

mgr inż. Daniel Reclik, dr hab. inż. Gabriel G. Kost, prof. Pol. Śl.
Instytut Automatykacji Procesów Technologicznych i Zintegrowanych Systemów
Wytwarzania Wydziału Mechanicznego Technologicznego Politechniki Śląskiej w Gliwicach

ZASTOSOWANIE METODY 2 ½ D W PLANOWANIU TRAJEKTORII ROBOTÓW MANIPULACYJNYCH

W pracy omówiono zastosowanie metody uproszczonej analizy przestrzeni 3D robotów manipulacyjnych. Analiza przestrzeni roboczej manipulatora konieczna jest do generowania bezpiecznych ścieżek ruchu. Autorzy zastosowali metodę 2 ½ D, której podstawą jest automatyczne generowanie kolejnych przekrojów sceny robota na algorytmicznie dobranych wysokościach. Przekroje są poddawane (zgodnie z przyjętą metodą) analizie płaskiej, przy czym wyniki są transponowane do postaci grafu przejść i poddawane działaniu algorytmu Floyda. Uzyskana w ten sposób pojedyncza ścieżka (o przejściach ortogonalnych) jest optymalizowana i wygładzana z zastosowaniem krzywych NURBS i B-Spline

THE 2 ½ D METHODE IN PLANNING PROCESS OF ROBOT TRAJEKTORY

In this paper there is presented the method of 3D manipulator's work-space analysis. The analysis of robot's work-space is necessary for generation the safety movement path. There was used 2 ½ D method, which is based on algorithm of following sections defining in robot work area. The heights of those sections are determining by another algorithm. Those sections are explored by using flat analysis, but the results are transposed into graph form. This graph is the record of all possible movements, so to get the optimum movement there must be used Floyd algorithm. This, shortest trace is optimized and smoothed by using NURBS and B-Spline curves.

1. WSTĘP

Technologiczne przygotowanie produkcji [1] obejmuje szereg istotnych zadań, których opracowanie jest niezbędne do uruchomienia nowego, bądź modernizacji istniejącego procesu technologicznego. Zadania te dotyczą ogólnego (koncepcja) oraz szczegółowego opracowania procesu technologicznego, a także niezbędnych do jego realizacji nowych rozwiązań konstrukcyjnych (narzędzia, magazyny, uchwyty itp.). W większości przypadków zagadnienia te rozwiązuje się w oparciu o techniki i narzędzia komputerowego wspomaganie prac inżynierskich, jakimi są systemy klasy CAx [1]. W ogólnym ujęciu systemy klasy CAx doskonale spełniają te wymagania. Jednakże w szczególnym ujęciu, możliwości poszczególnych systemów programowych grupy CAx bywają różne, a ich przydatność ograniczona. Szczególnym przykładem może być oprogramowanie z grupy CAD/CAM. W zakresie CAM, możliwości systemów programowania obrabiarek sterowanych numerycznie w pełni odpowiadają potrzebom technologii, zaś w zakresie programowania off-line robotów przemysłowych łatwość, szybkość i uniwersalność ich wykorzystania często jest ograniczona możliwością takiego systemu. Jednym z istotnych powodów takiego stanu jest konieczna do poprawnego zaprogramowania robota dokładna (a w wybranych miejscach szczegółowo przeprowadzona) analiza przestrzeni zadaniowej robota, dzięki której, technolog - programista robota może w sposób pewny wygenerować off-line bezpieczny, czyli bezkolizyjny ruch robota w jego przestrzeni zadaniowej.

2. PROBLEM PLANOWANIA TRAJEKTORII BEZKOLIZYJNEJ

Planowanie bezkolizyjnego ruchu robotów manipulacyjnych jest zadaniem trudnym i złożonym, i dotyczy rozwiązania szeregu szczegółowych zadań, począwszy od analizy przestrzeni wokółrobotowej 3D [2,3], której celem jest wyznaczenie obszarów wolnych od przeszkód, w których może być realizowany bezkolizyjny ruch robota, przez analizę kinematyki i algorytmów sterowania robotem [2,3,9], których celem jest wyznaczenie sparametryzowanego czasem równania drogi robota w przestrzeni (trajektoria ruchu robota), do możliwości praktycznego wykorzystania uzyskanego rozwiązania w systemach programowania off-line robotów [3]. Mimo wielu istniejących metod rozwiązujących te zagadnienia, ich praktyczne znaczenie jest dość ograniczone. Głównym powodem jest złożoność obliczeniowa istniejących metod, powodująca znaczną czasochłonność uzyskiwania ostatecznych rozwiązań. Dodatkowo, czasochłonność tę znacząco zwiększa konieczność dynamicznego modelowania trajektorii bezkolizyjnych wynikająca z niezbędnych modyfikacji ruchu robota w przestrzeni zadaniowej spowodowanych zachodzącymi w zrobotyzowanym, elastycznym systemie technologicznym zmianami wynikającymi z jego aktywności technologicznej (dostępność wybranych obszarów przestrzeni zadaniowej związana z koniecznością i niemożnością obsługi obrabiarek, zajętość magazynów, stanowisk reorientacji, wymianą chwytaków itp.). Źródłem tych trudności jest w głównej mierze trójwymiarowość przestrzeni zadaniowej robota manipulacyjnego i jej złożoność geometryczna, wynikająca z konfiguracji przestrzennej urządzeń technologicznych obsługiwanych przez niego.

Zagadnienie analizy przestrzeni, jako podstawowe, oraz problem wyznaczania bezkolizyjnych ścieżek robota w oparciu o wyznaczone podprzestrzenie bezkolizyjne przestrzeni zadaniowej, jest podstawą, dzięki której możliwe jest wyznaczanie poszukiwanej trajektorii bezkolizyjnej pomiędzy początkową i końcową pozycją robota wskazaną realizowanym zadaniem technologicznym.

2.1. Wyznaczenia bezkolizyjnej trajektorii robota

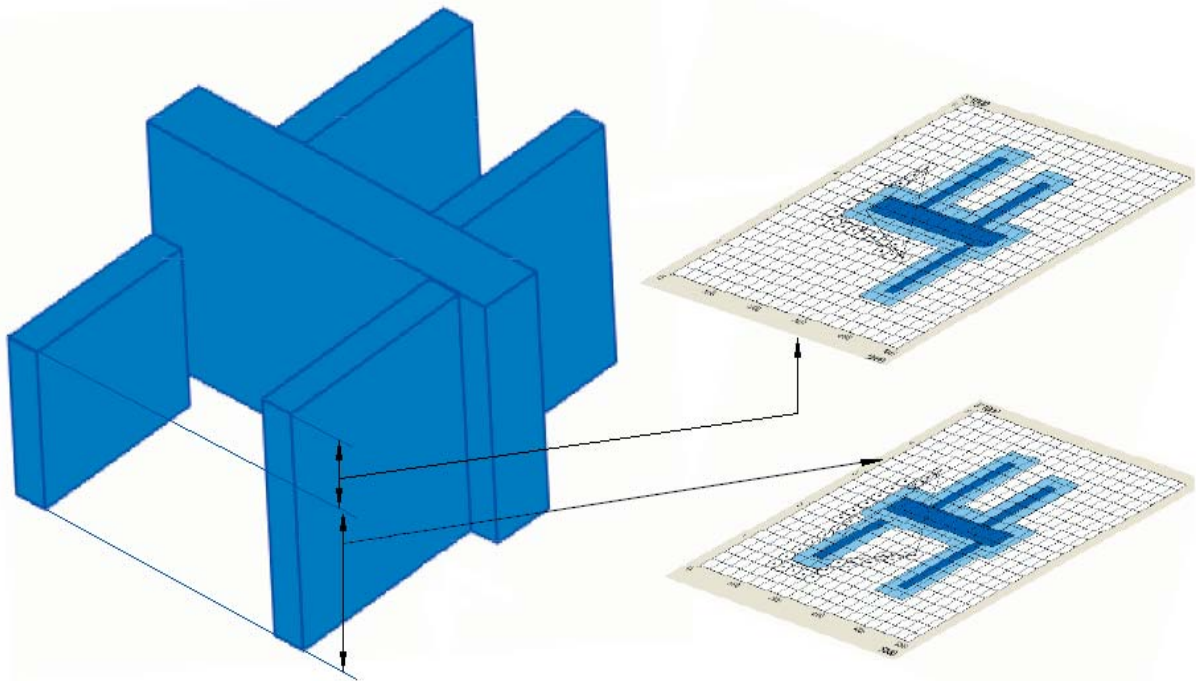
Wyznaczona w przestrzeni zadaniowej robota, w wyniku przeprowadzonego procesu planowania, bezkolizyjna ścieżka [1, 3] jest podstawą do realizacji kolejnego etapu planowania, jakim jest wyznaczenie sparametryzowanego czasem równania bezkolizyjnej drogi w przestrzeni zadaniowej 3D, czyli tzw. trajektorii bezkolizyjnej. Wyznaczoną bezkolizyjną ścieżkę robota w przestrzeni 3D określa zbiór tzw. punktów węzłowych, które wyznaczają łamaną w przestrzeni wg której robot powinien przejść z położenie początkowego do wskazanego końcowego. Równanie to musi być równaniem krzywej w przestrzeni spełniającej warunek ciągłości w kolejno wykorzystanych do jej zdefiniowania punktów węzłowych i musi spełniać warunek gładkości w sensie geometrycznym [4]. Warunek ten spełniają krzywe stopnia drugiego, tzw. krzywe klasy C^2 z ciągłym przyspieszeniem [2, 4, 10]. Krzywe te uzyskuje się metodami interpolacji zbioru punktów węzłowych wielomianami stopnia trzeciego. Do wykorzystywanych w zadaniu sterowania wielomianów zalicza się [6, 7, 8, 4]: wielomiany Bestseina-Béziara, NURBS, funkcje sklepane. Podstawowym problemem w definiowaniu przyjętego w zadaniu modelu krzywej ciągłej klasy C^2 jest trójwymiarowość (3D) definiowanego modelu geometrycznego, wynikająca z trójwymiarowości przestrzeni zadaniowej robota. W celu uproszczenia tego zadania autorzy proponują wykorzystanie modelu geometrycznego $2\frac{1}{2}D$, dzięki któremu możliwe będzie uproszczenie typowania kolejnych punktów węzłowych w zadaniu planowania.

3. ZAŁOŻENIA METODY 2 ½ D

W celu uproszczenia złożoności obliczeniowej zadania wyznaczania bezkolizyjnych ścieżek robota w przestrzeni 3D posłużono się, znanym z wcześniejszych generacji systemów sterowania maszyn CNC, mechanizmem interpolacyjnym 2½D [6]. W opisywanym podejściu metoda 2½D sprowadza się do implementacji metody 2D wraz z odpowiednim algorytmem sterującym umiejscawianiem analizowanej płaszczyzny w przestrzeni 3D (miejsce dokonywania odpowiedniego przekroju). Uproszczenie to ma na celu zastosowanie mechanizmu automatycznego generowania kolejnych przekrojów przestrzennej sceny robota, które w zestawieniu ze sobą tworzą postać przestrzenną analizowanego zagadnienia, zaś uzyskane płaszczyzny przekroju przestrzeni 3D rozpatrywane osobno pozwalają sprowadzić poszukiwanie bezpiecznych ścieżek do zagadnienia płaskiego 2D.

Aby możliwe było rozpatrywanie przestrzennej sceny otoczenia robota jako zbioru płaskich przekrojów muszą być zachowane pewne warunki. Warunkiem koniecznym jest zamodelowanie urządzeń technologicznych współpracujących z robotem, a tym samym mogących wywoływać kolizję, prostopadłościanami o podstawie kwadratu lub, co najwyżej prostokąta [3]. Zastosowane uproszczenie nie może być źródłem błędów procesu planowania, toteż ważnym aspektem pracy planeta trajektorii opartego na mechanizmie 2½D jest właściwe dobieranie kolejnych płaszczyzn przekrojów. Warto zauważyć, że nadmierna ilość przekrojów zwiększa niepotrzebnie ilość obliczeń (każdy dodatkowy przekrój generuje cały tok obliczeń płaskiego obszaru ruchu kiści robota). Aby z kolei zapewnić możliwość uzyskiwania optymalnych pod względem długości dróg przejścia konieczne jest rozpatrywanie wszystkich przekrojów, które różnią się ilością przecinanych urządzeń. Ponieważ teoretycznie każde urządzenie ma swoją wysokość, a zatem, gdyby chcieć wykonywać kolejne przekroje od płaszczyzny podstawy robota w górę, okazałoby się, że najprostszą drogą będzie ominięcie górą wszystkich urządzeń sceny robota i przejście po linii prostej nad nimi. Ponieważ nie zawsze jest możliwe podniesienie kiści robota na odpowiednią wysokość, zatem kolejnym warunkiem, jaki musiał być zawarty w tworzonym oprogramowaniu był warunek brzegowy maksymalnej wysokości ruchu kiści robota. Przyjęto więc, że poprawnym rozwiązaniem będzie generowanie kolejnych przekrojów sceny dla odpowiednich zakresów wysokości. Aby zapewnić bezpieczeństwo ruchu koniecznym było przyjęcie strefy bezpieczeństwa [3] wokół każdej maszyny w celu zagwarantowania odpowiednio bezpiecznego omijania kolejnych przeszkód. Zatem przyjęto zasadę podziału wysokości maksymalnej ruchu na zakresy, których granice wyznaczają kolejne wysokości maszyn otoczenia technologicznego powiększone o wartość ekwidystanty bezpieczeństwa. Podziały wykonywane są kolejno od dołu, tzn. pierwszy zakres obejmuje wysokości od zera (płaszczyzna podstawy robota) do wysokości najniższej z obrabiarek (lub jej elementu), powiększonej o przyjętą strefę bezpieczeństwa. Kolejny zakres obejmuje wysokości większe od wysokości maszyny najniższej do wysokości kolejnej, wyższej maszyny również powiększonej o wartość strefy bezpieczeństwa. W ten sam sposób dzielone są kolejne części pozostałego zakresu wysokości maszyn (aż do wysokości granicznej - maksymalnej wysokości ruchu kiści robota). Ponieważ w ten sposób określone zostają wszystkie wymagane w metodzie przekroje sceny robota, toteż złożoność obliczeniowa metody zależy wyłącznie od zróżnicowania wysokości obiektów otoczenia technologicznego robota i przyjętej wysokości granicznej ruchu kiści manipulatora. Ponieważ płaski planer trajektorii bezpiecznych wyznacza punkty podporowe przejścia o optymalnej w sensie geometrycznym długości ruchu (najkrótszej), toteż na każdym z przekrojów danego zakresu wysokości musi wskazać tą samą ścieżkę ruchu. Zatem uzyskanie informacji o przejściu bezpiecznym na

danej wysokości zakresu (współrzędne x , y , z punktów podporowych) wymaga jedynie przekształcenia punktów wzorcowych wyznaczonych na przekroju (aby uzyskać punkty na zadanej wysokości przekształcamy punkty wzorcowe – na przekroju – na nowe punkty w danym zakresie wysokości o tych samych współrzędnych x , y ze zmianą współrzędnej z). Zasadę działania podprogramu generującego przekroje pokazano na rys. 1.



Rys.1. Otoczenie robota wraz z wygenerowanymi przekrojami (po wykonanej analizie ruchu po przekroju)

W przedstawionym przykładzie (rys. 1), analiza topografii sceny robota przy przyjętym warunku maksymalnej wysokości ruchu końcówki robota, wykazała występowanie 2 różnych wysokości maszyn, więc w wyniku działania metody uzyskano dwa przekroje sceny robota (na rys. 1) z zaznaczonymi punktami startowym i końcowym ruchu o współrzędnych x i y (rzeczywista współrzędna z jest korygowana do wysokości położenia przekroju).

4. GRAF BEZPIECZNYCH PRZEJŚĆ W PRZESTRZENI ROBOCZEJ MANIPULATORA

Aby możliwe było zastosowanie metody $2\frac{1}{2}D$ wraz z całym mechanizmem generowania przekrojów do generowania przestrzennych krzywych ruchu robota w przestrzeni zadaniowej musi być w pierwszej kolejności wygenerowany, w oparciu o dane z przekrojów, graf bezpiecznego przejścia. Graf ten tworzą punkty węzłowe ścieżki bezkolizyjnej stanowiące dane wejściowe do omawianej metody, dla których modyfikuje się wartość współrzędnej Z , tak by pozwalały umieścić wszystkie punkty węzłowe ścieżki znajdujące się w analizowanym przekroju na wyznaczonej płaszczyźnie podziału. Ważnym aspektem tego etapu tworzenia grafu przejść jest dobranie właściwego przekroju dla punktu początkowego i dla punktu końcowego ruchu. Oczywiście punkty te mogą leżeć na różnych wysokościach, a ponadto ich wysokości mogą leżeć w różnych zakresach przekrojów. Uzyskane w ten

sposób zmodyfikowane punkty podporowe mogą być połączone w graf bezpiecznych przejść. W celu poprawienia szczegółowości przeprowadzonej analizy, postanowiono dodatkowo wypełnić graf przejść dodatkowymi punktami podporowymi uzyskanymi drogą kolejnych przekrojów sceny robota. Działanie to pozwoliło uzyskać bardziej liczny zbiór punktów podporowych, dzięki czemu możliwe było przeprowadzenie optymalizacji przejścia przez graf zarówno pod kątem „płynności ruchu robota” (ścieżka jest łamana), najkrótszej drogi – jak i pod względem najszybszego przejścia robota do położenia docelowego. Na rys. 2 pokazano wygenerowany w oparciu o wykonane przekroje graf bezpiecznych przejść. Warto zauważyć, że wierzchołki grafu tworzą środki rastrowe w oparciu o które wygenerowano bezkolizyjną ścieżkę robota (metody rastrowe analizy przestrzeni zadaniowej robota [3,5]. Dla lepszego zilustrowania na rys. 2 punktów wyznaczonych w oparciu o dokonane przekroje (punkty właściwe) zaznaczono je kolorem błękitnym, zaś czerwonym oznaczono punkty dodatkowe. Aby pokazać wyniki obliczeń mających na celu wytypowanie kolejnych przejść na każdym z przekrojów z prawej strony rys. 2 pokazano wyniki działania planera 2D dla każdego z nich. Opierając się na tak przygotowanych danych, w kolejnym etapie jest optymalizacja przejścia przez graf z zastosowaniem jednej z metod grafowych. Zastosowano w tym celu metodę Floyda [4, 10] znajdowania najkrótszej drogi w grafie o nieujemnych wagach naroży. Aby ograniczyć tzw. pesymistyczną złożoność obliczeniową algorytmu postanowiono zastosować funkcję rozpoznawania i zakazu ruchu.

4.1. Algorytm przeszukiwania grafu oparty na algorytmie Floyda

Algorytm, który został zastosowany do rozwiązania problemu wyznaczenia optymalnego przejścia robota w otoczeniu oblicza wszystkie drogi, tzn. sumy wag krawędzi łączących bezpośrednio lub pośrednio wszystkie pary wierzchołków grafu przejść. Aby uzyskać i zapamiętać podczas obliczeń wyniki konieczne jest stworzenie tablicy o rozmiarze $n \times n$, gdzie n oznacza ilość wierzchołków grafu. W związku z tym, graf najlepiej przedstawić w postaci macierzy sąsiedztwa. Macierz sąsiedztwa [4, 10] jest tablicą dwuwymiarową o rozmiarze n^2 . Kolejne wiersze i kolumny macierzy sąsiedztwa reprezentują wierzchołki grafu, natomiast wartość na przecięciu *i-tego* wiersza i *j-tej* kolumny tej macierzy to waga krawędzi łączącej wierzchołki (waga przejścia z punktu o numerze *i* do punktu o numerze *j*). Aby zachować oryginalne wartości wag krawędzi należy zdefiniować drugą tablicę o takim samym rozmiarze. Będzie to tablica robocza, w której po zakończeniu pozostaną wyniki. Aby umożliwić odtworzenie sekwencji kolejnych wierzchołków tworzących drogi przejścia pomiędzy wierzchołkami potrzebna jest jeszcze jedna tablica, tzw. macierz poprzedników (predecessor matrix). Macierz poprzedników musi być zainicjowana numerami wierzchołków początkowych. Macierz poprzedników jest wykorzystana do określenia możliwości bezpiecznego poruszania się robota pomiędzy poszczególnymi punktami grafu przejść. Taka organizacja danych pozwala na automatyczne sortowanie analizowanych punktów (podporowych). Ponieważ jako wagi przyjęto rzeczywiste odległości pomiędzy punktami wg wzoru:

$$L_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

gdzie: x, y, z to kolejne współrzędne punktów podporowych trajektorii w grafie przejścia.

Działania algorytmu Floyda w warunkach omawianego zadania można zdefiniować za pomocą pseudokodu w następujący sposób:

dla k równego 0 do $n-1$ zrób:

{

dla i równego 0 do $n-1$ zrób

{

dla j równego 0 do $n-1$ zrób

{

jeżeli istnieje droga pomiędzy i oraz k przy jednoczesnym istnieniu drogi pomiędzy k oraz j to zrób:

{

jeżeli nie istnieje droga pomiędzy i oraz j to przyjmij długość ruchu pomiędzy i oraz j jako sumę dróg $i-k$ oraz $k-j$

jeżeli istnieje droga $i-j$ to przyjmij długość ruchu pomiędzy i oraz j jako minimum dróg: $i-k + k-j$ lub $i-j$

aktualizuj macierz przejść – oznacz drogę $i-j$ o wyliczonej długości jako możliwą do przejścia

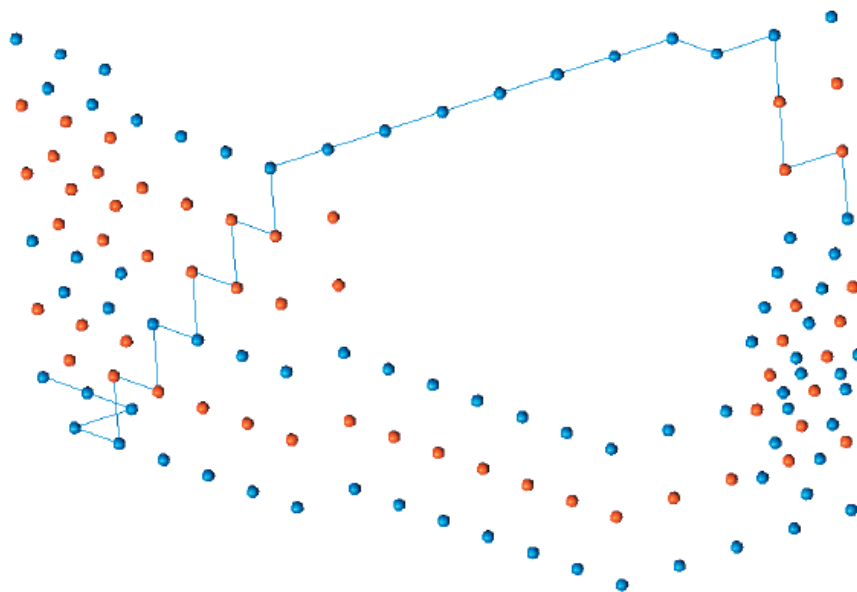
}

}

}

}

Wynikiem działania algorytmu (rys. 3) jest wyznaczenie minimalnej długości drogi pomiędzy każdą parą wierzchołków grafu. Ponieważ w powyższym zapisie zawarto dodatkową macierz zapisu trasy, toteż w wyniku działania algorytmu, oprócz wartości długości najkrótszego ruchu pomiędzy każdą parą wierzchołków, uzyskano dodatkowo zapis sekwencji przejścia w postaci kolejności wierzchołków grafu przejść. Po zakończeniu wyliczeń następuje ponowna kompozycja macierzy przejść w oparciu o numery dekomponowanych punktów (występujących w najkrótszym przejściu pomiędzy zadanymi punktami). Wynik działania algorytmu pokazano na rys. 3.



Rys. 3. Wynik działania algorytmu Floyda przeszukiwania grafu przejść robota

5. OPTIMALIZACJA ROZWIĄZANIA GRAFU PRZEJŚĆ W PRZESTRZENI 3D

Zastosowana metoda miała za zadanie zminimalizować złożoność obliczeń wymaganych do generowania przestrzennych, bezpiecznych ścieżek ruchu robotów manipulacyjnych. Koniecznym okazało się więc opracowanie odpowiedniego modułu optymalizującego liczbę uzyskanych w wyniku dokonywanych podziałów punktów węzłowych (punkty podporowe trajektorii wyznaczone z grafu przejść). Z założenia, wynikowa trajektoria musi mieć gładkość nie tylko w sensie geometrycznym, ale musi być spełniony warunek ciągłości 2 pochodnej poszukiwanej funkcji drogi (funkcji klasy C^2). Podstawowym narzędziem zastosowanym przez autorów do realizacji interpolacji krzywej płaskiej i przestrzennej, w oparciu o uzyskane w procesie analizy geometrycznej punkty, są funkcje B-sklejane, a w postaci wymiernej NURBS [2, 3, 5, 6, 7, 8].

5.1. Algorytm generowania krzywych interpolujących

Krzywe interpolujące oparte na wielomianach Bersteina-Béziara, jak i też na NURBS i B-splajnach są funkcjami złożonymi. Podstawą generowania tych krzywych są funkcje bazowe, których odpowiednie sklejenie ze sobą daje w rezultacie zamierzony kształt przestrzennej krzywej o zadanych parametrach. Ponieważ funkcje bazowe są określone w dziedzinie czasu, toteż powstała z ich sklejenia krzywa również jest pośrednio określona w dziedzinie czasu. Zatem aby móc wykreślić krzywą interpolującą ruch robota po wyznaczonej ścieżce bezkolizyjnej należy w pierwszym etapie przewidzieć całkowity czas trwania ruchu. Czas ten w dalszej części przyjmowany jest jako umowna jednostka podstawowa, tzn. ruch trwa w umownym przedziale czasowym od 0 do 1, czyli od zera do 100 % rzeczywistego czasu. Dalsze rozważania nad procesem interpolacji dotyczą krzywych NURBS [2, 8]. W procesie tworzenia wielomianu interpolującej funkcji giętkiej konieczne jest wyznaczenie węzłów interpolacji. Ponieważ ruch po krzywej trwać ma od 0 do 1, toteż węzłami interpolacji będą przedziały czasowe z tego zakresu. Aby przyjąć właściwą liczbę węzłów interpolacji musimy ustalić stopień n wielomianu interpolacyjnego oraz poznać liczbę punktów podporowych interpolacji Li . Matematycznie zbiór węzłów można zapisać w następujący sposób [7]:

$$T=\{t_0, t_1, t_2, t_3, t_4, t_5, \dots, t_k\} \quad (5.1)$$

przy czym ilość węzłów t_i musi być równa liczbie U [7]:

$$U=Li+n+1 \quad (5.2)$$

Pierwsze $n+1$ węzłów należy wypełnić wartościami 0, ostatnie $n+1$ węzłów wartościami 1, natomiast pozostałe węzły wypełniane są proporcjonalnie rosnącymi wartościami czasu. Przykładowy zbiór węzłów interpolacji dla krzywej stopnia 3-go i 7 punktów podporowych ma postać [7]:

$$T=\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\} \quad (5.3)$$

Mając wyznaczone punkty węzłowe interpolacji (punkty T na osi czasu – wzór 5.3) można wyznaczyć bazowe funkcje składowe wymiernego wielomianu B-sklejanego (NURBS). Wyznaczanie funkcji bazowych jest procesem iteracyjnym, ponieważ aby wyznaczyć funkcje bazowe danego stopnia konieczne jest wyznaczenie wcześniej funkcji bazowych stopnia niższego. W związku z tym cały proces rozpoczyna się od wyznaczenia funkcji bazowych stopnia zerowego, które definiowane są następująco: dla zadanego przedziału czasu są równe 1, natomiast dla pozostałych okresów czasu przyjmują wartości 0, co można zapisać [7]:

$$\begin{aligned}
 N_{i,0} &= 1 \text{ dla chwili } t \text{ należącej do przedziału czasu pomiędzy węzłami } t_i \text{ oraz } t_{i+1} \text{ oraz:} \\
 N_{i,0} &= 0 \text{ dla pozostałego czasu } t
 \end{aligned}
 \tag{5.4}$$

Funkcje bazowe stopni wyższych wyliczane są w oparciu o następującą zależność [7]:

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t)
 \tag{5.5}$$

gdzie:

p przyjmuje kolejno wartości: 1, 2, 3 aż do wartości stopnia wielomianu n

i przyjmuje kolejne wartości od 0 aż do ilości przedziałów pomiędzy węzłami $Li+n$

Mając wyliczone wartości funkcji bazowych można wyznaczyć współrzędne punktów wielomianu interpolującego $X(t)$, $Y(t)$, $Z(t)$ jako superpozycji wartości poszczególnych współrzędnych układu kartezjańskiego. W przypadku konwencjonalnego wielomianu B-sklejanego wartości współrzędnych liczone są jako suma iloczynów funkcji bazowych i wartości współrzędnych punktów podporowych P , co można zapisać jako [2, 7]:

$$X(t) = \sum_j N_{j,n}(t) P_{x_j} \quad Y(t) = \sum_j N_{j,n}(t) P_{y_j} \quad Z(t) = \sum_j N_{j,n}(t) P_{z_j}
 \tag{5.6}$$

Dla wyliczenia wymiernego wielomianu B-sklejanego (NURBS) musimy uwzględnić ponadto wagi punktów podporowych P oznaczone jako w . Znając wagi punktów podporowych można wyliczyć poszczególne współrzędne wielomianu interpolującego wykorzystując wzory [2, 7]:

$$X(t) = \frac{\sum_j N_{j,n}(t) P_{x_j} w_j}{\sum_j N_{j,n}(t) w_j}
 \tag{5.7}$$

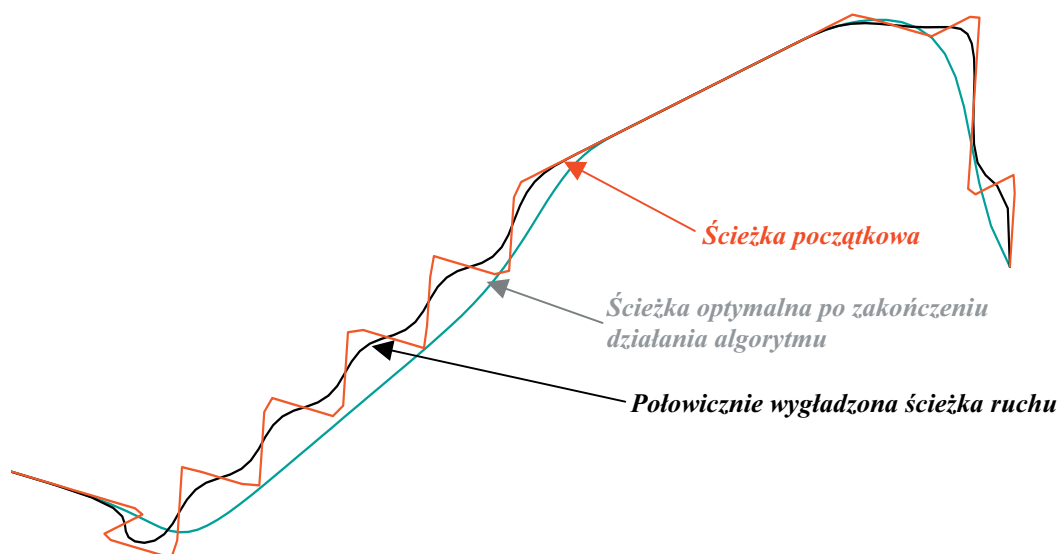
$$Y(t) = \frac{\sum_j N_{j,n}(t) P_{y_j} w_j}{\sum_j N_{j,n}(t) w_j}
 \tag{5.8}$$

$$Z(t) = \frac{\sum_j N_{j,n}(t) P_{z_j} w_j}{\sum_j N_{j,n}(t) w_j}
 \tag{5.9}$$

Zwiększanie stopnia wielomianu interpolującego powoduje również zmniejszenie przegięć krzywej, co w rezultacie daje krzywą o bardziej łagodnych zaokrągleniach.

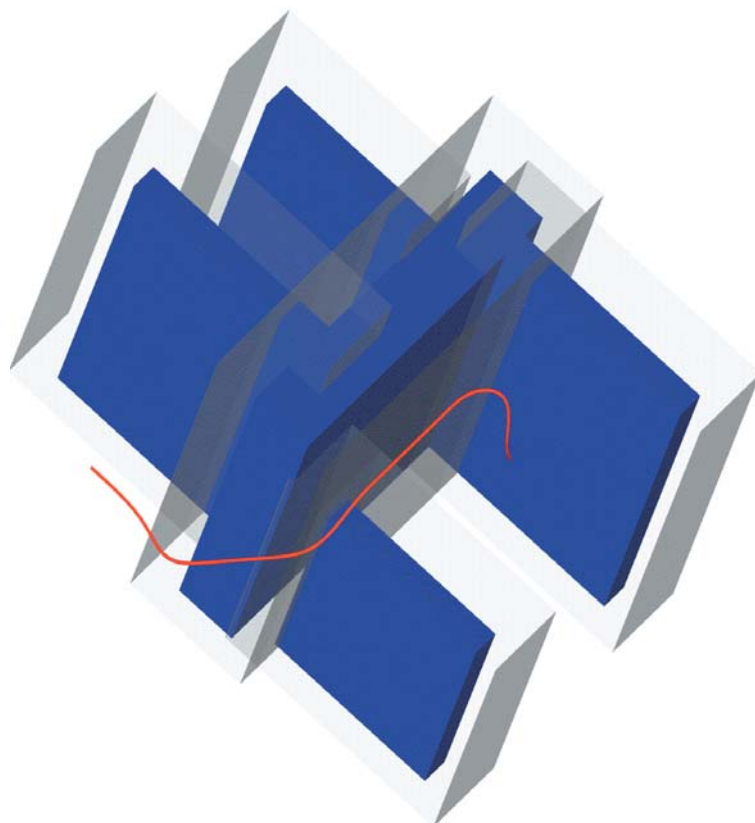
5.2. Wynik działania algorytmu

Ponieważ proces optymalizowania trajektorii przestrzennych, ze względu na zastosowane narzędzie i opracowane wcześniej algorytmy posiada taką samą pesymistyczną złożoność obliczeniową jak optymalizowanie płaskich przestrzeni ruchu, toteż może być z powodzeniem stosowany do wspomaganie programowania off-line robotów przemysłowych. Uzyskiwane wyniki w postaci przestrzennej krzywej NURBS można dyskretyzować, co oznacza, że można ustalić liczbę wygenerowanych punktów utworzonej funkcji. Podobnie na płaszczyźnie, proces wygładzania przebiega iteracyjnie z każdorazowym sprawdzaniem poprawności obliczeń (bezkolizyjności ruchu). Na rys. 4 pokazano jak zmienia się optymalizowana trajektoria w toku obliczeń.



Rys. 4. Widok kolejnych faz wygładzania ruchu w przestrzeni zadaniowej 3D

Na rys. 5 pokazano zaś wygenerowaną, zoptymalizowaną trajektorię ruchu robota w przestrzeni zadaniowej. Zaznaczono przyjętą strefę bezpieczną w postaci półprzezroczystych brył nałożonych na prostopadłościany aproksymujące rzeczywisty kształt przeszkód robota. Kolorem niebieskim oznaczono właściwe wymiary przeszkód robota (maszyn), a kolorem czerwonym oznaczono uzyskaną optymalną i bezkolizyjną ścieżkę ruchu.



Rys. 5. Widok sceny robota wraz z naniesioną wygenerowaną ścieżką bezpiecznego ruchu o gładkości C^2

Wynik działania metody pokazany na rys. 5 reprezentuje postać graficzną funkcji o gładkości C^2 , reprezentującej poszukiwaną trajektorię bezpieczną robota. Warto podkreślić, że uzyskane rozwiązanie powstało w oparciu o optymalizację długości ruchu robota

6. PODSUMOWANIE

Dzięki zastosowanym rozwiązaniom algorytmicznym uzyskano w pełni zadowalający przestrzenną drogę robota mimo, że całe obliczenia sprowadzone zostały do zagadnienia płaskiego, reprezentowanego przez graf przejść w oparciu o który wygenerowano płaskie przekroje sceny robota. Przedstawiony w niniejszej publikacji system poszukiwania bezkolizyjnych ścieżek ruchu robotów w ich przestrzeniach zadaniowych nie został jeszcze w pełni ukończony. Trwają prace nad dalszą rozbudową systemu, a w szczególności nad opracowaniem w pełni funkcjonalnego systemu, który mógłby być przyłączany do jednego z istniejących systemów programowania off-line robotów przemysłowych. Opracowane rozwiązanie analizy przestrzeni wokółrobotowej 3D i jej upraszczania techniką $2 \frac{1}{2} D$ z zastosowaniem metod grafowych pozwala na znaczne uproszczenie i ograniczenie liczby iteracji obliczeniowych potrzebnych do wygenerowania równania trajektorii robota manipulacyjnego.

LITERATURA

1. Chlebus E.: techniki komputerowe CAx w inżynierii produkcji. WN-T, Warszawa 2000.
2. Dulęba I.: Metody i algorytmu planowania ruchu robotów mobilnych i manipulacyjnych. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001.
3. Kost G.G. Planowanie bezkolizyjnych ścieżek manipulacyjnych i stacjonarnych robotów przemysłowych oparte na procesach decyzyjnych Markowa i funkcji ocen. Wydawnictwo Politechniki Śląskiej. Gliwice 2004.
4. Demaine E., Gweightwasser S.: Introduction to Algorithms. MIT. 2002.
5. Morecki A., Knapczyk J.: Podstawy robotyki. Teoria i elementy manipulatorów i robotów. WN-T. Warszawa.
6. Kozłowski K., Dutkiewicz P., Wróblewski W.: Modelowanie i sterowanie robotów. Wydawnictwa Naukowe PWN. Warszawa 2003.
7. Fortuna Z., Macukow B., Wąsowski J.: Metody numeryczne. s. Podręczniki Akademickie. Elektronika. Informatyka. Telekomunikacja. Wydawnictwa Naukowe PWN. Warszawa.
8. Majchrzak E., Mochnacki B.: metody numeryczne. Podstawy teoretyczne, aspekty praktyczne i algorytmy. Wydawnictwo Politechniki Śląskiej, Gliwice 2004.
9. Latombe J.-C.: Robot motion planning. Kluwer Academic Publishers. Boston/London. 1993.
10. Rosen K.H.: Discrete Mathematics and Its Applications, 2nd edition. McGraw-Hill Publishing. McGraw-Hill. 1991.