

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

## **ZASTOSOWANIE GRAFIKI 3D I BIBLIOTEK OPENGL DO SYMULACJI STANÓW KOLIZYJNYCH ROBOTA**

*W pracy omówiono aplikację PLANER będącą praktyczną implementacją metody planowania ruchu robotów manipulacyjnych. Autorzy zastosowali metodę 2½ D, do wyznaczania bezpiecznych trajektorii ruchu robotów, której podstawą działania jest płaska analiza kolejnych przekrojów sceny robota. Opracowana aplikacja PLANER realizuje obliczenia, dokonuje interpolacji wynikowej trajektorii ruchu z wykorzystaniem krzywych B-Spline, jak również pozwala na ocenę wizualną wyników dzięki przestrzennej prezentacji wyników obliczeń z wykorzystaniem grafiki OpenGL. Ze względu na znaczną ilość dostępnych funkcji opracowanego systemu, w pracy postanowiono omówić jedynie najistotniejsze z punktu widzenia praktycznego jej zastosowania. Omówione zostały mechanizmy obliczeniowe, system symulacji ruchu z wykrywaniem w czasie rzeczywistym kolizji typu face-to-face, interfejs użytkownika systemu PLANER, jak również moduł eksportu programów roboczych do systemu programowania robotów off-line FANUC Roboguide. Pracę zakończono opisem przeprowadzonych testów na rzeczywistych manipulatorach oraz uzyskanymi na ich podstawie wnioskami.*

### **THE 3D GRAPHICS BASED ON THE OPENGL LIBRARY IN SIMULATIONS OF THE ROBOT'S COLLISION STATE**

*In this paper there is presented the PLANER software, which is the practical implementation of the collision free robot movement planning method. There was used 2½ D method, which is based on algorithm of following sections defining in robot work area. Those sections are explored by using flat analysis. Described application is able to determine collision free paths, make interpolation of those with the B-Spline curves algorithm and allows user to optical check the results by the 3D perspective image generated by using of the OpenGL graphics library. Because the PLANER software has many properties, we decided to describe only the most important and useful in practical usage of those application. There are described the calculations algorithms, module of the real time simulations of the movement along determined path, module of collision detect in face-to-face mode, application GUI, and the export module, which allow to transfer results into FANUC Roboguide robots off-line programming system. At the end of this paper there are conclusions, which are result of practical tests the PLANER software.*

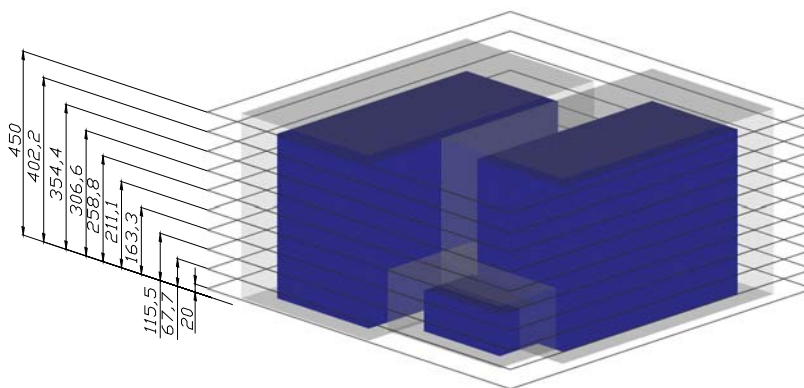
#### **1. WPROWADZNIENIE**

Ze względu na wysoki koszt urządzeń technologicznych w elastycznych systemach wytwarzania można zaobserwować coraz szersze stosowanie środków komputerowego wspomaganie procesu przygotowania produkcji. Wykorzystanie możliwości współczesnych komputerów oraz odpowiedniego oprogramowania wspomagającego pozwala na zwiększenie wydajności pracy systemu produkcyjnego, co powoduje wzrost atrakcyjności cenowej gotowego wyrobu. Ponadto dzięki zastosowaniu komputerowego wspomaganie procesu przygotowania programów obróbczych, w tym także sekwencji ruchu manipulatorów (systemy off-line), możliwe jest przygotowanie produkcji nowego przedmiotu bez wyłączania z produkcji linii technologicznych, co pozwala maksymalizować wydajność urządzeń i tym

samym obniża cenę końcową wyrobu. W grupie środków wspomagających prace z zakresu programowania maszyn wytwórczych, systemy komputerowego programowania off-line robotów przemysłowych są najmniej zaawansowane technologicznie [2]. Szczególnie słabo rozwinięty w tych systemach jest obszar zagadnień dotyczących planowania zadań, a w tym obszarze – wyznaczanie dróg bezkolizyjnych przejść robota pomiędzy obiektami technologicznymi [2, 4] znajdującymi się w jego przestrzeni zadaniowej. Brak skutecznego narzędzia pozwalającego na automatyzację pracy programisty, poprzez automatyczne generowanie w skończonym czasie wysokiej jakości ścieżek ruchu sprawia, że podczas programowania ruchów manipulatora programista musi każdorazowo przewidzieć możliwość wystąpienia kolizji, przez co proces przygotowania programu ruchu znacząco się wydłuża. Wprowadzenie oprogramowania narzędziowego pozwalającego na automatyczne wygenerowanie trajektorii omijającej wszelkie obiekty otoczenia technologicznego pozwoliłoby na znaczne usprawnienie pracy podczas generowania zadań manipulatorów, jak również na poprawienie jakości wygenerowanego programu. Brak algorytmizacji w procesie podejmowania decyzji o przyjętym torze ruchu narzędzia powoduje, że o jakości programu roboczego decyduje przede wszystkim doświadczenie programisty.

## 2. GENEZA POWSTANIA SYSTEMU PLANER

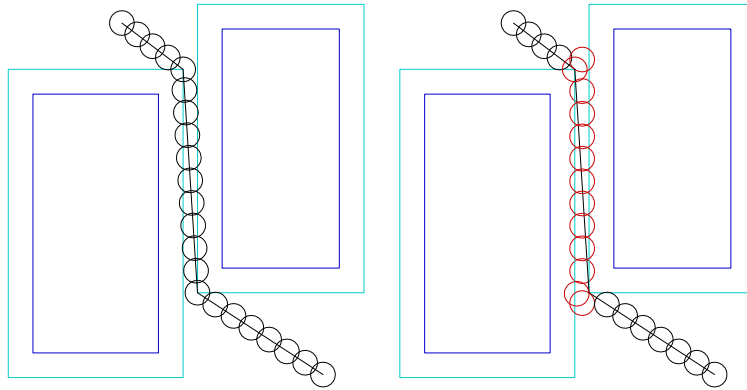
Brak dostępności na rynku oprogramowania pozwalającego na automatyczne generowanie trajektorii ruchu robotów przemysłowych spowodował rozpoczęcie w Instytucie Automatykacji Procesów Technologicznych i Zintegrowanych Systemów Wytwarzania prac nad utworzeniem takiego systemu. Pierwsza postać algorytmu planowania ruchu robotów została opracowana w 2008 r. Główną ideą opracowanego algorytmu obliczeniowego [7-11] było generowanie kolejnych przekrojów przestrzennej sceny robota, dzięki czemu wytyczanie bezpiecznej ścieżki ruchu ograniczone było każdorazowo do zadania płaskiego. Przykładowy rozkład przekrojów sceny pokazano na rys. 1. Wynikowe zbiory punktów podporowych pozwalały na wygenerowanie grafu możliwych przejść, którego optymalizacja pozwalała na wykonanie interpolacji krzywymi B-Spline [2, 5], dzięki czemu końcowa postać trajektorii zapewnia utrzymanie gładkości klasy  $C^2$ [5].



Rys. 1. Przykładowy rozkład przekrojów sceny

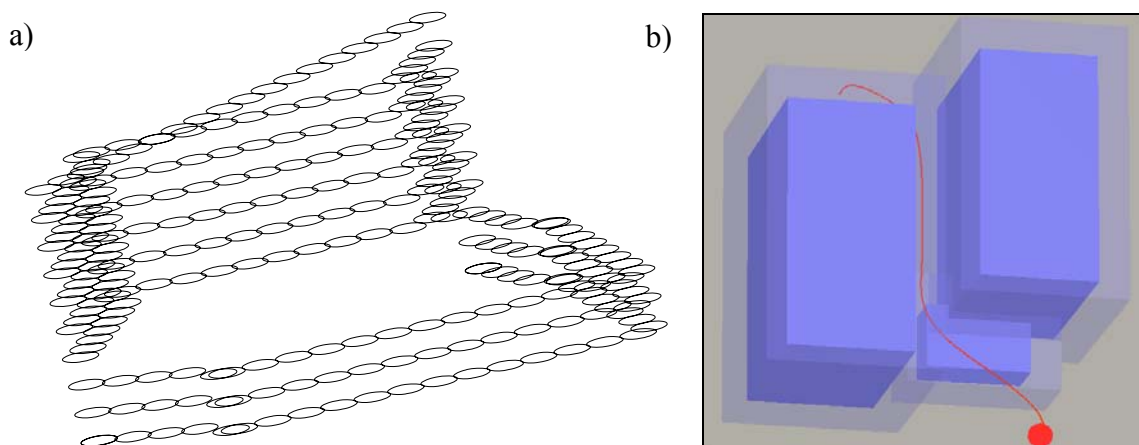
Gęstość rozkładu kolejnych przekrojów sceny uzależniona została od wielkości przemieszczanego przedmiotu, który na potrzeby zadania planowania ruchu aproksymowany jest za pomocą opisanej na nim kuli. Wprowadzana przez użytkownika wartość średnicy kuli opisanej na przedmiocie stanowi jednocześnie wielkość stref bezpieczeństwa, przez co ma znaczący wpływ na przeprowadzaną analizę danego przekroju. Przekroje analizowane są od najwyżej położonych w kierunku płaszczyzny sceny. Każdy z przekrojów analizowany jest

przy wykorzystaniu metody grafu widoczności. Powstała na skutek analizy przekroju sieć połączeń jest następnie optymalizowana algorytmem Floyd-Warshalla, celem uzyskania najkrótszego przejścia danego przekroju. Kolejnym etapem obliczeniowym jest dyskretyzacja wynikowego grafu, oraz modyfikacja położenia powstałych punktów podporowych metodą pól potencjałowych o ograniczonym zasięgu działania. Wynikowy zbiór punktów podporowych dla przykładowego przekroju pokazano na rys. 2.



Rys. 2. Wynik realizowanej przez opracowany system PLANER analizy przykładowego przekroju sceny

Zgodnie z przyjętą metodą przestrzennego planowania ruchu robotów, aplikacja PLANER, po przeprowadzeniu analizy wszystkich przekrojów danej sceny, generuje pokazany na rys. 3a graf możliwych przejść (suma wyników analiz wszystkich przekrojów), który w dalszej części jest optymalizowany pod względem minimalnej długości ruchu. Wynikowy zbiór punktów podporowych stanowi podstawę wykonywanej interpolacji B-Spline, której końcową postać pokazano na rys. 3b.

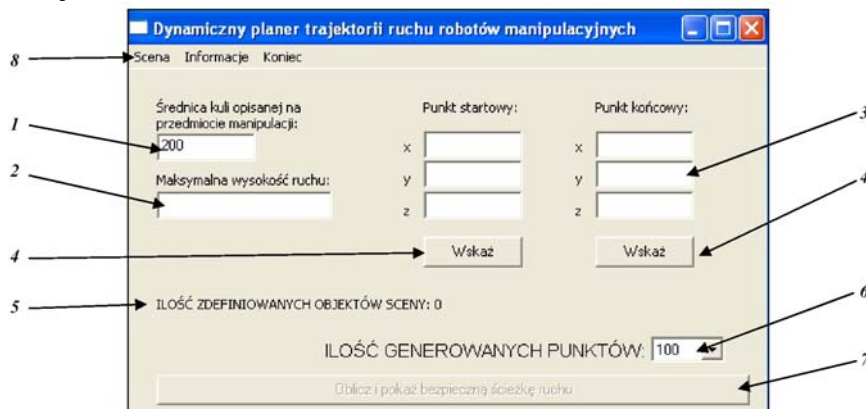


Rys. 3. Graf możliwych przejść oraz wynikowa interpolowana trajektoria ruchu – aplikacja PLANER

### 3. INTERFEJS UŻYTKOWNIKA APLIKACJI PLANER

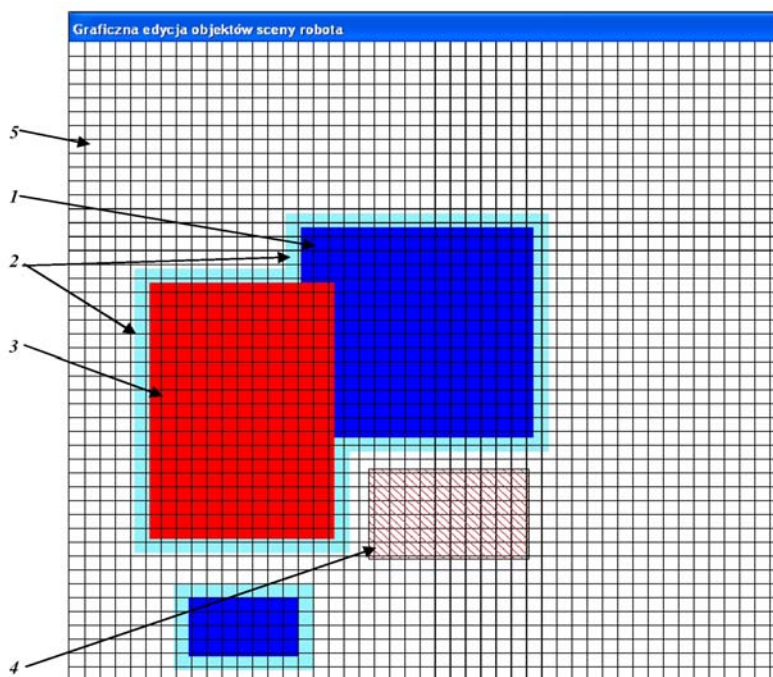
Utworzone oprogramowanie zostało podzielone na dwie części. Część wprowadzania danych oraz część prezentowania wyników obliczeń. Główny formularz wprowadzania danych obliczeniowych pokazano na rys. 4. Za praktyczną realizację wizualizacji obliczeń odpowiedzialny jest tylko jeden formularz. Wynikowa trajektoria ruchu jak również

przestrzenna projekcja obrazu sceny wraz z otoczeniem technologicznym i odpowiednimi strefami bezpieczeństwa została zrealizowana za pomocą biblioteki OpenGL [6]. Ponieważ wynikowa trajektoria z założenia ma zapewnić pełne bezpieczeństwo ruchu, zatem konieczne stało się zaimplementowanie dodatkowych funkcji symulacyjnych. Podczas procesu symulacji (przejęcia kuli opisanej na obiekcie manipulacji po wyznaczonej ścieżce ruchu) sprawdzany jest warunek koincydencji kuli z obiektami otoczenia technologicznego. Osobny wątek programowy sprawdza kolizyjność typu *face-to-face* [6]. Dzięki takiemu rozwiązaniu po poprawnym zakończeniu symulacji możemy być pewni, że w przypadku realizacji wyznaczonej trajektorii ruchu na rzeczywistym obiekcie (manipulatorze) nie dojdzie do kolizji, do jakiej mogłoby dojść w przypadku ustawienia zbyt małej liczby punktów podporowych trajektorii.



Rys. 4. Okno główne aplikacji PLANER:

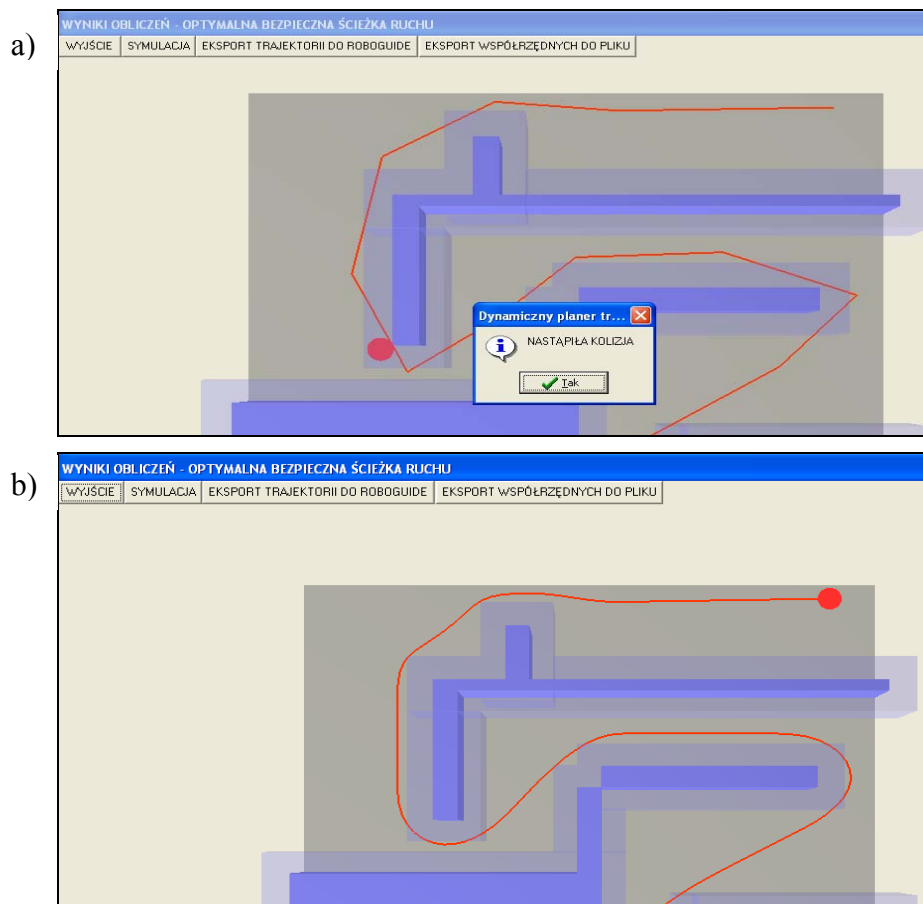
1 – definicja wielkości przedmiotu manipulowanego, 2 – określenie maksymalnej wysokości ruchu, 3 – pola tekstowe określenia współrzędnych punktów początkowego i końcowego ruchu, 4 – klawisz graficznego wskazania współrzędnych punktów początku i końca ruchu, 5 – informacja o aktualnie zdefiniowanych obiektach sceny, 6 – określenie ilości punktów trajektorii wyjściowej, 7 – klawisz rozpoczęcia obliczeń i prezentacji wyników, 8 – menu górne



Rys. 5. Graficzny edytor sceny: 1 – obiekt sceny, 2 – strefa bezpieczeństwa obiektów, 3 – edytowany obiekt położenie początkowe, 4 – edytowany obiekt o nowym wymiarze i położeniu, 5 – scena podzielona siatką o gradacji równej średnicy kuli opisanej na przedmiocie

Aby ułatwić wprowadzanie danych dotyczących wielkości oraz rozmieszczenia obiektów sceny robota aplikacja PLANER została wyposażona w graficzny kreator sceny. Wprowadzanie kolejnych obiektów może następować poprzez wprowadzanie wartości współrzędnych z klawiatury komputera, lub za pośrednictwem myszy metodą przeciągania (rys. 5). Każdy obiekt definiowany jest za pośrednictwem wskazania przekątnej podstawy.

Symulacja przebiega po torze identycznym z torem narzędzia (przedmiotu manipulacji) rzeczywistego manipulatora. W przypadku ustawienia zbyt małej ilości punktów wyznaczonej trajektorii do zrealizowania danego ruchu oprogramowanie PLANER pokaże możliwość wystąpienia kolizji, dzięki czemu użytkownik będzie miał możliwość ustawienia poprawnej (większej) wartości zmiennej ilości generowanych punktów trajektorii. Wpływ ilości generowanych punktów trajektorii na zachowanie warunku bezkolizyjności pokazano na rys. 6. Złożona trajektoria omijająca wszystkie przeszkody (obiekty otoczenia robota) może w przypadku niewłaściwej obsługi programu prowadzić do wystąpienia kolizji. Zaimplementowane narzędzie symulacyjne pozwala uniknąć kolizji poprzez wczesne wykrycie niewłaściwych danych wejściowych algorytmu wygładzania i optymalizowania ścieżki ruchu. Na rys. 6a pokazano bezpieczną ścieżkę ruchu zdefiniowaną za pomocą 20 punktów trajektorii. Zbyt mała liczba punktów trajektorii spowodowała odsunięcie od właściwego toru i kolizję. Na rys. 6b pokazano tą samą ścieżkę zapisaną za pomocą 200 punktów trajektorii (symulacja potwierdziła bezpieczeństwo tak zdefiniowanej trajektorii ruchu).



Rys. 6. Wpływ ilości punktów wynikowej trajektorii na zachowanie warunku bezkolizyjności ruchu podczas symulacji w programie PLANER

#### 4. INTEGRACJA APLIKACJI PLANER Z PRZEMYSŁOWYMI SYSTEMAMI PROGRAMOWANIA ROBOTÓW OFF-LINE

Głównym celem przyświecającym autorom podczas prac nad utworzeniem systemu PLANER było jego praktyczne zastosowanie. Warunkiem dopuszczenia aplikacji do stosowania w warunkach przemysłowych było przeprowadzenie licznych testów poprawności działania zaimplementowanych algorytmów. Ze względu na posiadaną przez Instytut Automatykacji Procesów Technologicznych i Zintegrowanych Systemów Wytwarzania bazę laboratoryjną, zdecydowano się na przeprowadzenie testów oprogramowania na robotach FANUC ARCMate 100iB. Roboty te wyposażone są w kontrolery RJ3iB, które pozwalają na korzystanie z kilku formatów plików. Z punktu widzenia integracji z oprogramowaniem PLANER najbardziej istotne okazały się pliki przechowywania i wymiany samych programów roboczych. Wyróżnić tutaj można pliki kompilatora KAREL (\*.ps) oraz pliki programów teachpendanta (\*.tp oraz \*.ls), które pozwalają na przechowywanie danych o sekwencjach ruchów manipulatora oraz zapis osiągniętych pozycji w globalnym układzie współrzędnych robota. Pliki programu teachpendanta występują w dwóch odmianach:

- kompresowanej binarnej formie (\*.tp)
- niekompresowanej formie tekstowej ASCII (\*.ls).

Obie dostępne formy plików programu roboczego mają ściśle określoną strukturę. Pliki \*.tp, ze względu na występujące sumy kontrolne, mogą być bezpośrednio wgrywane do układu sterowania. Pliki niekompresowane (\*.ls) mają znacznie prostszą strukturę, jednakże nie mogą być wgrywane do kontrolera manipulatora w sposób bezpośredni, dlatego chcąc korzystać z programów zapisanych trybem ASCII przed wgraniem do układu sterowania robota, muszą zostać przekompilowane do postaci binarnej. Niestety z powodu braku jakiegokolwiek dokumentacji na temat algorytmu kompresji plików binarnych (\*.tp), eksport danych z programu PLANER do tego formatu stał się niemożliwy. Postanowiono zatem do realizacji eksperymentu symulacyjnego działania programu PLANER zastosować format plików ASCII (\*.ls), a wymaganą kompresję zrealizować w dedykowanym oprogramowaniu firmy FANUC – systemie programowania off-line Roboguide.

##### 4.1. Definicja niekompresowanych plików roboczych ASCII (\*.ls)

Aby zapewnić maksymalny stopień bezpieczeństwa, producent systemu Roboguide wyposażył swoje oprogramowanie w narzędzie kontroli składni. Jeżeli plik ASCII (\*.ls) będzie odbiegał od ściśle określonego wzorca, to podczas importu programu roboczego zostanie wygenerowany komunikat o błędzie składni i proces importu trajektorii zostanie przerwany. Ze względu na brak dostępnej specyfikacji plików \*.ls, koniecznym okazało się przeprowadzenie analizy składni metodą inżynierii odwrotnej (reverse engineering). W tym celu wygenerowano szereg programów roboczych w programie Roboguide, a następnie wykonano ich eksport do postaci plików niekompresowanych ASCII. Analiza otrzymanych plików pozwoliła na sprecyzowanie warunków poprawności składniowej, jak również samej struktury plików programu roboczego \*.ls. Każdy plik wymiany w formie niekompresowanej ma trzy główne części poprzedzone blokiem nagłówka, który rozpoczyna znacznik **/PROG**. Jest to znacznik określający nazwę programu roboczego. Nazwa występująca po znaczniku **/PROG** poprzedzona musi być dwoma znakami spacji (**0x20** w kodzie Hex). Bardzo ważną kwestią jest zgodność fizycznej nazwy pliku z występującym po znaczniku **/PROG** łańcuchem tekstowym. W przypadku niezgodności nazw lub braku podwójnego znaku spacji zostanie wykryty błąd składni, co uniemożliwi import pliku na platformę Roboguide. Linia nagłówkowa musi zostać zakończona znacznikiem ASCII końca linii (dwuznak: **0x0D**, **0x0A**

w kodzie Hex). Kolejne znaczniki również mają odpowiedni układ uzupełniania łańcuchów tekstowych znakami spacji oraz tabulacji. Przykładowy zapis linii programu roboczego pokazano na rys. 7.

```

0x20 0x20 0x20 9:L 0x20 P T 9 1 0x20 2 0 0 0 m m / s e c 0x20 C N T 1 0 0 0x20 0x20 0x20 0x20 ; 0x0D 0x0A
0x20 0x20 1 0 : L 0x20 P T 1 0 1 0x20 2 0 0 0 m m / s e c 0x20 C N T 1 0 0 0x20 0x20 0x20 0x20 ; 0x0D 0x0A
    
```

Rys. 7. Zapis przykładowych linii programu roboczego podczas eksportu do formatu ASCII (\*.ls) Każda linia programu roboczego musi mieć odpowiadającą jej w sekcji /POS definicję orientacji i położenia punktu podporowego trajektorii. Przyjęty, sztywny format ramek zapisu pozycji (punktów podporowych) pokazano na rys. 8.

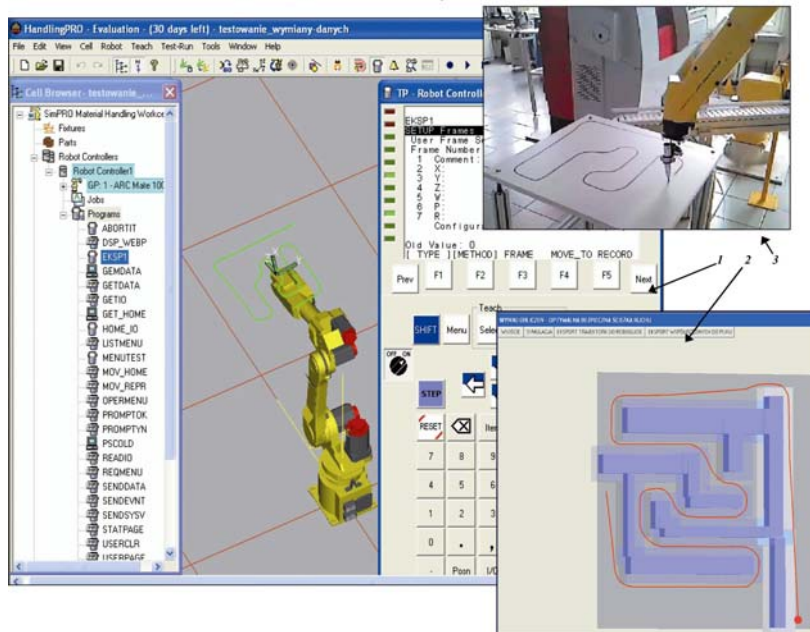
```

P T 1 1 { 0x0D 0x0A
0x20 0x20 0x20 G P 1 : 0x0D 0x0A
0x09 U F : 0x20 1 , 0x20 U T : 0x20 1 , 0x09 C O N F I
G : 0x20 N U T , 0x20 0 , 0x20 0 , 0x20 - 1 , 0x0D 0x0A
0x09 X = 0x20 0x20 0x20 0x20 1 5 0 . 0 0x20 0x20 m m , 0x09 Y = 0x20 0x20 0x20 0x20 5 0 0 .
0 0x20 0x20 m m , 0x09 Z = 0x20 0x20 0x20 0x20 1 1 5 0 . 0 0x20 0x20 m m , 0x0D 0x0A
0x09 W = 0x20 0x20 0x20 0x20 1 8 0 . 0 0 0x20 0x20 d e g , 0x09 P = 0x20 0x20 0x20 0x20 0 . 0
0 0x20 d e g , 0x09 R = 0x20 0x20 0x20 0x20 0 . 0 0 d e g 0x0D 0x0A
} : 0x0D 0x0A
    
```

Rys. 8. Zapis definicji pozycji robota.

Kolorem białym zaznaczono modyfikowane w aplikacji PLANER parametry

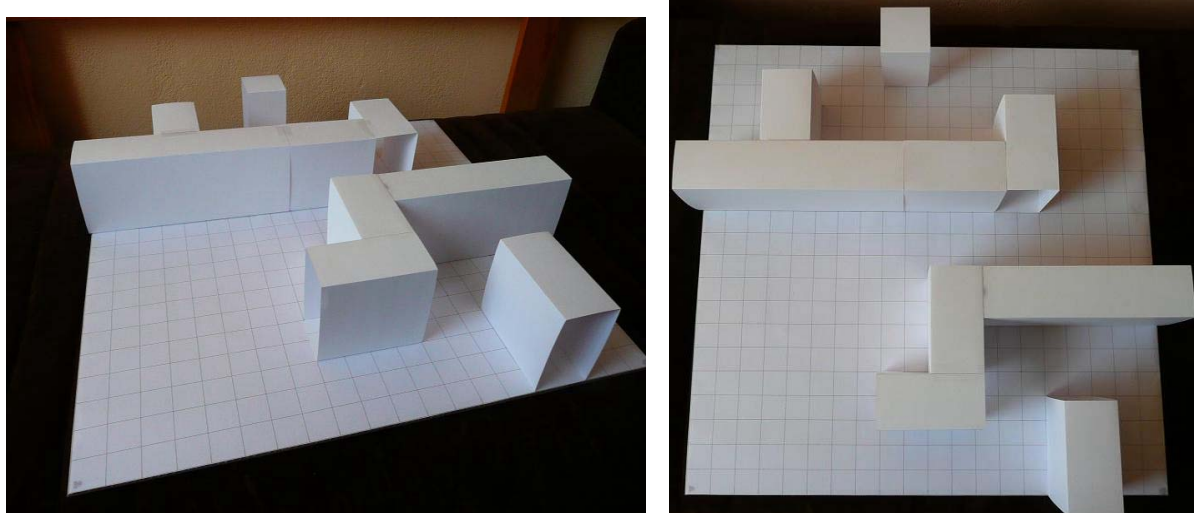
Korzystając z opracowanych formuł eksportu danych możliwe było wczytanie wyznaczonej w systemie PLANER, bezpiecznej i gładkiej trajektorii ruchu do oprogramowania Roboguide. Przykładowy, zaimportowany do oprogramowania Roboguide, a następnie wgrany do kontrolera robota program ruchu pokazano na rys. 9.



Rys. 9. Wygenerowana w aplikacji PLANER oraz zaimportowana do FANUC Roboguide przykładowa trajektoria: 1 – oprogramowanie Roboguide, 2 – aplikacja PLANER, 3 – robot podczas realizacji wytyczonej trajektorii

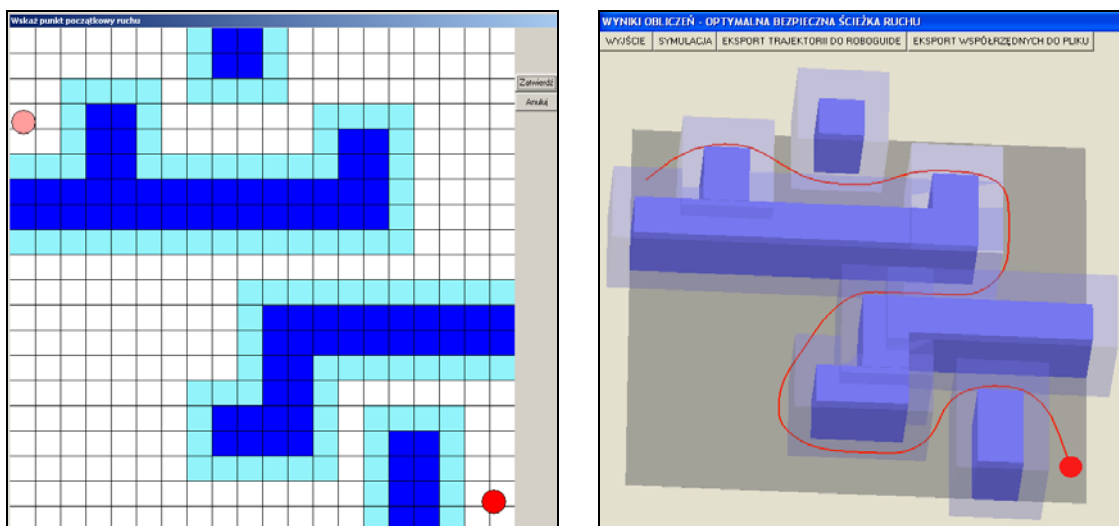
## 5. EKSPERYMENT SYMULACYJNY

Dzięki opracowanemu modułowi eksportu wyznaczonych w opracowanym systemie PLANER trajektorii ruchu do oprogramowania narzędziowego FANUC możliwe stało się przeprowadzenie eksperymentów symulacyjnych. Ze względów bezpieczeństwa postanowiono pierwsze próby odtworzenia wyznaczonej trajektorii z wykorzystaniem rzeczywistego robota przeprowadzić na makiety sceny. Wszystkie obiekty makiety zostały wykonane w taki sposób, aby ewentualna kolizja nie spowodowała uszkodzenia manipulatora. Na rys. 10 pokazano wykonaną makiety.



Rys. 10. Wykonana makieta sceny

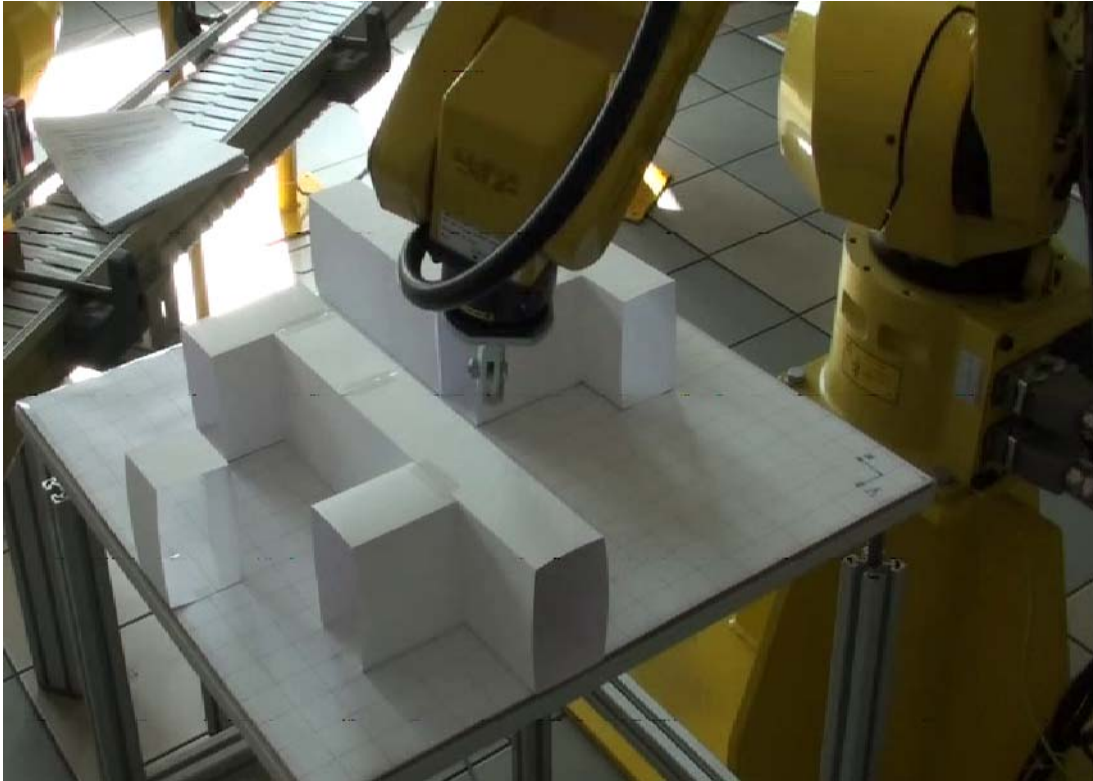
Mając zwymiarowane wszystkie obiekty utworzonej makiety sceny, przystąpiono do wprowadzania danych o ich położeniu do opracowanego systemu PLANER. Po wprowadzeniu danych ustawiono przypadkowo położone punkty początku i końca ruchu, po czym uruchomiony został algorytm obliczeniowy. Przyjęte punkty początku i końca ruchu oraz wygenerowaną bezpieczną, gładką i optymalną ścieżkę ruchu pokazano na rys. 11.



Rys. 11. Przyjęte punkty początku i końca ruchu oraz wygenerowana trajektoria wynikowa (aplikacja PLANER)



Po wyeksportowaniu uzyskanej trajektorii ruchu do omawianego już wcześniej formatu niekompresowanego ASCII (pliki \*.Is) przystąpiono do wgrywania nowoutworzonego programu roboczego do kontrolera RJ3iB posiadanego w laboratorium robota FANUC. Podczas testów bardzo ważne było poprawne zdefiniowanie wysokości ruchu narzędzia, tak, aby pomiędzy przeszkodami otoczenia technologicznego przemieszczana była jedynie sama końcówka spawalnicza bez uchwytu mocującego do kiści robota, co pokazano na rys. 12.



Rys. 12. Końcówka spawalnicza MIG/MAG przemieszczana podczas przeprowadzanego eksperymentu

Jak się okazało opracowany system PLANER wygenerował w pełni bezpieczną trajektorię ruchu. W przypadku podjęcia próby wygenerowania w sposób ręczny programu przejścia przez utworzony labirynt okazałoby się, że czas programowania robota byłby nieporównywalnie większy, co pozwala stwierdzić, że opracowane oprogramowanie po przeprowadzeniu większej liczby testów, będzie mogło być stosowane na szeroką skalę w warunkach przemysłowych.

## 6. PODSUMOWANIE I WNIOSKI

Uzyskane wyniki przestrzennego planowania trajektorii pokazały, że przyjęte rozwiązanie pozwoliło zapewnić osiągnięcie skutecznego kompromisu pomiędzy jakością planowania w pełnej przestrzeni 3D z szybkością obliczeń dostępną jedynie dla metod bazujących na płaskich scenach roboczych. Przeprowadzone badania zachowania algorytmu, jak również eksperymenty symulacyjne pokazały, że opracowany system planowania bezkolizyjnych ścieżek ruchu robotów manipulacyjnych w przestrzeniach zadaniowych działa poprawnie i może być istotnym rozszerzeniem możliwości istniejących na rynku systemów programowania robotów metodami off-line. Opracowana aplikacja PLANER, będąca implementacją opracowanego algorytmu, pozwoliła na integrację z istniejącym na rynku

systemem programowania off-line robotów firmy FANUC. Przeprowadzone dzięki temu eksperymenty pozwoliły na potwierdzenie skuteczności metody. Uzyskiwane przez aplikację PLANER trajektorie zostały dokładnie zaimportowane do systemu Roboguide, co w efekcie pozwoliło wygenerować program ruchu, zapewniający wiernie odwzorowanie wytyczonej, bezkolizyjnej trajektorii ruchu przez rzeczywistego robota.

Zastosowane narzędzie graficzne bazujące na obsłudze biblioteki graficznej OpenGL pozwoliło na utworzenie złożonego interfejsu prezentacji wyników. Dzięki zastosowaniu odpowiednich procedur użytkownik może za pomocą (zadeklarowanych na etapie tworzenia aplikacji PLANER) gestów myszy przemieszczać widok, przybliżać i oddalać scenę, jak również dokonywać rotacji punktu obserwacji sceny. Topologię gestów myszy przyjęto zgodną ze stosowaną w większości zaawansowanych programów graficznych, co czyni pracę z opracowanym systemem prostszą i intuicyjną.

Dodatkowym atutem przemawiającym na korzyść stosowania bibliotek OpenGL do wspomagania pracy programistów robotów jest możliwość weryfikacji wyników obliczeń na drodze symulacji. Zastosowane mechanizmy wykrywania kolizji face-to-face pozwalają z niezwykłą precyzją zweryfikować otrzymaną trajektorię ruchu, przez co zwiększa się bezpieczeństwo obsługi systemu.

## LITERATURA

1. Duleba I.: Metody i algorytmy planowania ruchu robotów mobilnych i manipulacyjnych. Akademska Oficyna Wydawnicza EXIT, Warszawa 2001.
2. Fortuna Z., Macukow B., Wąsowski J.: Metody numeryczne. s. Podręczniki Akademickie. Elektronika. Informatyka. Telekomunikacja. Wydawnictwa Naukowe PWN. Warszawa.
3. Jankowski B.: Programowanie w praktyce. Wydawnictwo MIKOM, Warszawa 1999.
4. Latombe J.-C.: Robot motion planning. Kluwer Academic Publishers. London. 1993.
5. Majchrzak E., Mochnacki B.: Metody numeryczne. Podstawy teoretyczne, aspekty praktyczne i algorytmy. Wydawnictwo Politechniki Śląskiej, Gliwice 2004.
6. Orłowski A.: OpenGL. Leksykon kieszonkowy. Wydawnictwo HELION, Gliwice 2005.
7. Reclik D., Kost G.: The comparison of elastic band and B-Spline polynomials methods in smoothing process of collision-less robot trajectory, Journal of Achievements in Materials and Manufacturing Engineering, Volume 29, Issue 2, 2008.
8. Reclik D., Kost G.: The 2 ½ D Algorithm in Robot Workspace Analysis, 4th Conference Mechatronic Systems and Materials 2008, MSM 2008, Acta Mechanica et Automatica, Volume 2, No. 3, Białystok Technical University, 2008.
9. Reclik D., Kost G.: Zastosowanie metody 2 ½ D w planowaniu trajektorii robotów manipulacyjnych, Pomiary Automatyka Robotyka, Nr 2/2008,
10. Reclik D., Kost G.: Dynamiczne planowanie trajektorii robotów manipulacyjnych. Przegląd mechaniczny Nr 1'08. str 23-29.
11. Reclik D., Kost G.: A rational B-Spline curves in robot collision-free movement planning, Journal of Automation, Mobile Robotics and Intelligent Systems, Volume 2, No. 3, Industrial Research Institute for Automation and Measurements PIAP, Warszawa 2008.