 + Dy^2} ;$$

тобто $Dx := \text{ABS}((X_o - X_t))$; $Dy := \text{ABS}((Y_o - Y_t))$; $w := \text{sqrt}(\text{sqrt}(Dx) + \text{sqrt}(Dy))$.

Для зручності на рис. 7 зображено фронтальну проекцію, яка показує відстань від точки до осі інструмента та її висоту в натуральній величині. Висота, на яку необхідно підняти фрезу, позначена Dzt . При підйомі інструменту на зазначену висоту точка, що розглядається, буде розташована на поверхні сфери фрези. Таким чином, стає відомою відстань від центру сфери до точки – це радіус сфери. В даному випадку це 4 мм. Також відома відстань від точки до осі Z – це $W' = W$. Таким чином є можливість вирахувати відстань від центру фрези по осі Z до нової координати точки:

$$X = \sqrt{R^2 - W^2} ;$$

тобто $X := \text{sqrt}(\text{sqrt}(R) - \text{sqrt}(W))$.

Для подальших розрахунків необхідно визначити різницю між висотою центру фрези та висотою точки, що розглядається:

$$DZ = \sqrt{Z_0 - Z_t} .$$

З рис. 7 видно, що необхідна різниця висот Dzt – це різниця між Dz та відстанню X :

$$DZ_t = X - DZ .$$

Таким чином, отримано формулу для різниці висот, на яку необхідно підняти центр фрези для запобігання зарізу матеріалу. Для зручності візуалізації на вихідному зображенні пікселі, в яких довелося змінювати положення фрези з урахуванням сусідніх пікселів, позначені червоним кольором:

$Z_o := Z_o + DZ_t$; `image1.Canvas.Pixels[Xo, Yo] := c1Red`;

Для безпеки обробки необхідно передбачити підйом фрези на необхідну висоту при переході між етапами циклу, тобто при доходженні фрези до максимальних значень координат по осях X та Y і при проходженні по мінімальним значенням цих координат:

`if X_local=image1.Width then Zo := form1.Zmax+25`; `if X_local=0 then Zo := form1.Zmax+25`;

Отримані координати відразу записуються в файл керуючого коду. Враховується поправка на радіус фрези для координати Z – тепер необхідно відняти цю відстань для отримання висоти знаходження кінця фрези:

`writeln (chist, 'N ' + intToStr(n) + ' X ' + intToStr(x_local) + ' Y ' + intToStr(Yo) + ' Z ' + StringReplace(floatToStrf(Zo - r, ffFixed, 5, 2), ',', ', ', [rfReplaceAll, rfIgnoreCase]))`; `n := n + 1`;

Описаний процес виконується в циклі для кожного пікселя вихідного зображення. Швидкість обчислення рівня висоти фрези для кожного пікселя залежить від розміру оброблюваної поверхні та діаметру обраної фрези.

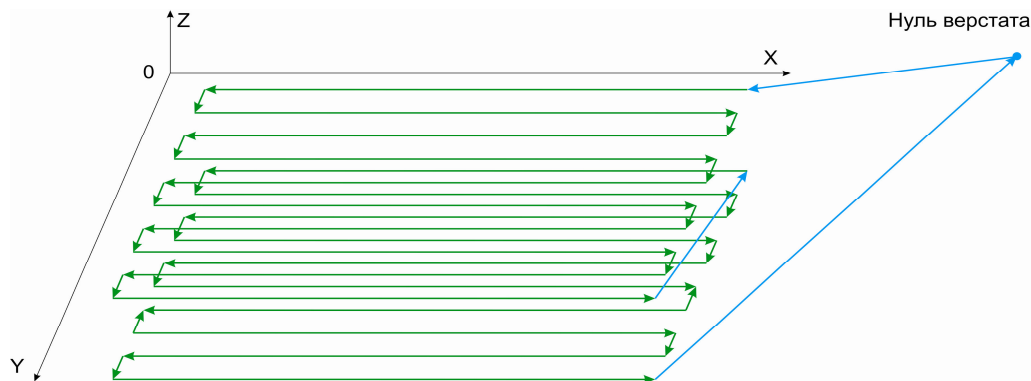


Рис. 8 – Схема руху інструменту при чорновій обробці.

По завершенню циклу керуючий код містить координати всіх точок, які має пройти фреза при обробці. Отриманий в процесі програмування керуючий код записано до файлу, що знаходиться в основному каталозі програми. Код готовий до використання в якості програми чистової обробки на системі ЧПК.

Мета чорнової обробки – зняти та видалити основну частину матеріалу заготовки, що не утворює тіло деталі. На поверхні деталі залишається невелика кількість матеріалу, що видаляється при чистовій обробці. Аналогічно до чистової обробки реалізовано програмування чорнового фрезерування.

Для чорнової обробки основними параметрами є відстань між проходами фрези на горизонталі та глибина, на яку фреза переміщується для обробки наступного шару матеріалу. Також важливим параметром є рівень висоти відходу фрези, значення якої завдається на формі обробки. Під час обробки фреза пересувається, проходячи від нуля координат до максимального значення по осі X . Після цього положення по осі Y змінюється на значення ширини, що міститься в змінній `step_over`, і фреза переміщується по осі X до мінімального значення. Таким чином, інструмент переміщується до максимального значення по осях X та Y (рис. 8). Переміщення по осі Z реалізовано за наступним принципом: матеріал видаляється шарами з товщиною, значення якої міститься

в змінній `step_depth`. При попаданні на матеріал, що належить до тіла деталі, фреза піднімається на попередній рівень, і рух продовжується по осі X на цьому рівні до тих пір, доки фреза не мине матеріал тіла деталі.

Для нумерації рядків керуючого коду використано *автоінкрементне зростання* змінної n . По завершенню запису заключної частини програми закриваються вихідні файли та обнулюються використані змінні. Швидкість обчислення рівня висоти фрези для кожного пікселя залежить від розміру оброблюваної поверхні та діаметру обраної фрези. Отриманий в процесі програмування керуючий код записано до файлу, що знаходиться в основному каталозі програми. Код готовий до використання в якості програми чорнової обробки на системі ЧПК.

Проведення натурного експерименту. Кінцевим етапом роботи над програмним забезпеченням було проведення випробування дії програми на практиці. Було створено карту рельєфу, яка містить прямолінійні і криволінійні поверхні (рис. 9). Габаритні розміри деталі 240x160x120 мм.

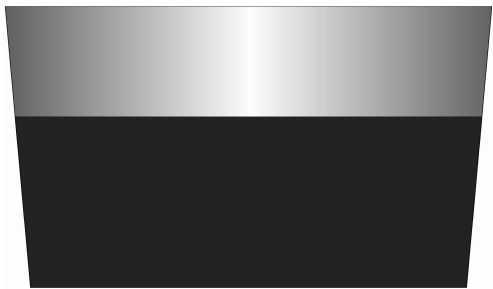


Рис. 9 – Карта рельєфу для натурного експерименту. для надання виробу товарного вигляду (рис. 11).

Відхилення габаритних розмірів деталі від очікуваних склали 0,5 мм, що вважається припустимим.

Висновки. В результаті виконання поставленої задачі була отримана програма, яка автоматично генерує керуючий код, готовий для використання на фрезерувальному обробляючому комплексі. Особливо розглянуто особливості створення коду керуючої програми для чорнового та чистового фрезерування деталей.

Шляхом аналізу створених геометричних моделей процесу різання чорнового та чистового торцевого фрезерування було скореговано траєкторію руху інструменту в напрямку висотної координати для запобігання небажаного зрізу матеріалу в процесі обробки. Робота програми займає набагато менше часу порівняно із часом роботи програміста для створення рівноцінного коду засобами програм аналогів Pro/Engineer чи ArtCam. При цьому точність обробки зменшена, що обумовлено якістю зображення на карті рельєфу. При збільшенні роздільної здатності карти помітно уповільнюється робота програми. Але якість обробки є достатньою для використання розробленого додатку при обробці невідповідальних в промисловому сенсі поверхонь.



Рис. 10 – Готова деталь.



Рис. 11 – Поверхня деталі.

Результати роботи (керуюча програма) було апробовано на практиці та планується до впровадження в технологічний процес обробки деталей меблевих виробів.

Список літератури: 1. ЧПУ серии 10. Руководство по программированию. – Италия, Турин: Компания OSAI, – 2003. – 579 с. 2. Mineev M.A. Введение в Pro/Engineer 3.0. – 2002. – 421 с. 3. Mantek Chia. Базовый курс Pro/Engineer. – 2001. – 84 с. 4. ArtCam Фрезерование дерева. Брошюра / Delkam «ArtCam Русская версия 5.5. – Канада, Онтарио: Delkam, – 2007. – 14 с.

Bibliography (transliterated): 1. ChPU serii 10. Rukovodstvo po programirovaniju. Ytalija, Turin: Kompanija OSAI, 2003. Print. 2. Mineev, M. A. Vvedenie v Pro/Engineer 3.0. 2002. Print. 3. Mantek, Chia. Bazovij kurs Pro/Engineer. 2001. Print. 4. ArtCam Frezerovanie dereva. Broshjura. Art-Cam Russkaja versija 5.5. Kanada, Ontario: Delkam, 2007. Print.

Глібо Олена Анатоліївна – кандидат технічних наук, доцент, доцент кафедри геометричного моделювання та комп'ютерної графіки, Національний технічний університет «Харківський політехнічний інститут», м. Харків; тел.: (057) 707-64-31; e-mail: e_glibko@mail.ru.

Глибо Елена Анатольевна – кандидат технических наук, доцент, доцент кафедры геометрического моделирования и компьютерной графики, Национальный технический университет «Харьковский политехнический институт», г. Харьков; тел.: (057) 707-64-31; e-mail: e_glibko@mail.ru.

Glibko Elena Anatolievna – Candidate of Technical Sciences, Associate Professor, Chair of Geometrical Modeling and Computer Graphics, National Technical University "Kharkiv Polytechnic Institute", Kharkov; tel.: (057) 707-64-31; e-mail: e_glibko@mail.ru.

Гречка Ірина Павлівна – кандидат технічних наук, доцент кафедри теорії і систем автоматизованого проектування механізмів і машин, Національний технічний університет «Харківський політехнічний інститут», м. Харків; тел. (057) 707-69-01; e-mail: Iri_@mail.ru.

Гречка Ирина Павловна – кандидат технических наук, доцент кафедры теории и систем автоматизированного проектирования механизмов и машин, Национальный технический университет «Харьковский политехнический институт», г. Харьков; тел.: (057) 707-69-01; e-mail: Iri_@mail.ru.

Grechka Iryna Pavlivna – Candidate of Technical Sciences, Associate Professor, Chair of Theory and Computer Aided Design of Mechanisms and Machines, National Technical University "Kharkiv Polytechnic Institute", Kharkov; tel.: (057) 707-69-01; e-mail: Iri_@mail.ru.

Мінаков Андрій Петрович – студент механіко-технологічного факультету кафедри геометричного моделювання та комп'ютерної графіки, Національний технічний університет «Харківський політехнічний інститут», м. Харків; тел.: (057) 707-64-31.

Минаков Андрей Петрович – студент механико-технологического факультета кафедры геометрического моделирования и компьютерной графики, Национальный технический университет «Харьковский политехнический институт», г. Харьков; тел. (057) 707-64-31.

Minakov Andrei Petrovich – student, Chair of Geometrical Modeling and Computer Graphics, Department of Mechanics and Technology, National Technical University "Kharkiv Polytechnic Institute", Kharkov; tel.: (057) 707-64-31.

УДК 517.968.519.6

С. В. ДУХОПЕЛЬНИКОВ

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДИФРАКЦИИ ТМ-ВОЛНЫ НА ЗАПОЛНЕННОЙ ДИЭЛЕКТРИКОМ ЦИЛИНДРИЧЕСКОЙ АНТЕННЕ КОЛЬЦЕВОГО СЕЧЕНИЯ С ПРОДОЛЬНЫМИ ЩЕЛЯМИ

Дан вывод сингулярных интегральных уравнений задачи дифракции ТМ-волны на цилиндрической кольцевой антенне, заполненной диэлектриком, внутренняя поверхность которой идеально проводящая, а идеально проводящая внешняя поверхность имеет продольные щели. Дискретная математическая модель построена методом дискретных особенностей и проведен численный эксперимент на ее основе. Построены поперечник полного рассеяния, поля в ближней зоне и диаграммы направленности.

Ключевые слова: метод дискретных особенностей, сингулярный интеграл, антенна.

Введение. В работе рассматривается задача дифракции плоской монохроматической волны на идеально проводящей цилиндрической поверхности, покрытой диэлектрическим цилиндром, на внешней поверхности которого – идеально проводящая цилиндрическая поверхность со щелями.

Задача падения плоской Е-поляризованной волны (ТМ-волны) на «кольцевой волновод» с одной продольной щелью во внешнем цилиндре, находящемся в вакууме, решались *методом парных рядов* [1 – 2], а при отсутствии внутреннего цилиндра – *методами конечных элементов, методом задачи Римана-Гильберта* [3 – 5], однако они реализовывались только для одной продольной щели. Применяемый подход ранее использовался для решения задачи дифракции плоской монохроматической волны и цилиндрической волны на круговом цилиндре с конечным числом продольных щелей, находящемся в однородной среде (предполагался вакуум) [6].

Рассматриваемые в данной работе задачи представляют значительный практический интерес, в связи с необходимостью рассчитывать соответствующие антенные устройства. Эта задача теории дифракции волн приводит к краевой задаче для *уравнений Максвелла*. Эти задачи для уравнений Максвелла служат математическими моделями цилиндрических электродинамических структур, и нашли широкое применение при проектировании и создании апертурных и поверхностных антенн, открытых резонаторов, щелевых волноводов и фильтров на их основе.

Целью работы является создание математической модели для данных структур и анализ некоторых физических характеристик, полученных путем численных вычислений.