

Radosław SIWIEC¹

Opiekun naukowy: Roman STRYCZEK²

OPRACOWANIE POSTPROCESORA DLA GENEROWANIA ŚCIEŻKI NARZĘDZIA ROBOTA PRZEMYSŁOWEGO

Streszczenie: Niniejszy artykuł poświęcony jest budowie oraz zasadom działania oprogramowania *PCB CAM Processor*. Aplikacja stanowi zintegrowany pakiet CAD / CAM umożliwiający projektowanie i wytwarzanie płyt PCB metodą grawerowania. Programy CNC generowane przez oprogramowanie mogą być elastycznie dostosowane do danego typu obrabiarki. Prezentowany pakiet oprogramowania pozwala wygenerować programy sterujące zarówno dla obrabiarek CNC, jak i robotów przemysłowych.

Słowa kluczowe: CAD, CAM, CNC, PCB, C++, OpenGL grafika komputerowa, postprocesory

POSTPROCESSOR DEVELOPMENT FOR GENERATING INDUSTRIAL ROBOT TOOL PATH

Summary: This article is dedicated to the principles of construction and operation of the *PCB CAM Processor* software. The application is an integrated suite of CAD / CAM enables the design and manufacture of PCB engraving method. CNC programs generated by the software can be flexibly adapted to the type of machine. It is possible to use CNC machines as well as industrial robots.

Keywords: CAD, CAM, CNC, PCB, C++, OpenGL, computer graphics, postprocessors

1. Wstęp

Ręczne opracowanie programów na obrabiarki sterowane numerycznie jest trudne i wymaga niemałego doświadczenia. Programy CNC (Computerized Numerical Control) dla skomplikowanych modeli mogą zajmować nawet kilkaset linii kodu. Ponadto istnieje duże ryzyko popełnienia błędu, którego wykrycie jest możliwe jedynie przy użyciu symulatorów obróbki. Rozwiązaniem tych problemów są

¹ Akademia Techniczno-Humanistyczna, Wydział Budowy Maszyn i Informatyki, Automatyka i Robotyka: Projektowanie zrobotyzowanych stanowisk pracy, radov@poczta.fm
² prof. ATH dr hab. inż., Akademia Techniczno-Humanistyczna, Wydział Budowy Maszyn i Informatyki, rstryczek@ath.bielsko.pl

pakiety oprogramowania typu CAM (Computer Aided Manufacturing), które na podstawie modelu CAD (Computer Aided Design) generują gotowe programy obróbki.

Oprogramowania CAM dostępne na rynku mają jednak swoje wady. Ceny takich aplikacji sięgają nawet setek tysięcy Euro, a możliwości bywają ograniczone. Przykładowo, aby generować kod dla konkretnej maszyny CNC należy dokupić odpowiedni moduł zgodny z jej typem, co oczywiście przekłada się na dodatkowe koszty. Rozwiązaniem tych problemów jest własne środowisko CAM, dzięki któremu koszty oprogramowania i rodzaj posiadanej obrabiarki nie mają znaczenia.

Wytwarzanie płyt PCB (Printed Circuit Board) metodą grawerowania jest sposobem na szybkie prototypowanie. Przeznaczone jest dla testowania układów elektronicznych w środowisku domowym czy akademickim. Umożliwia szybkie i tanie budowanie prototypów, podczas gdy zamawianie płytek drukowanych w profesjonalnych zakładach produkcyjnych niesie za sobą duże koszty. Ewentualne błędy w projekcie nie wymagają ponownego zamawiania prototypu, dzięki czemu oszczędza się czas i pieniądze.

2. Możliwości oprogramowania

Oprogramowanie opisane w tym artykule umożliwia szybkie prototypowanie układów elektronicznych. Na podstawie modelu stworzonego w oprogramowaniu generowany jest program na niemal dowolny typ maszyny sterowanej numerycznie. Jest to możliwe dzięki zastosowaniu szablonów zgodnych z wybraną obrabiarką. Program obróbki obwodu drukowanego podzielony jest na dwie operacje: grawerowanie układu ścieżek oraz wiercenie otworów montażowych dla elementów elektronicznych. Każda z tych operacji została rozbita na szereg elementarnych zabiegów, które składają się na gotowy program. Użytkownik musi zdefiniować dwa szablony, po jednym dla każdej z operacji. Każdy szablon składa się z fragmentów kodu CNC zgodnych z językiem wybranej maszyny, zaś każdy fragment opisuje pewien elementarny zabieg. Każde pole szablonu posiada szczegółowy tekst odpowiedzi.

2.1. Wiercenie

Szablon wiercenia definiuje:

- Ustawienia generatora
- Początek programu obróbki
- Podprogram wiercenia
- Podprogram wymiany wiertła
- Zakończenie programu

W ustawieniach generatora definiuje się rozszerzenie pliku wyjściowego zgodne z wybraną maszyną oraz przesunięcie układu współrzędnych. Początek i koniec programu definiują składnię programu w wybranym języku oraz wszystkie wymagane operacje przygotowawczo-zakończeniowe. Podprogram wiercenia

opisuje pełny cykl wykonania pojedynczego otworu, zawierający: najazd nad otwór, wykonanie wiercenia oraz wycofanie - przygotowanie do kolejnego wiercenia. Podprogram wymiany wiertła jest uruchamiany po wykonaniu wszystkich otworów danej średnicy. Otwory wykonywane są w kolejności rosnącej.

2.2. Grawerowanie

Szablon grawerowania definiuje:

- Ustawienia generatora
- Początek programu obróbki
- Dojazd do materiału
- Ruch roboczy - interpolacja liniowa
- Ruch roboczy - interpolacja kołowa zgodnie z ruchem wsk. zegara
- Ruch roboczy - interpolacja kołowa przeciwnie do ruchu wsk. zegara
- Wyjazd z materiału
- Zakończenie programu

W tej operacji wykonywane są ruchy robocze po zamkniętych konturach, tworzących na płycie drukowanej pola o wspólnym potencjale elektrycznym, będące całkowicie odizolowane od pozostałej części obwodu drukowanego. Ustawienia generatora zawierają dodatkowo definicję promienia narzędzia, dzięki czemu program umożliwia jego korekcję. Jest to szczególnie istotne dla maszyn takich jak roboty przemysłowe, których oprogramowanie nie posiada funkcji kompensacji promienia narzędzia.

Początek i koniec programu pełni tu analogiczną funkcję jak w operacji wiercenia. Rozpoczęcie i zakończenie obróbki poszczególnych konturów definiują dojazd i wyjazd z materiału. Interpolacja liniowa i kołowa używana podczas prowadzenia narzędzia po zadanym konturze jest również definiowana w odpowiednich polach szablonu. W przypadku opisu interpolacji kołowej istnieje rozróżnienie ze względu na kierunek obróbki. Jest to podyktowane zapisem tego ruchu roboczego w G kodzie, gdzie G2 oznacza ruch zgodny z ruchem wskazówek zegara, a G3 w kierunku przeciwnym. W przypadku stosowania robotów przemysłowych istotny jest również punkt pośredni łuku. Z kolei w maszynach sterowanych układami Sinumerik można używać promienia łuku, który jest liczbą ujemną, gdy opisywany łuk ma kąt rozwarcia większy niż 180°. Z uwagi na różny zapis interpolacji kołowej program wylicza szereg jej paramentów, umożliwiając elastyczne dostosowania do konkretnej maszyny. Wszystkie dostępne parametry pokazano na rysunku poniżej.

Wykonanie przejazdu roboczego po łuku - dostępne parametry interpolacji kołowej:

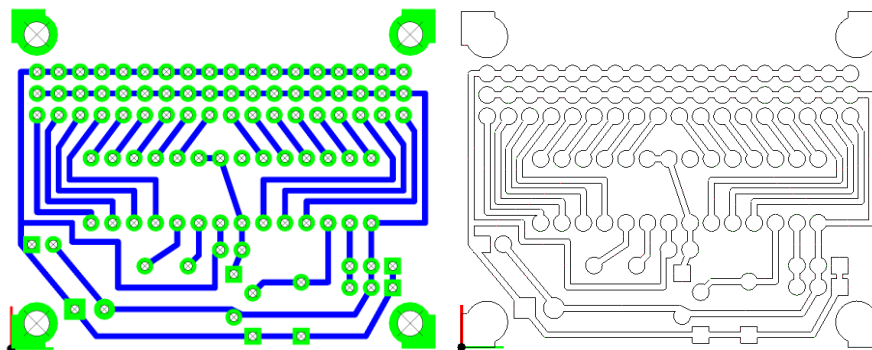
- @X@, @Y@ - wsp. punktu końcowego
- @AX@, @AY@ - wsp. punktu pośredniego
- @I@, @J@ - wsp. środka łuku względem punktu początkowego (G-code)
- @R@ - promień łuku
- @CR@ - promień łuku (Sinumerik - ujemny dla rozwarcia > 180°)
- @P@ - kąt rozwarcia łuku

Rysunek 1. Podpowiedź kreatora szablonu dot. interpolacji kołowej

Jak widać na powyższej ilustracji wszystkie parametry używane w polach szablonu muszą być zawarte pomiędzy znakami @. Aby użyć wybranego parametru wewnątrz bloku szablonu, należy wpisać: @nazwa parametru@. Podczas generowania pliku z kodem programu wszystkie miejsca użycia powyższej składni są zamieniane na dane liczbowe obliczone na podstawie modelu.

2.3. Tryb symulacji

Oprogramowanie umożliwia wygenerowanie podglądu ścieżki narzędzia oraz prześledzenie jego toru. Możliwe jest sterowanie prędkością symulacji podając prędkość ruchów roboczych w m/s. Symulacja możliwa jest zarówno dla operacji grawerowania jak i wiercenia. W tym drugim przypadku pokazana jest kolejność wykonywania otworów. Użycie tego trybu pozwala zweryfikować poprawność wykonanych przez program obliczeń. Kolejność operacji widoczna na animacji jest całkowicie zgodna z tym co jest generowane do pliku wyjściowego. Poniżej przedstawiono przykładowy model oraz symulację grawerowania.



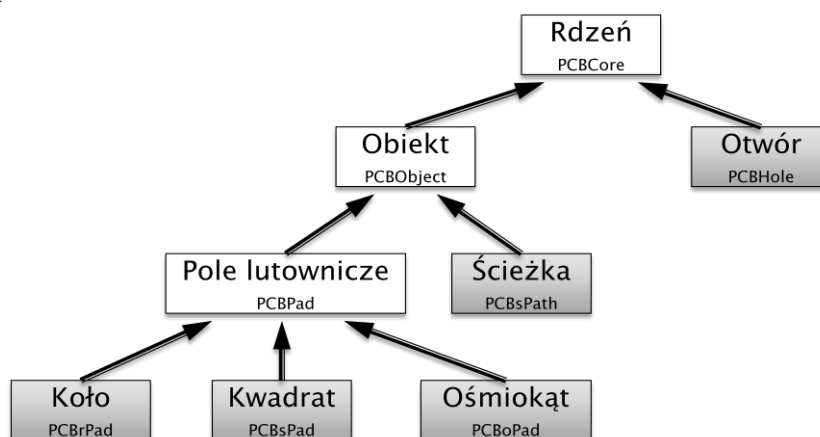
Rysunek 2. Przykładowy model oraz jego obliczony kontur

3. Opis modelu

Modele tworzone w prezentowanym oprogramowaniu składają się z elementów takich jak: prostoliniowy odcinek ścieżki, pola lutownicze w trzech możliwych kształtach (koło, kwadrat, ośmiokąt) oraz otwory. Elementy te są ze sobą powiązane za pomocą węzłów, które wyznaczają ich pozycję oraz wiążą je ze sobą. Otwory oraz pola lutownicze do swojego opisu wykorzystują jeden węzeł. Odcinki ścieżek potrzebują dwóch węzłów, po jednym dla każdego z końców. Zmiana pozycji węzła powoduje przemieszczenie wszystkich elementów do niego podłączonych.

Wymienione powyżej elementy (z pominięciem otworów) posiadają pole ograniczone pewnym zamkniętym konturem. Kontur taki można opisać zbiorem linii i łuków zwanych dalej prymitywami. Wspomniane prymitywy opisujące kontur danego elementu są w sensie programistycznym obiektami i posiadają metody umożliwiające obliczanie m.in. punktów przecięć z innymi prymitywami.

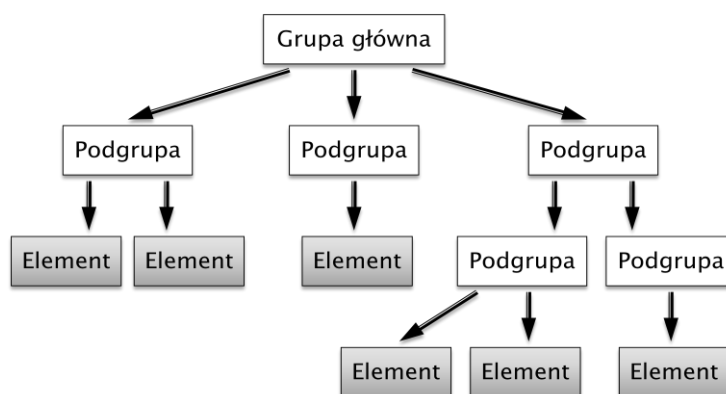
Wszystkie elementy opisujące model należą, w sensie programistycznym, do różnych klas i są dynamicznie tworzone podczas pracy aplikacji. Wskaźniki na nowopowstające obiekty są zapisywane na listach dwukierunkowych. Dzięki zastosowaniu dziedziczenia pokazanego poniżej wszystkie obiekty mogą być wspólnie tablicowane.



Rysunek 3. Hierarchia klas stosowanych do opisu modelu

Klasy takie jak PCBCore, PCBObject czy PCBPad są abstrakcyjne. Oznacza to, że nie mogą istnieć obiekty tych typów. Klasy abstrakcyjne zawierają metody, które po odziedziczeniu przez klasy pochodne muszą być zdefiniowane, aby można było tworzyć instancje takich klas. Przykładem metody wirtualnej jest rysowanie. Każdy obiekt jest rysowany w inny sposób. Dzięki zastosowaniu dziedziczenia metod wirtualnych, można na rzecz każdego z nich wywołać metodę rysowania, używając wskaźników typu PCBCore i za każdym razem uzyskać inny wynik.

Widoczne na ekranie aplikacji elementy mogą być łączone w grupy. Grupa jest obiektem posiadającym listę podgrup oraz listę elementów. Dzięki takiemu podejściu możliwe jest zapisanie drzewa o dowolnej liczbie poziomów.

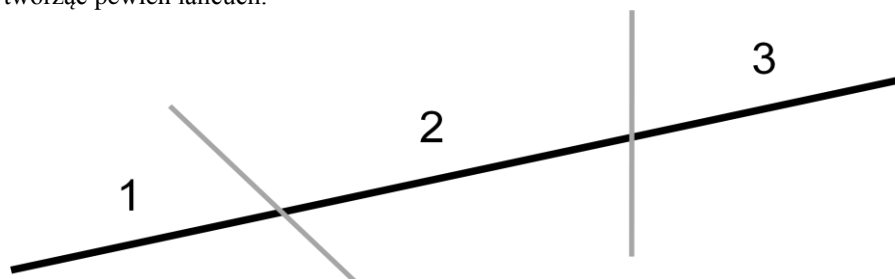


Rysunek 4. Schemat zapisu elementów z podziałem na grupy

4. Problematyka obliczeń

4.1. Wstęp

W poprzednim rozdziale pokazano wyniki pracy modułu CAM. Do obliczenia wspólnych konturów stosuje się wspomniane wcześniej prymitywy. Posiadają one nie tylko początek i koniec ale również zbiór punktów pośrednich powstających na skutek przecięć z innymi prymitywami. Po odpowiednim posortowaniu tych punktów prymityw może być podzielony na fragmenty, które łączą się ze sobą tworząc pewien łańcuch.



Rysunek 5. Przycinanie się prymitywów oraz podział na fragmenty

4.2. Opis linii

Linia do swojego opisu wymaga jedynie dwóch punktów które można interpretować jako początek i koniec. Kolejność tych punktów jest jednak dowolna. Dodatkowe punkty powstające na skutek przecięć z innymi prymitywami są sortowane wzdłuż kierunku linii i umieszczane na liście. W efekcie powstaje posortowany zbiór, w którym każde dwa kolejne punkty tworzą linię stanowiącą fragment pierwotnej. Zostało to pokazane na rysunku 5.

4.3. Opis łuku

Do opisu łuku, jak wspomniano wcześniej, można podejść na wiele sposobów. Do obliczeń zastosowano zapis, w którym znany jest środek łuku oraz jego promień. Punkty tworzące łuk można zapisać w postaci wektorowej, jako wektory łączące środek łuku z wybranym punktem. Każdy z takich wektorów tworzy pewien kąt z dodatnią półosią X. Zatem można zapisać jedynie zbiór wspomnianych kątów, ponieważ długości wektorów są równe promieniowi łuku. Stosując sortowanie zbioru kątów uzyskano analogiczny podział na fragmenty jak w przypadku linii.

Istotna jest tutaj kolejność zapisu początku i końca. Dwa punkty zapisane kątowno mogą reprezentować dwa różne łuki, dlatego trzeba zdecydować o kierunku łuku. W prezentowanym modelu obliczeniowym zastosowano kierunek przeciwny do ruchu wskazówek zegara, co jest zgodne z naturalnym pomiarem kąta względem dodatniej półosi X. Warto jeszcze zwrócić uwagę na fakt, że obliczane w tej metodzie kąty są z zakresu od -180 do $+180$, co podczas sortowania sprawia pewien problem. Przykładowo dla łuku znajdującego się w II i III ćwiartce układu współrzędnych, którego początek w zapisie kątowym wynosi 170° , a koniec -170° sortowanie w porządku rosnącym zamieniłoby miejscami początek i koniec.

Powstałby w ten sposób łuk po przeciwnej stronie, którego kąt rozwarcia wyniósłby 340° . W związku z tym należało przyjąć, że w zapisie kątowym każdy punkt łuku musi posiadać wartość większą niż punkt początkowy, co wymaga dodania kąta pełnego w prezentowanym przykładzie. Wówczas opisywany łuk rozpoczyna się w kącie 170° , a kończy w 190° . W obliczeniach kąty te zapisane są oczywiście w radianach, stopnie zostały tu zastosowane dla lepszego zobrazowania.

4.4. Obliczanie wspólnego konturu

Mając już narzędzie w postaci prymitywów można przystąpić do kluczowej części obliczeń. Każdy zamknięty kontur posiada pewną liczbę punktów wspólnych z dowolnym prymitywem. Jeśli ta liczba jest większa od zera, prymityw zostaje podzielony na fragmenty, które mogą znaleźć się w całości wewnątrz albo na zewnątrz tego konturu. Można to zobrazować rysując linię na mapie. Granice państw podzielią ją na fragmenty, z których każdy znajdzie się w całości w którymś z państw. Nie będzie takiego fragmentu który jednocześnie znajdzie się w dwóch państwach. Można zatem sprawdzić czy środkowy punkt fragmentu prymitywu należy do danego konturu. Jeśli należy, oznacza to, że powinien zostać usunięty, ponieważ przecina kontur. W przeciwnym wypadku fragment jest prawidłowy i powinien zostać uwzględniony w wyniku.

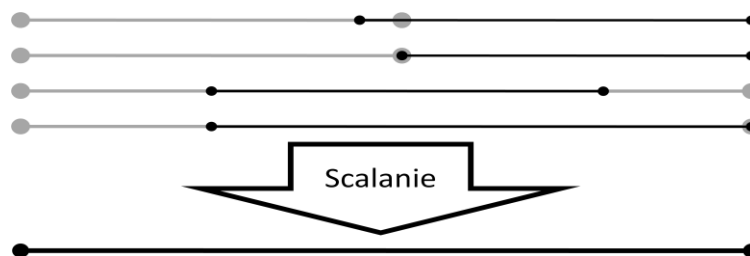
Po zastosowaniu powyższej metody dla wszystkich prymitywów tworzących model, powstaje pewien zbiór fragmentów. Aby można było wykonać grawerowanie potrzebne jest jeszcze ułożenie wspomnianych fragmentów w odpowiedniej kolejności, tak aby koniec poprzedniego pokrywał się z początkiem następnego. Jeśli odległość do kolejnego fragmentu jest zbyt duża oznacza to, że należy rozpocząć wykonywanie kolejnego konturu. Generator ścieżki narzędzia wyszukuje w takim przypadku pierwszy najbliższy niewykonany fragment i od niego rozpoczyna tworzenie kolejnego konturu.

4.5. Problem obliczeń zmiennie-przecinkowych

Prowadząc obliczenia komputerowe należy mieć na uwadze fakt, że wyliczenie tej samej wartości dwiema różnymi drogami nie da nigdy identycznego wyniku. Nie można zatem bezpośrednio porównywać tych liczb. Twórcy różnego rodzaju algorytmów często zapominają o tym, pisząc warunki równościowe. Oczywiście w sensie matematycznym liczby te są równe, ale sprawdzenie tej równości w programie komputerowym nie jest takie oczywiste z uwagi na błędy zaokrągleń. Problem ten można łatwo rozwiązać badając różnicę liczb. Jeśli różnica dwóch liczb jest dostatecznie mała oznacza to, że można je uznać za równe. Przyjmując zatem pewnego rodzaju skalę podobieństwa S , można powiedzieć że dwie liczby są sobie równe gdy wartość bezwzględna ich różnicy jest mniejsza lub równa S . Stosując tą metodę dalej, można wyprowadzić zależności umożliwiające skonstruowanie pozostałych operatorów: mniejszy, większy, mniejszy lub równy oraz większy lub równy. Przykładowo jeśli $a < b$ oznacza to, że $a - b$ musi być mniejsze od S . Wartość skali podobieństwa można dobrać dowolnie, zależnie od wymaganej dokładności operatorów porównania. Obliczenia prowadzone w oprogramowaniu stosują zamiennie S na poziomie 10^{-9} , 10^{-6} oraz 10^{-3} . Zmiana S jest czasem wymagana aby ustalić pewną hierarchię warunków porównania.

4.6. Scalanie prymitywów

Scalanie prymitywów w końcowym etapie wyznaczania ścieżki narzędzia jest bardzo istotne. Mowa tu o sytuacji, w której dwie linie lub dwa łuki posiadają część wspólną. Nie można wówczas liczyć ich punktów przecięć, ponieważ jest ich nieskończenie wiele. Ponadto istnieje problem zbudowania z nich ścieżki narzędzia, gdyż nie można ich wykonać jeden po drugim. Narzędzie wykonałoby jeden ruch roboczy następnie musiałoby się cofnąć i wykonać ruch po tej samej ścieżce. W szczególnym przypadku takie współliniowe prymitywy tworzą łańcuch. Również tutaj narzędzie będzie wykonywać niepożądane ruchy. W przypadku obróbki prowadzonej na robocie przemysłowym, dla uzyskania większej dokładności, następuje zatrzymanie po osiągnięciu zadanego punktu. Zatem wykonanie prostego odcinka w postaci serii fragmentów z wielokrotnym zatrzymaniem się narzędzia jest również pewnego rodzaju błędem. Generuje to również zbędny kod programu obróbki.



Rysunek 6. Możliwe przypadki scalania

5. Podsumowanie

Rozwój własnego środowiska CAD/CAM daje nieograniczone możliwości, jest jednak bardzo pracochłonny i wymaga rozwiązywania wielu złożonych problemów. Warto wspomnieć, że przedstawiony pakiet oprogramowania jest systemem klasy 2,5D. Rozbudowa pakietu do klasy 3D, z możliwością generowania programów na 5-cio osiowe obrabiarki, wymagałaby pracy wieloosobowego zespołu. Zaletą zaprezentowanego pakietu oprogramowania jest elastyczność, prosty interfejs oraz wygoda użytkowania. Dostęp do kodu źródłowego oprogramowania umożliwia rozbudowę pakietu o nowe funkcje i postprocesory.

LITERATURA

1. Serwis internetowy algorytm: <http://www.algorytm.org/geometria-obliczeniowa/wyznaczenie-punktow-przeciecia-okregu-z-prosta.html>, 2016.10.10.
2. Serwis internetowy obliczeniowo: <http://www.obliczeniowo.com.pl/?id=175>, 2016.10.10.
3. Serwis internetowy cplusplus - język c++: <http://www.cplusplus.com/reference/>, 2016.10.10.
4. Serwis internetowy wxWidgets: <http://www.wxwidgets.org/>, 2016.10.10.