

СОКОЛАН ЮЛІЯ

Хмельницький національний університет
<https://orcid.org/0000-0002-0273-5719>
e-mail: sokolan.julia@gmail.com

ЛАРІОНОВ ІВАН

Хмельницький національний університет
e-mail: Larionov674@gmail.com

МАЙДАН ПАВЛО

Хмельницький національний університет
<https://orcid.org/0000-0003-3319-8730>
e-mail: maidanps@gmail.com

СОКОЛАН КАТЕРИНА

Хмельницький національний університет
<https://orcid.org/0000-0002-3513-8312>
e-mail: sokolan.kateryna@gmail.com

АВТОМАТИЗОВАНЕ ГЕНЕРУВАННЯ КОДІВ ОБРОБКИ ДЕТАЛЕЙ НА ВЕРСТАТАХ З ЧИСЛОВИМ ПРОГРАМНИМ КЕРУВАННЯМ

В сучасних реаліях процес генерації кодів для верстатів з ЧПК проводиться вручну, або ж оператор верстату пише код для САД-моделі, яка була запроектована на попередньому етапі. На наступному етапі код верстату з ЧПК симулюється для подальшої перевірки на наявність помилок, тобто проводиться процес компіляції коду. Цей етап є обов'язковим перед початком роботи та проведення оброблення деталей на верстатах з ЧПК. Написання та подальше калібрування коду для верстату з ЧПК є достатньо трудомісткою процедурою. При цьому слід відзначити, що більша частина часу витрачається на етап роботи із САД/САМ моделями та операціями з ЧПК, що виконуються вручну. Така проблематика значно сповільнює процес оброблення деталей на верстатах з ЧПК. Ключем вирішення цієї проблеми є автоматизація процесу генерації коду для верстату з ЧПК.

У роботі запропоновано використання розробленого скрипта, результатом роботи якого є автоматична генерація кодів обробки деталей. Такі скрипти в подальшому можна використовувати для проведення обробки деталей на верстатах різних виробників. Крім того, проведено порівняння кодів обробки деталей, які були згенеровані у програмному забезпеченні Catia V5 та кодів, згенерованих автоматично. В результаті встановлено, що використання автоматичного підходу більш гнучке, оскільки зміна геометричних параметрів деталей викликає автоматичну зміну коду обробки такої деталі, на відміну від кодів, створених у Catia V5, в яких при мінімальній зміні розмірів деталі необхідно повторювати весь процес створення коду обробки.

Ключові слова: моделювання, коди обробки деталі, підпрограма, Catia V5, HAAS VM3, автоматизоване проектування, траєкторія руху інструменту, верстат з числовим програмним керуванням.

IULIA SOKOLAN

Khmelnytskyi National University
<https://orcid.org/0000-0002-0273-5719>
e-mail: sokolan.julia@gmail.com

IVAN LARIONOV

Khmelnytskyi National University
e-mail: Larionov674@gmail.com

PAVLO MAIDAN

Khmelnytskyi National University
<https://orcid.org/0000-0003-3319-8730>
e-mail: maidanps@gmail.com

KATERYNA SOKOLAN

Khmelnytskyi National University
<https://orcid.org/0000-0002-3513-8312>
e-mail: sokolan.kateryna@gmail.com

AUTOMATED GENERATION OF PART PROCESSING CODES ON CNC MACHINES

In modern realities, the process of generating codes for CNC machines is carried out manually, or the machine operator writes the code for the CAD model that was designed at the previous stage. At the next stage, the CNC machine code is simulated for further checking for errors, i.e., the code is compiled. This stage is mandatory before starting work and machining parts on CNC machines. Writing and subsequent calibration of the code for a CNC machine is a rather laborious procedure. It should be noted that most of the time is spent on the stage of working with CAD/CAM models and CNC operations performed manually. This problem significantly slows down the process of machining parts on CNC machines. The key to solving this problem is to automate the process of generating code for a CNC machine.

The paper proposes the use of a developed script, the result of which is the automatic generation of part processing codes. Such scripts can be used in the future to process parts on machines from different manufacturers. In addition, a comparison of the part processing codes generated in the Catia V5 software and the codes generated automatically is carried out. As a result, it was found that the use of an automatic approach is more flexible, since a change in the geometric parameters of parts causes an automatic change in the machining code of such a part, in contrast to the codes created in Catia V5, in which, with a minimal change in the dimensions of the part, it is necessary to repeat the entire process of creating a machining code.

Keywords: modeling, part machining codes, subroutine, Catia V5, HAAS VM3, computer-aided design, tool path, computer numerical control machine.

Постановка проблеми

Для генерації кодів з ЧПК використовуються дані, які в подальшому завантажуються в систему автоматизованого виробництва (САМ). Внаслідок цього можна досягти скорочення часу виконання замовлення, підвищення продуктивності та якості, а також покращення життєвого циклу розробки продукції. В роботі розглядається можливість запровадження систем автоматизованого планування проєктів (САРР) в якості ланку зв'язку між САМ та САД системами.

Слід враховувати, що в результаті автоматизації процесу генерації кодів для верстатів з ЧПК зано зменшується участь людини в процесі обробки, оскільки більшість завдань виконуються обчислювальною технікою. В результаті такої автоматизації можна досягнути двох протилежних наслідків. Першим наслідком стане мінімізація людського фактору та можливості виникнення помилок. Другий фактор призводить до усунення потреби у висококваліфікованих працівниках на етапі генерації коду.

Метою статті є процес автоматизації генерації кодів для верстатів з числовим програмним керування (ЧПК) на основі геометричних моделей автоматизованого проєктування (САПР), а також визначення потенціалу генерації таких кодів.

Виклад основного матеріалу

Основні етапи написання коду обробки деталі на верстаті з ЧПК наведені на рис. 1. Кожна із програм оброблення складається з певної кількості рядків, які, в свою чергу, називають кадрами керуючої програми (ККП). Кадри керуючої програми – це складові керуючої програми, які відпрацьовуються та вводяться як єдине ціле, при цьому містять не менше однієї команди. Зчитування програми відбувається покадрово. Кожному кадру прийнято призначати свій власний номер, який позначається літерою N та розташовується на початку кадру. Підготовчі коди позначаються адресою, яка позначається літерою G та вони визначають налаштування системи ЧПК на виконання роботи. Допоміжні кадри позначаються M, які призначені для керування режимами роботи.

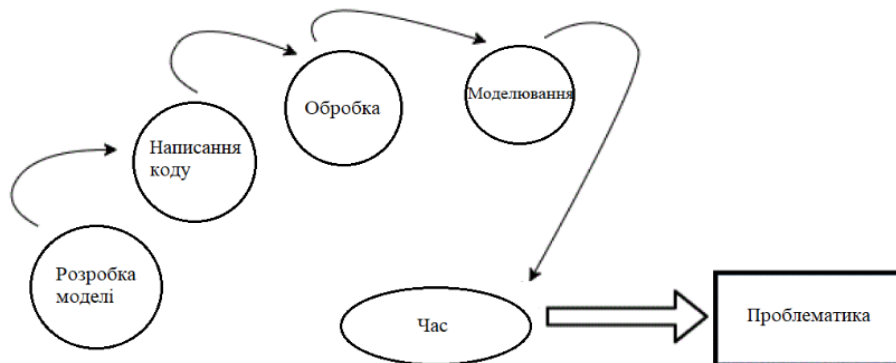


Рис. 1. Проблематика, що розглядається

В роботі досліджується процес автоматизованої генерації кодів для двох верстатів. Перший верстат виробництва Roland (Німеччина) моделі MDX-40A (рис. 2) представляє собою універсальний фрезерно-гравіювальний верстат з ЧПК [1]. Цей верстат може підтримувати установку як 3D-скануючої головки, так і поворотної осі. Оцифрування тривимірних об'єктів можна виконувати на 3D сканері, після чого можна виконувати кругову та чотириох сторонню обробку за допомогою поворотної осі. Оператору достатньо легко освоїти управління верстатом MDX-40A, оскільки в ньому наявна можливість автоматичного визначення нульової точки по осі Z, а також в нього просте програмне забезпечення та USB інтерфейс. На цьому верстаті можна створювати майстер-моделі, готові вироби та прототипи.

Другий досліджуваний верстат – HAAS VM-3 (рис. 3). Верстати HAAS серії VM представляють собою високопродуктивні вертикальні оброблювальні центри, які здатні забезпечувати не тільки жорсткість та точність обробки, але й температурну стабільність для виготовлення штампів та прес-форм. Кожна модель відрізняється розміром робочого простору, швидкість обертання шпинделя сягає 12000 об/хв, стіл розрахований на різні пристрої. До стандартних функцій належать пристрій заміни інструменту бокової установки, автоматичний пневматичний пістолет, мають в наявності сопло подачі змащувально-охолоджувальної рідини із можливістю його програмування, а також функцію повного попереднього перегляду [2].

Робота виконується на двох різних верстатах. Для автоматичної генерації G-кодів запуску верстату MDX 40A і подальшого створення будь-якої заданої САД-моделі створюється відповідний фреймворк. Робота із фреймворком поетапна, на першому етапі розробляється та імпортується модель у програмне забезпечення фрезерного верстату MDX 40A SRP pro, яке використовує відповідні формати файлів. Генерування коду для програмного забезпечення SRP pro на цьому етапі виконується вручну.

Наступний фреймворк створюється для автоматичної генерації кодів в програмному забезпечення SRP засобами автоматизації для обробки на верстатах з ЧПК. В результаті порівнюються результати по часу виконання ручного та автоматично згенерованого коду, а також аналізуються дані стосовно точності проведених обробок.



Рис. 2. Верстат Roland MDX 40A



Рис. 3. Верстат HAAS VM-3

В автоматизованому режимі проектується параметрична модель в програмі Catia V5. Код моделі повинен бути написаний на основі аналізу траєкторії руху інструменту, його положення, розміру, аналізу розмірів моделі, тощо. Цей набір параметрів буде першим для моделі.

Після аналізу коду в програмі Catia V5 буде створено другий набір параметрів. Такий набір створюється для присутніх у коді змінних, до яких можна віднести глибину різання, координати положення по осям X та Y, тощо. На цьому етапі відбувається зв'язок наборів параметрів за допомогою формул, в результаті чого будується взаємозв'язок геометрії із кодом. У випадку, якщо використовується автоматизований метод генерації NC-кодів, то вони генеруються автоматично.

Таким чином після проведеного аналізу коду для будь-якого із вимірів виробу та/або деталі у оператора є можливість згенерувати нові коди у випадку зміни розмірності моделі. Змінити розмірність моделі можна шляхом зміни основних її параметрів.

Генерування коду відбувається автоматично в текстовому форматі, після чого його можна змінити на формат файлу «.NC» для подальшого запуску на верстаті з ЧПК. Такий підхід буде також використовуватись для верстату з ЧПК HAAS VM 3. Використання цього коду призводить до моделювання на панелі дисплея верстату з ЧПК відповідної симуляції та перевірки точності написаного коду. Поки триває симуляція відповідно до стандартів верстата та вимог проектування можна ввести необхідні зміни, наприклад, розмір інструменту, встановити інший початок координат, змінити інструмент, тощо.

На рис. 4 наведено покрокову методика процесу автоматизованої генерації кодів для обробки деталей на верстатах з ЧПК, який починається із створення тривимірної CAD-моделі. Ця модель на наступному етапі експортується в САМ інтерфейс з метою визначення параметрів та послідовності операцій. На етапі подальшої обробки відбувається пов'язування визначених для моделі параметрів із сценарієм роботи ЧПК, тобто будується взаємний зв'язок геометрії та коду. На наступному етапі код експортується у верстат з ЧПК для оброблення. Такий скрипт дозволяє визначати такі параметри оброблення деталі, як швидкість подачі, швидкість обертання шпинделя, глибину різання, тощо.

Робочі рухи органів у верстатах з ЧПК, а також здійснення їх переміщень, програмуються в межах системи координат (рис. 5). В основу системи координат покладена стандартна права система координат з осями x, z, y, які вказують напрями руху інструменту (додатні) відносні нерухомої заготовки. У випадку, якщо сам інструмент нерухомий, при цьому відбувається рух заготовки, то систему координат позначають x', z', y' [3].

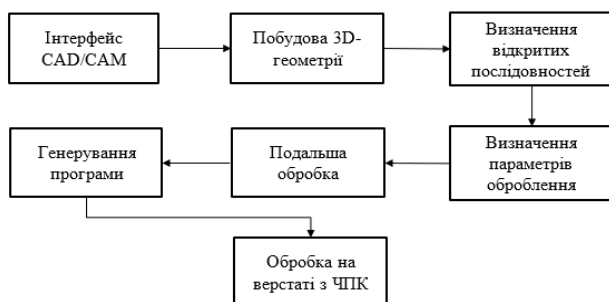


Рис. 4. Методика проведення досліджень

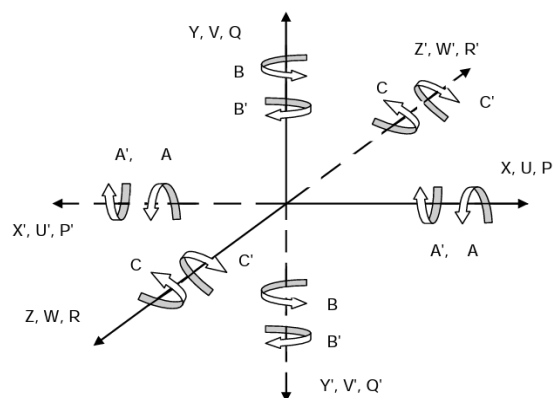


Рис. 5. Системи координат верстатів з ЧПК

Програмне забезпечення CATIA V5 відноситься до важких систем автоматизованого проектування. В CATIA V5 реалізовано всебічний комплексний інтегрований підхід, що дозволяє успішно здійснювати в межах однієї системи весь цикл створення нового виробку – від етапу розробки концептуальних моделей до етапу підготовки керуючих програм для обробки на верстатах з ЧПК. У ПЗ CATIA V5 наявні модулі для створення програм на верстатах з ЧПК. Використання таких модулів дають можливість створювати програми обробки для 2,5-, 3-, 4-, 5-осової токарної та фрезерної обробки. Ряд науковців проводили дослідження точності та ефективності використання автоматично створених у CATIA V5 кодів для проведення обробки деталей на верстатах з ЧПК [4–7].

Для порівняння результатів автоматичної генерації кодів обробки деталей використовувалось два методи. Перший, ручний метод, полягав у тому, що на основі побудованої в CATIA V5 геометричної моделі, генерувався код обробки такої деталі самим програмним забезпеченням. У другому, автоматизованому методі генерації кодів, на першому етапі також використовувалась геометрична модель з CATIA V5. В подальшому, використовувався скрипт генерації коду, який був написаний на Visual Basic for Applications [8].

Коди обробки деталей, написані ручним методом та автоматизованим, перевірялись для моделей, які наведені на рис. 6.

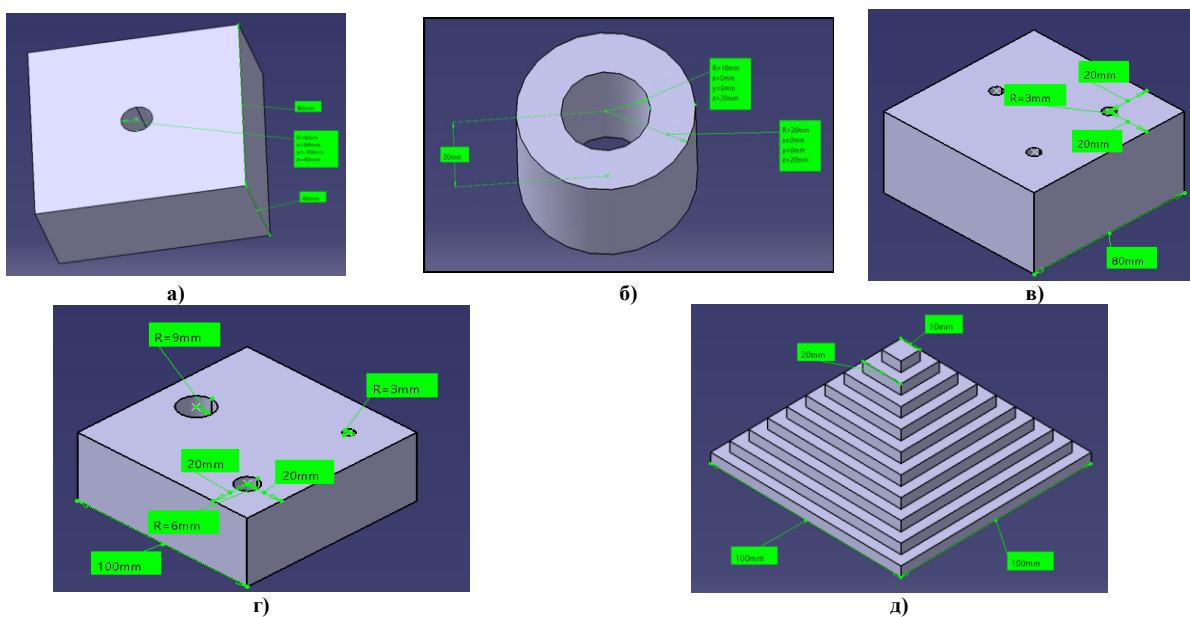


Рис. 6. Моделі деталей, процес обробки яких розглядається в роботі:
 а - корпус з отвором; б - порожнистий циліндр; в - корпус з отворами однакового радіусу;
 г - корпус з отворами різного радіусу; д - піраміда

Основним недоліком генерації кодів ручним методом за допомогою CATIA V5 є те, що у випадку зміни будь-яких параметрів моделі або її розмірів, процес створення коду необхідно повністю проводити з першого до останнього етапу, тобто вибір поверхонь і площин, процес обробки, тощо. Іншим недоліком при генерації кодів з CATIA V5 є те, що ці коди достатньо довгі і містять від 300 до 2500 рядків.

Для порівняння згенерованих кодів для корпусу із отвором, який показаний на рис. 6, а, спочатку виконувалась генерація коду обробки ручним методом у CATIA V5. В результаті код обробки деталі, сформований у CATIA V5 формувє 388 рядків коду.

У випадку застосування автоматизованого методу, першим кроком автоматичної генерації коду обробку деталі є створення параметрів та прив'язка параметрів, які використовуються в Catia V5 і VB скрипті до геометрії деталі. Параметри профілю корпусу – це X1, Y1, X2, Y2, а параметри профілю отвору – X3 та Y3 (рис. 7).

```

Point_Lenght=40mm=Length /2
point_breadth=40mm=breathd /2
x1=40mm=Length /2
y1=40mm=breathd /2
Radius=3mm=holedia /2
Height=40mm=40mm
x2=40mm=x1
y2=57mm=y1 +Workpiece_and_tool\Tool_Radius +1mm
x3=0mm
y3=0mm
    
```

```

x_2val = parameters1.Item("x2").Value
y_2val = parameters1.Item("y2").Value
x_1val = parameters1.Item("x1").Value
y_1val = parameters1.Item("y1").Value
x_3val = parameters1.Item("x3").Value
y_3val = parameters1.Item("y3").Value
    
```

Рис. 7. Параметри деталі:
 а – параметри у Catia V5, б – параметри обробки деталі у скрипті

Розглядається корпус із габаритними розмірами 80 мм (ширина і довжина). Оскільки центром координат при обробці деталей на верстатах з ЧПК є центр деталі, то координати зовнішніх граней корпусу мають відповідно значення 40 та -40 (рис. 8). Другим етапом обробки деталі є створення отвору. Для цього необхідно задати правило для визначення параметрів X3 та Y3 (рис. 9).

Третім етапом обробки деталі є встановлення інструменту та заготовки. Параметри "XW" і "yW" створюються за допомогою формули, яка в свою чергу створює прив'язку довжини і ширини заготовки відповідно. Ці параметри для позиціонування інструмента в певній точці. Параметри uw_val і xw_val – це параметри, що викликаються VB скриптом для отримання відповідних значень параметрів uw та XW. В результаті роботи написаного скрипта було автоматично згенеровано код обробки деталі, який представлено у таблиці 1.

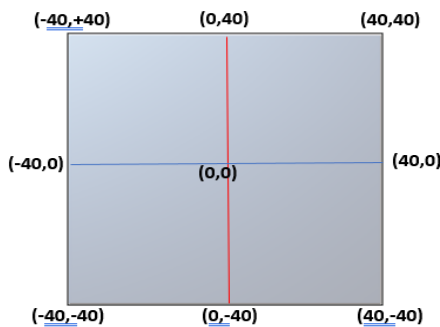


Рис. 8. Координати точок обробки корпусу без отвору

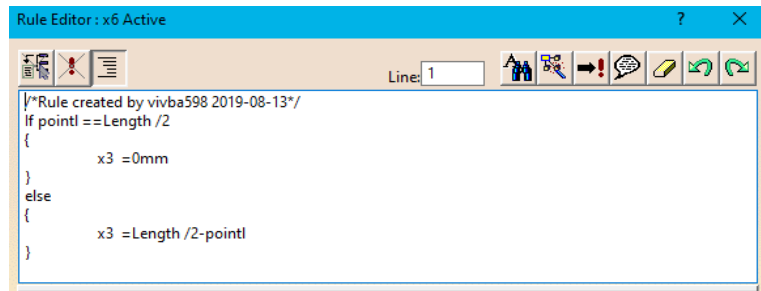


Рис. 9. Правило для параметрів отвору

Команда Print #1 використовується для автоматичної генерації коду під час запуску програми. Розглянемо більш детально кожен із рядків.

1. Знак % використовується на початку та в кінці програми обробки, оскільки він є обов'язковим елементом синтаксису програми обробки на верстаті з ЧПК. Відсутність цього знаку призведе до помилки.
2. Назва програми, яка починається з літери O, після якої вказуються цифри і назва моделі деталі.
- 3-4. Ініціалізація програми обробки за допомогою стандартних кодів, що є обов'язковими на всіх верстатах з ЧПК.
5. За конкретно обраний інструмент відповідає команда T14, а за зміну інструменту – команда M6.
6. G54 означає зміщення робочих координат, що при обробці деталі використовується встановлення початкової/опорної точки ріжучого інструменту на заготовці. Напрямки і значення параметрів X і Y обираються на основі бажаного положення початку траєкторії руху інструменту. Швидкість обертання шпинделя – S, запуск шпинделя – команда M3.

7. В цьому рядку вказується відстань інструменту від заготовки та подача ЗОР. Команда M88 відповідає за потік ЗОР через інструмент або всередині самого інструменту.

8. Запуск роботи інструменту.

9. Виклик підпрограми, позначення L16 відповідає за кількість циклів роботи інструменту для отримання спроектованого профілю корпусу. У випадку виклику підпрограми, процес обробки деталі переходить до рядка 23.

Посилання на підпрограму, яка викликається рядком 9.

Значення Z задається як -2,5 мм, що означає рух вглибину. Кількість циклів обробки деталі вказується при виклику підпрограми (16 разів). Це означає, що під час кожного циклу обробки деталі проводиться різання на глибину 2,5 мм.

На цьому етапі дуже важливо обрати оптимальне значення глибини різання, тобто значення Z з метою зменшення ризику пошкодження інструменту. Виробники інструменту рекомендують використовувати значення, яке лежить в межах від 2 до 3 мм. Враховуючи, що необхідна висота корпусу становить 40 мм, кількість циклів виконання підпрограми обиралась значенням, яке буде кратне Z ($16 \times 2,5 = 40$ мм). Рядки коду з 26 по 30 застосовуються для виконання операцій над заготовкою для отримання корпусу і видалення зайвого матеріалу в заготовці.

Значення D застосовується для визначення діаметру інструменту. На початку обробки інструмент позиціонується в точці із координатами Y=+80 та X=-80 на відстані від заготовки (рис. 10). Після обробки корпусу рядок коду під номером 32 відповідає за закінчення підпрограми і програма переходить до зчитування рядку під номером 10. Цей рядок застосовується для зупинки виконання підпрограми після завершення циклу зазначених в ній операцій по обробці деталі. Рядок 11, в свою чергу, вимикає подачу ЗОР через ріжучий інструмент.

У 14 рядку для обробки отвору у корпусі обирається інструмент. 15 рядок відповідає за установку інструменту в центрі. За сам процес свердлування отвору відповідають команди у 18 рядку. В цьому рядку коду зазначається, що радіус отвору становить 3 мм, а глибина – 30 мм. Показник Q застосовується для зазначення кількості циклів свердлування. Зупинка процесу свердлування забезпечується командами, які вказані у рядку 19.

Автоматично згенерований код обробки корпусу із отвором

Номер рядка коду	Код у скрипті	Кінцевий код обробки деталі	Номер рядка коду	Код у скрипті	Кінцевий код обробки деталі
1	Print #1, "%"	%	2	Print #1, "O00004 (CubeWithSingleHole)"	O00004 (CubeWithSingleHole)
3	Print #1, "G28 G91 Z0"	G28 G91 Z0	4	Print #1, "G0 G40 G80 G90"	G0 G40 G80 G90
5	Print #1, "T14 M6"	T14 M06	6	Print #1, "G54 X" & -xw_val; "Y" & yw_val; "S" & s_1val; "M3"	G54 X-80 Y80 S1200 M3
7	Print #1, "G43 Z3 H14 M88"	G43 Z3 H14 M88	8	Print #1, "G01 Z0. F" & f_1val; ""	G01 Z0. F120
9	Print #1, "M97 P901 L16"	M97 P901 L16	10	Print #1, "M01"	M01
11	Print #1, "M89"	M89	12	Print #1, "G28 G91 Z0"	G28 G91 Z0
13	Print #1, "G0 G40 G80 G90"	G00 G40 G80 G90	14	Print #1, "T6 M6"	T6 M6
15	Print #1, "G54 X" & x_3val; " Y" & y_3val; " S" & s_2val; " M3"	G54 X0. Y0. S1100 M03	16	Print #1, "G43 Z3 H6 "	G43 Z3. H06
17	Print #1, "M08"	M08	18	Print #1, "G83 Z" & -D_val; " R" & h_rad; " Q5"; " F" & f_2val; ""	G83 Z-30. R3. Q5. F100.
19	Print #1, "G80 M9"	G80 M9	20	Print #1, "G28 G91 Z0"	G28 G91 Z0
21	Print #1, "G28 G91 Y0"	G28 G91 Y0	22	Print #1, "M30"	M30
23	Print #1, "N901"	N901	24	Print #1, "G91 Z-2.5 F" & f_1val; " M08"	G91 Z-2.5 F120 M08
25	Print #1, "G90"	G90	26	Print #1, "G01 G41 X" & -xw_val; " Y" & y_1val; " D14"	G01 G41 X-80 Y40 D14
27	Print #1, "G01 X" & x_1val; ""	G01 X40	28	Print #1, "G01 Y" & -y_1val; ""	G01 Y-40
29	Print #1, "G01 X" & -x_2val; ""	G01 X-40	30	Print #1, "G01 Y" & y_2val; "	G01 Y57
31	Print #1, "G01 G40 X" & -xw_val; " Y" & yw_val;	G01 G40 X-80 Y80	32	Print #1, "M99"	M99

За сам процес свердлування отвору відповідають команди у 18 рядку. В цьому рядку коду зазначається, що радіус отвору становить 3 мм, а глибина – 30 мм. Показник Q застосовується для зазначення кількості циклів свердлування. Зупинка процесу свердлування забезпечується командами, які вказані у рядку 19. Рядки 20 і 21 застосовуються наприкінці програми для відведення інструменту від заготовки після завершення процесу обробки деталі. Кінець програми зазначений у рядку 22.

Аналогічним чином було проведено дослідження та порівняння кодів ручним та автоматизованим методом для деталей, зображених на рис. 6, б-д. Траєкторія руху інструменту та її відповідність автоматично згенерованим кодам обробки порожнистого циліндру наведена на рис. 11.

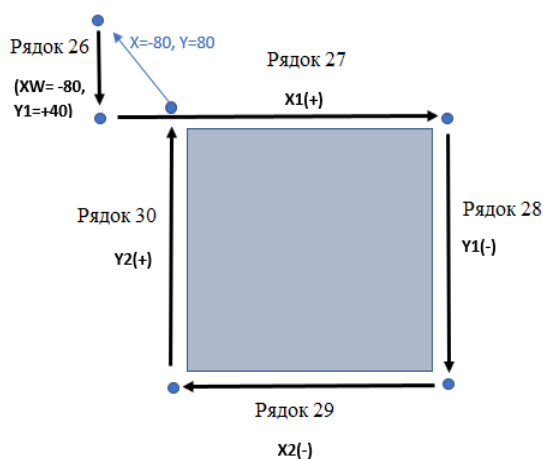


Рис. 10. Траєкторія руху інструменту при обробці корпусу

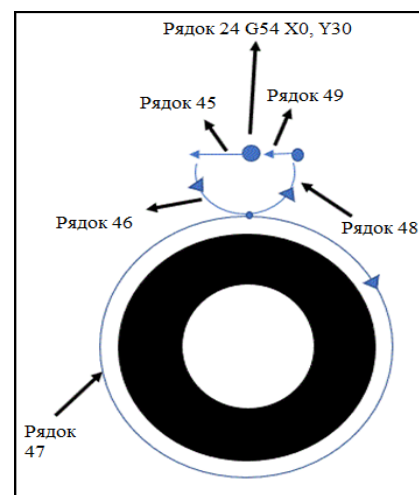


Рис. 11. Траєкторія руху інструменту при обробці порожнистого циліндру

При розгляді деталей типу корпус із отворами (рис. 6 в, г) використовувались два різні підходи. В першому підході розміщення отворів привласнювалось значення, яке не мало математичного зв'язку із габаритними розмірами корпусу (умовно фіксований отвір). В другому підході створювалась математична залежність між розмірами корпусу і розміщенням отворів на площині корпусу (умовно рухомий отвір). Основні відмінності у згенерованих кодах обробки отворів показані у таблиці 2 та на рис. 12.

При аналізі відмінностей у кодів обробки корпусу із отворами різного діаметру (рис. 6, г) у порівнянні із корпусом із отворами однакового діаметру було встановлено, що відмінність у кодах стосується лише рядків зміни інструменту.

Основна відмінність моделей, зображених на рис. 6, а-г у порівнянні із гранюю пірамідою (рис. 6, д) з точки зору формування кодів обробки деталі на верстаті з ЧПК полягає в тому, що при обробці піраміди ріжучий інструмент на кожному етапі обробки змінює своє положення відносно центру заготовки та відбувається збільшення глибини різання.

Аналогічно до попередніх моделей, перший етап роботи із пірамідою полягає у створення відповідних параметрів та їх подальша прив'язка в Catia V5 і скрипті VB до геометрії деталі. Для проведення такої обробки в скрипті автоматичної генерації кодів буде використовуватись цикл For.

Таблиця 2

Відмінності при обробці отворів

Номер рядку обробки деталі	Модель із математичною залежністю між отворами та зовнішніми розмірами корпусу	Модель із фіксованим положенням отворів по відношенню до корпусу
23	T6 M6	T6 M6
24	G54 X25 Y25 S1100 M3	G54 X30 Y30 S1100 M3
25	G43 Z3 H6 M8	G43 Z3 H6 M8
26	G83 Z-40 Q5 R3 F120	G83 Z-40 Q5 R3 F120
27	X-25 Y25	X-30 Y30
28	X0 Y-25	X0 Y-30
29	G80 M9	G80 M9

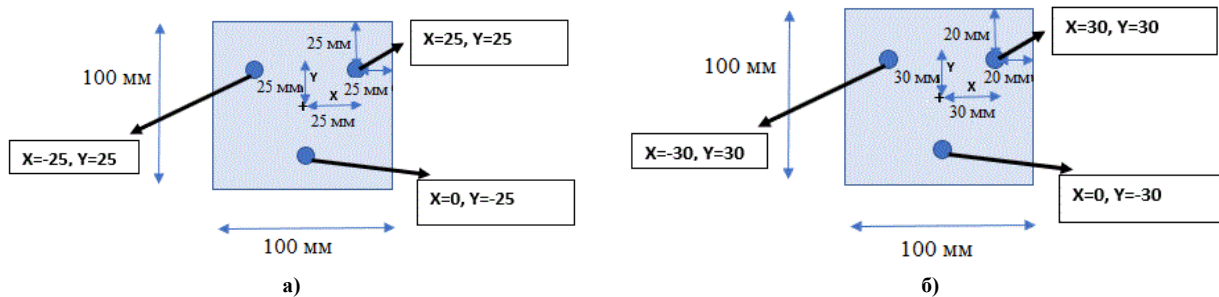


Рис. 12. Відмінності при обробці отворів у корпусі:
а – умовно фіксований отвір; б – умовно рухомий отвір

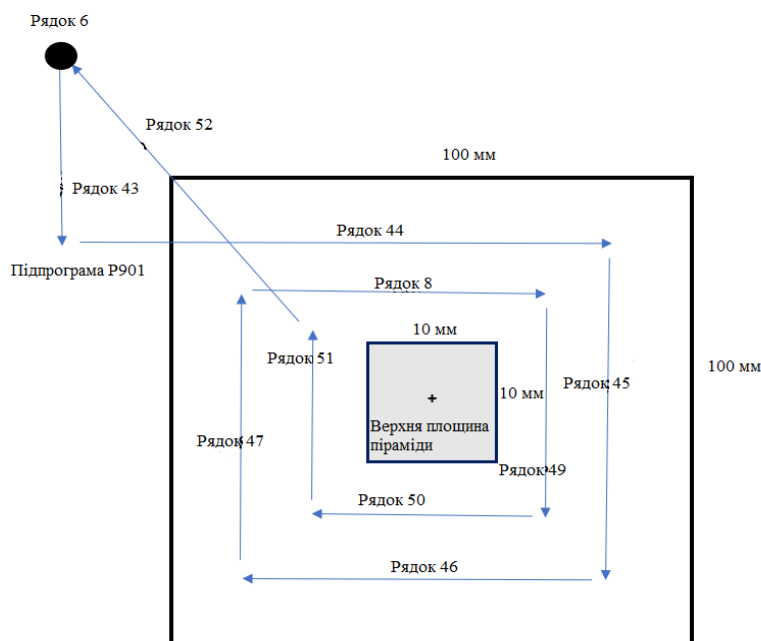


Рис. 13. Траєкторія руху інструменту при обробці піраміди

Після обробки верхньої площини піраміди в другій підпрограмі вже буде вирізана глибина 5 мм. Отже, при проведенні наступної операції ріжучий інструмент повинен в напрямку Z рухатись ще на 5 мм. По цій причині в рядку значення становить Z-5. Ця значення для кожної підпрограми отримуються на основі параметрів, створених в Catia V5, які в подальшому викликаються у скрипті. Далі викликається наступна підпрограма. Такі підпрограми викликаються до тих пір, поки ріжучий інструмент не проведе обробку основи піраміди. На рис. 13 наведено траєкторію руху інструменту при обробці піраміди та її відповідність рядкам автоматично згенерованого коду обробки деталі.

Порівняння кількості рядків коду обробку деталі ручним та

автоматизованим методом наведено у таблиці 3.

Порівняння кількості рядків коду різними методами

Деталь	Кількість рядків коду		Кількість підпрограм
	Ручний метод	Автоматизований метод	
Корпус із отвором	388	32	1
Порожнистий циліндр	267	50	1
Корпус із отворами однакового радіусу	418	45	3
Корпус із отворами різного радіусу	432	53	3
Піраміда	1206	142	9

Висновки

Після проведення аналізу щодо генерації кодів для деталей ручним та автоматизованим методом, можна відмітити, що такий метод є більш оптимальним для обробки деталей, оскільки код обробки деталі автоматизованим методом зменшується в декілька разів у порівнянні із кодом, який формується в програмному забезпеченні Catia V5. Крім того, використання такого методу має більший рівень гнучкості програм обробки деталей, оскільки при зміні параметрів моделі у програмі, в якій створювалась геометрія деталі, згенерований автоматично код обробки деталі буде змінюватись без втручання оператора верстату з ЧПК. Внаслідок використання скриптів автоматизації кодів обробки деталей, такі коди можна використовувати на різних верстатах, наприклад, Roland MDX 40A та HAAS VM3.

Література

1. Фрезерно-гравіювальний верстат MDX-40A [Електронний ресурс]. – Режим доступу : <https://rdm24.ua/products/frezernye-stanki/mdx-40a/>
2. VM3 [Електронний ресурс]. – Режим доступу : <https://www.haascnc.com/ps/machines/vertical-mills/mold-machines/models/vm-3.html>
3. Міранцов С.Л. Системи автоматизованого проектування верстатів з ЧПК : навчальний посібник / С.Л. Міранцов, В.І. Тулупов, С.Г. Онишук, Ю.Б. Борисенко, Є.В. Мішура, О.С. Ковалевська. – Краматорськ : ДДМА, 2011. – 152 с.
4. Konig O., Wintermantel M. CAD-based Evolutionary Design Optimization with CATIA V5. https://www.researchgate.net/publication/228713225_CAD-based_Evolutionary_Design_Optimization_with_CATIA_V5
5. Dubovska R., Jambor J., Majerik J. Implementation of CAD/CAM system CATIA V5 in Simulation of CNC Machining Process. *Procedia Engineering* 69. 2014. p. 638-645
6. Kowalski M., Zawadzki P. Intelligent manufacturing templates for CNC machine programming. 15th Global Congress on Manufacturing and Management (GCMM 2021). 20198 (2022). p. 1-7.
7. Кришук М.Г. Робота в програмному продукті CATIA. Загальні відомості. Частина 1 : методичні вказівки до виконання комп'ютерного практикуму / М.Г. Кришук, А.В. Трубін, Н.Ф. Тертишна, В.О. Єщенко. – К. : НТУУ “КПІ імені Ігоря Сікорського”, 2017. – 78 с.
8. Visual Basic for Applications. https://ru.wikipedia.org/wiki/Visual_Basic_for_Applications

References

1. Milling and engraving machine MDX-40A. <https://rdm24.ua/products/frezernye-stanki/mdx-40a/>.
2. VM3. <https://www.haascnc.com/ps/machines/vertical-mills/mold-machines/models/vm-3.html>
3. Mirantsov S.L., Tulupov V.I., Onyshchuk S.G., Borisenko Y.B., Mishura E.V., Kovalevska O.S. Systems of computer-aided design of CNC machine tools: a textbook. Kramatorsk: DGMA, 2011. 152 p.
4. Konig O., Wintermantel M. CAD-based Evolutionary Design Optimization with CATIA V5. https://www.researchgate.net/publication/228713225_CAD-based_Evolutionary_Design_Optimization_with_CATIA_V5
5. Dubovska R., Jambor J., Majerik J. Implementation of CAD/CAM system CATIA V5 in Simulation of CNC Machining Process. *Procedia Engineering* 69. 2014. p. 638-645
6. Kowalski M., Zawadzki P. Intelligent manufacturing templates for CNC machine programming. 15th Global Congress on Manufacturing and Management (GCMM 2021). 20198 (2022). p. 1-7.
7. Kryshchuk M.G., Trubin A.V., Tertyshna N.F., Yeshchenko V.O. Work in the CATIA software product. General information. Part 1: methodical instructions for the implementation of a computer workshop. K.: NTUU "KPI", 2017. 78 p.
8. Visual Basic for Applications. https://ru.wikipedia.org/wiki/Visual_Basic_for_Applications.