

Dr inż. Adam Słota
Dr inż. Krzysztof Krupa
Mgr inż. Andrzej Konicki
Mgr inż. Piotr Doktor
Politechnika Krakowska, Wydział Mechaniczny
słota@mech.pk.edu.pl

Programowanie w systemie Delmia robotów przemysłowych dla zadań paletyzacji i spawania

Streszczenie

W pracy przedstawiono możliwości wykorzystania oprogramowania Delmia do programowania robotów przemysłowych. Tok postępowania przy budowie i konfiguracji modelu stanowiska, definiowaniu zadań dla robotów, symulacji i generowaniu programów sterujących został pokazany na dwóch przykładach: stanowiska do paletyzacji oraz stanowiska spawalniczego. Przygotowane w środowisku wirtualnym programy sterujące dla robotów zostały uruchomione na stanowiskach rzeczywistych.

Abstract

The paper presents application of Delmia system for industrial robot programming. A procedure of a robotized cell modeling and configuration, robot task definition, simulation and control program generation is presented for two cases: robotized palletization cell and welding cell. Prepared in virtual environment control programs have been verified in real robotized cells.

1. Wprowadzenie

Systemy komputerowego wspomaganie obejmują coraz szersze obszary i dostarczają wspomaganie w coraz większym zakresie: w obszarze konstruowania, począwszy od projektowania 2D, poprzez modelowanie 3D kompletnych produktów, do analizy kolizji, analiz kinematycznych i wytrzymałościowych. W zakresie wytwarzania wspomaganie systemów CAX, które początkowo wyznaczały ścieżki narzędzia oraz generowały kod NC dla obróbki na obrabiarkach OSN, później uzupełniono o możliwość symulacji z uwzględnieniem modelu 3D obrabiarki i jej oprzyrządowania. Obserwowany w ostatnich latach rozwój w głównej mierze dotyczy oprogramowania z zakresu wirtualnego wytwarzania. Podejmowane działania ukierunkowane są na opracowanie środków, które mają na celu przeniesienie jak największej ilości działań dotyczących przygotowania produkcji do środowiska wirtualnego. Oprogramowanie tego rodzaju umożliwia modelowanie i analizę w środowisku 3D, coraz częściej z możliwością wykorzystania technik wirtualnej rzeczywistości, zautomatyzowanych operacji obróbki, transportu i manipulacji, jak również realizowanych manualnie operacji montażu lub demontażu. W przypadku wykorzystania w procesie produkcyjnym robotów przemysłowych, dla zdefiniowanych i zweryfikowanych poprzez symulację w środowisku wirtualnym zadań dla robotów, możliwe jest wygenerowanie programów sterujących. Jeżeli wirtualny model stanowiska zrobotyzowanego zawiera wszystkie istotne cechy środowiska rzeczywistego, przygotowane programy sterujące mogą być przeniesione i uruchomione na stanowisku rzeczywistym. Jest to alternatywny w stosunku do

metody „uczenia”, coraz częściej stosowany, sposób programowania robotów przemysłowych.

Producenci robotów udostępniają aplikacje umożliwiające programowanie off-line robotów przez nich dostarczanych: *RoboGuide* firmy Fanuc Robotics , *PC Roset* firmy Kawasaki, czy *Robot Studio* firmy ABB [2].

W pracy – jako środowisko programowania – wykorzystano system *Delmia* [4]. Jest to oprogramowanie z obszaru wirtualnego wytwarzania, w którym zagadnienia modelowania, symulacji i programowania stanowisk zrobotyzowanych to tylko jeden z obszarów wspomagania. System *Delmia* umożliwia pracę z robotami różnych producentów i zawiera obszerną bibliotekę modeli robotów. Nie jest to jednak baza zamknięta – pełna kompatybilność z systemem CAD *Catia* pozwala budować własne modele geometryczne robotów i urządzeń dodatkowych, a możliwość importu danych w standardowych formatach (step, model) pozwala na korzystanie z baz modeli, udostępnianych przez producentów sprzętu wykorzystywanego w procesie wytwarzania.

W pracy przedstawiony zostanie tok postępowania przy programowaniu robotów przemysłowych, na przykładzie zadań paletyzacji i spawania.

2. Programowanie zadania paletyzacji

2.1 Budowa stanowiska i jego model

Głównym zasobem stanowiska jest 4-osiowy robot Kawasaki *ZD130S* dedykowany do zadania paletyzacji. Powtarzalność pozycjonowania końcówki roboczej wynosi 0,5 [mm], maksymalny zasięg 3255 [mm]. Możliwość chwytania zapewnia specjalnie skonstruowany chwytak, składający się z ssawki, w której powietrze zasysane jest przez ejektor. Integracja chwytaka z robotem sprowadza się do zamocowania go na kiści robota oraz zrealizowaniu przepływu sygnału poprzez połączenie rozdzielacza pneumatycznego z kontrolerem robota przez przekaźnik. Na stanowisku rozłożono trzy palety z odpowiednio ułożonymi dziewięcioma pudełkami kartonowymi (cztery z nich mają mniejszą wysokość). Test nie odbywał się w warunkach produkcyjnych, a pudełka mogły być zdeformowane, w związku z tym ustawiono je na paletach w odległości około 3 cm, w celu uniknięcia kolizji. Widok stanowiska rzeczywistego oraz początkowe ułożenie pudełek przedstawiono na rysunku 1¹.

Delmia zawiera biblioteki robotów wielu producentów – między innymi są to: Fanuc, Kawasaki, Kuka, Deichen, Panasonic, Motoman. Oznacza to, że jeśli chcemy sprawdzić, jak w produkcji zachowa się na przykład robot Fanuc R-2000iB, wystarczy odnaleźć i wczytać model obejmujący dane geometryczne oraz parametry kinematyczne. Oszczędza to sporo czasu poświęconego na generowanie modeli geometrycznych oraz relacji kinematycznych. Jednak gdy wymagany model nie znajduje się w bibliotece, istnieje możliwość przygotowania go samodzielnie. Ponadto można wirtualizować dowolne urządzenie – nie tylko robota.

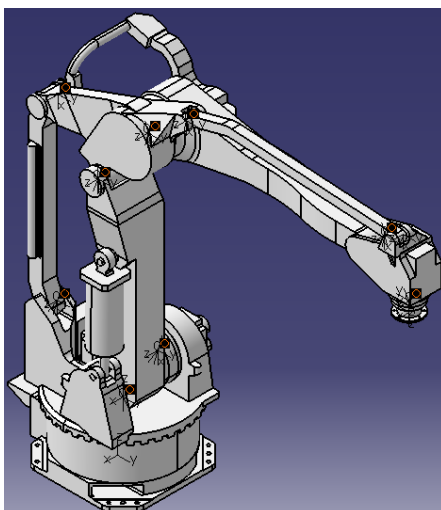
Ponieważ model wykorzystywanego na stanowisku robota *ZD130S* jest niedostępny w bibliotece robotów systemu *Delmia V5R19*, koniecznym było samodzielne zbudowanie takiego modelu, korzystając z jego dokumentacji technicznej [3].

Model geometryczny robota *ZD130S*, przedstawiony na rysunku 2, został zbudowany w oparciu o dane uzyskane od producenta robota.

¹ Stanowisko zostało udostępnione przez firmę Roboty Przemysłowe Krzysztof Sulikowski z Krakowa.



Rys. 1 Stanowisko do paletyzacji z robotem Kawasaki ZD130S

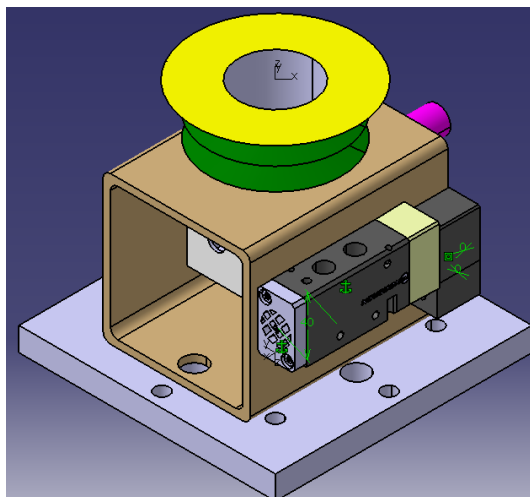


Rys. 2 Model Robotu Kawasaki ZD130S

Dodatkowo do modelu wprowadzono parametry, które są wymagane do zdefiniowania kinematyki urządzenia:

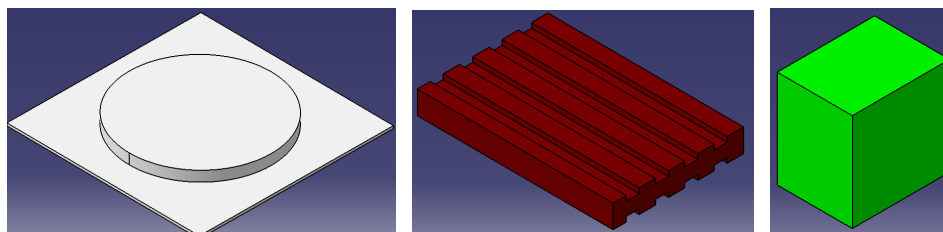
- element bazowy, punkt *TCP*, sposób rozwiązania zadania odwrotnego, ogniwa napędowe i napędzane,
- stopnie swobody ogniw ruchomych,
- parametry solvera: dokładność obliczeń, liczba obliczanych stopni swobody ruchu w zadaniu odwrotnym dla punktu *TCP*,
- zakresy ruchu poszczególnych osi obrotowych i przesuwnych, relacje kinematyczne pomiędzy ogniwami związanymi kinematycznie.

Kolejnym elementem stanowiska jest chwytak. Poprawne zamodelowanie go jest niezbędne, ponieważ od jego wymiarów zależy niezawodność chwytania pudeł kartonowych. Przy chwytaniu pudeł ssawka skraca się, dlatego przed utworzeniem modelu chwytaka przeprowadzono doświadczenia, które wykazały, że wielkość skrócenia wynosi 20 mm. Zbudowany model chwytaka ma postać jak w stanie zasysania (rysunek 3).



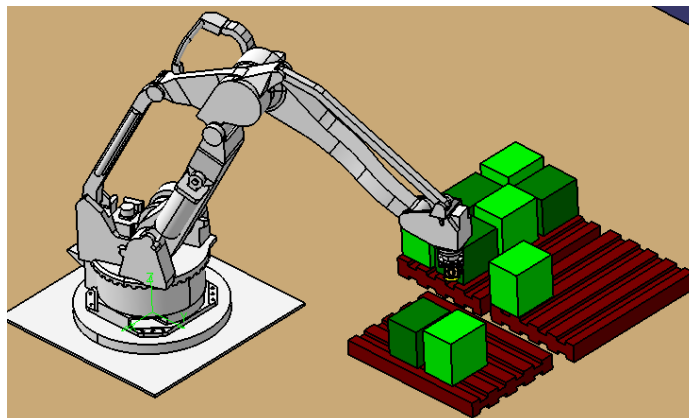
Rys. 3 Model chwytaka

Modele pozostałych elementów stanowiska (podstawy pod robota, palety i pudełek) przedstawiono na rysunku 4.



Rys. 4 Modele podstawy, palety i pudełka kartonowego

Po utworzeniu wszystkich wirtualnych komponentów stanowiska należy ustawić je na scenie programu (w dokumencie procesu), w pozycji i orientacji odpowiadającej rozmieszczeniu komponentów na stanowisku rzeczywistym. W tym celu wykorzystano narzędzia umożliwiające przesuwanie, obracanie oraz wzajemne dopasowywanie komponentów na stanowisku wirtualnym. Dodatkowo należy połączyć model chwytaka z modelem robota, w sposób pozwalający współpracować obydwu urządzeniom. Widok wirtualnego stanowiska do paletyzacji przedstawiono na rysunku 5.



Rys. 5 Widok wirtualnego stanowiska do paletyzacji

2.2 Definiowanie zadania robota i symulacja

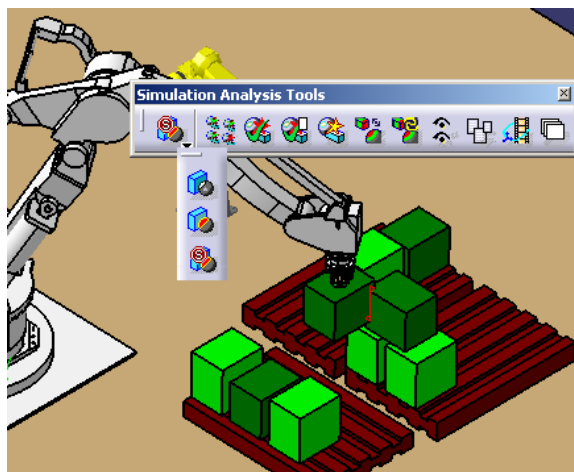
Stanowisko zbudowano w celu prezentacji procesu paletyzacji. Paleta załadowana pudełkami do połowy ma zostać przeładowana na pustą paletę. Podczas tego przeładunku niższe pudełka mają być ułożone na dwóch wyższych, a orientacja zmienia się o 90 stopni. Następnie pudełka z całkowicie załadowanej palety zostają przeniesione na zwolnioną paletę.

Definiowanie zadania dla robota realizowane jest za pomocą przejrzystej i czytelnej metody. Użytkownik umieszcza na scenie roboczej *tagi*, czyli punkty charakterystyczne stanowiska. Określają one położenie i orientację końcówki roboczej robota. *Tagi* na scenie w symulacji przedstawione są za pomocą małych czarnych układów współrzędnych. Dostępne sposoby wstawiania *tagów* oraz narzędzia do ich modyfikacji pozwalają swobodnie definiować elementy zadania robota, czyli operacje ruchu. Ruch pomiędzy *tagami* może być realizowany według interpolacji liniowej, przegubowej lub kołowej. Parametry ruchu, takie jak prędkość czy przyspieszenie, definiuje się w postaci profili ruchu.

Po zdefiniowaniu wymaganej kolejności przejścia końcówki roboczej przez poszczególne *tagi* zadanie uzupełniono o operacje pomocnicze, takie jak: uchwycenie czy upuszczenie przedmiotu, oczekiwanie na sygnał wejścia lub wysłanie sygnału do innego urządzenia, zmiana właściwości obrabianego przedmiotu.

W przypadku omawianego stanowiska do paletyzacji, definiowanie ruchu wykonywano stopniowo etapami, uwzględniając kolejne spostrzeżenia:

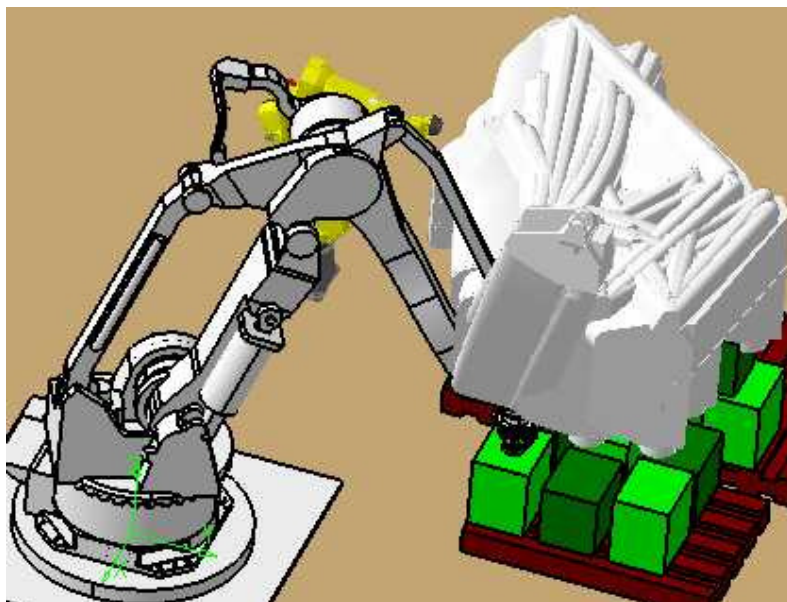
- interpolacja ruchu musi być liniowa - wynika to z kształtu pudełek kartonowych, ponieważ przed i po uchwyceniu/upuszczeniu pudełka ruch chwytaka powinien być wykonywany w kierunku prostym do podłoża,
- kolejne ruchy muszą uwzględniać zmieniający się układ pudełek kartonowych - na rysunku 6 pokazano kolizję, którą wykryto w trakcie symulacji.



Rys. 6 Przykład kolizji na stanowisku paletyzacji

Delmia posiada kilka użytecznych narzędzi pozwalających zweryfikować poprawność zdefiniowanych zadań. Możliwa jest ocena przestrzeni roboczej robota, sprawdzenie osiągalności zdefiniowanych dla zadania *tagów*, czy automatyczne ustawienie robota, tak aby zdefiniowane dla niego zadanie można było zrealizować. Ciekawą funkcją jest możliwość stworzenia bryły, która powstaje jako suma przestrzeni, zajmowanych kolejno w trakcie ruchu wybranego elementu mechanizmu. Na rysunku 7 przedstawiono wygenerowany taki obszar dla nadgarstka robota. Łatwo można

zauważyć, że w żadnym wypadku nie koliduje on z żółtym robotem Fanuc, znajdującym się w tle stanowiska.



Rys. 7 Sprawdzenie wymaganej przestrzeni roboczej dla końcówki robota

Ponadto przewidziano możliwość synchronizacji pracy wielu urządzeń poprzez przekazywanie sygnałów wejścia/wyjścia.

Poprzez symulację zweryfikowano poprawność działania stanowiska, a następnie przystąpiono do generowania kodu sterującego za pomocą specjalnej funkcji *OLP*. Na rysunku 8 przedstawiono fragment wygenerowanego kodu sterującego dla stanowiska paletyzacji.

```
Program.1
.PROGRAM RobotTask.2()
SPEED 30 ALWAYS
ACCEL 90 ALWAYS
ACCURACY 0 ALWAYS
TOOL gripper.1.Tool1
LMOVE Tag21
LMOVE TRANS(-749.999979 1540.000011 789.999991 -180.000000 0.000000 89.99
LMOVE TRANS(-1639.999874 1999.999917 810.000002 -180.000000 0.000000 89.9
LMOVE Tag31
LMOVE TRANS(-1639.998653 2000.000314 810.001001 -180.000000 0.000000 89.9
LMOVE TRANS(-750.001509 2219.999593 790.000010 -180.000000 0.000000 89.99
LMOVE Tag23
LMOVE TRANS(-750.000064 2219.999989 790.000005 -180.000000 0.000000 89.99
LMOVE TRANS(-1639.998928 2339.999834 810.000000 -180.000000 0.000000 89.9
LMOVE Tag32
LMOVE TRANS(-1639.999701 2339.999834 810.000000 -180.000000 0.000000 89.9
```

Rys. 8 Fragment wygenerowanego kodu sterującego

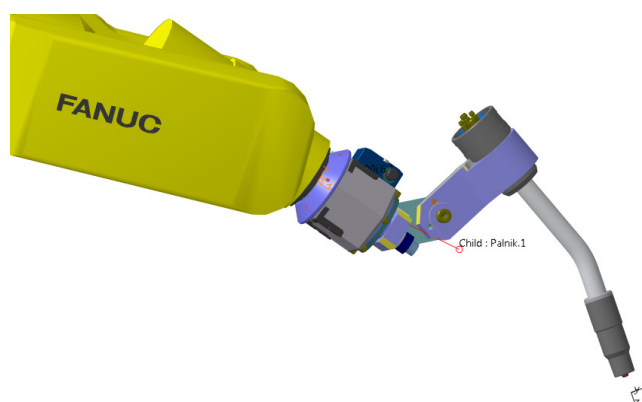
Po wygenerowaniu programu sterującego zauważono, że kolejność współrzędnych określających orientację *TCP* jest inna od wymaganej przez kontroler *D43*. Dlatego uzyskany z *Delmii* program został zmodyfikowany przed przesłaniem do kontrolera i uruchomieniem na stanowisku rzeczywistym.

3. Programowanie zadania spawania

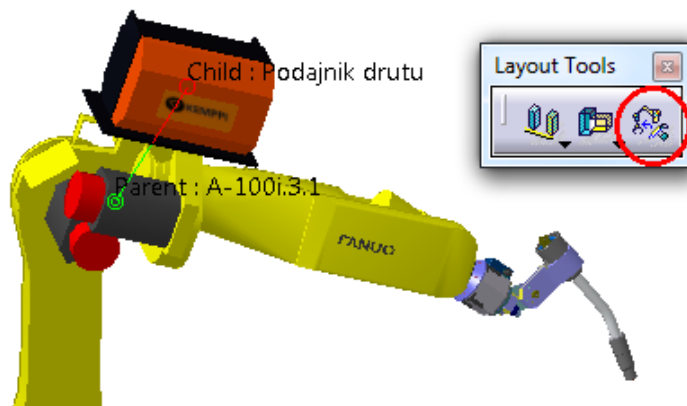
3.1 Budowa stanowiska i jego model

Zrobotyzowane stanowisko spawalnicze przeznaczone jest do spawania metodą *MIG/MAG*. W jego skład wchodzi: robot *Fanuc ArcMate 100i*, kontroler robota *RJ2*, źródło spawalnicze wraz z chłodnicą, podajnik drutu elektrodowego, złącze antykolizyjne wraz z kołnierzem mocującym, palnik, butla z gazem osłonowym, szpula z drutem elektrodowym i ogrodzenie ochronne [1]. Wykorzystany na stanowisku robot jest robotem dedykowanym do celów spawalniczych. Posiada sześć osi sterowanych, a jego specjalna konstrukcja umożliwia położenie spoiny w trudno dostępnych miejscach.

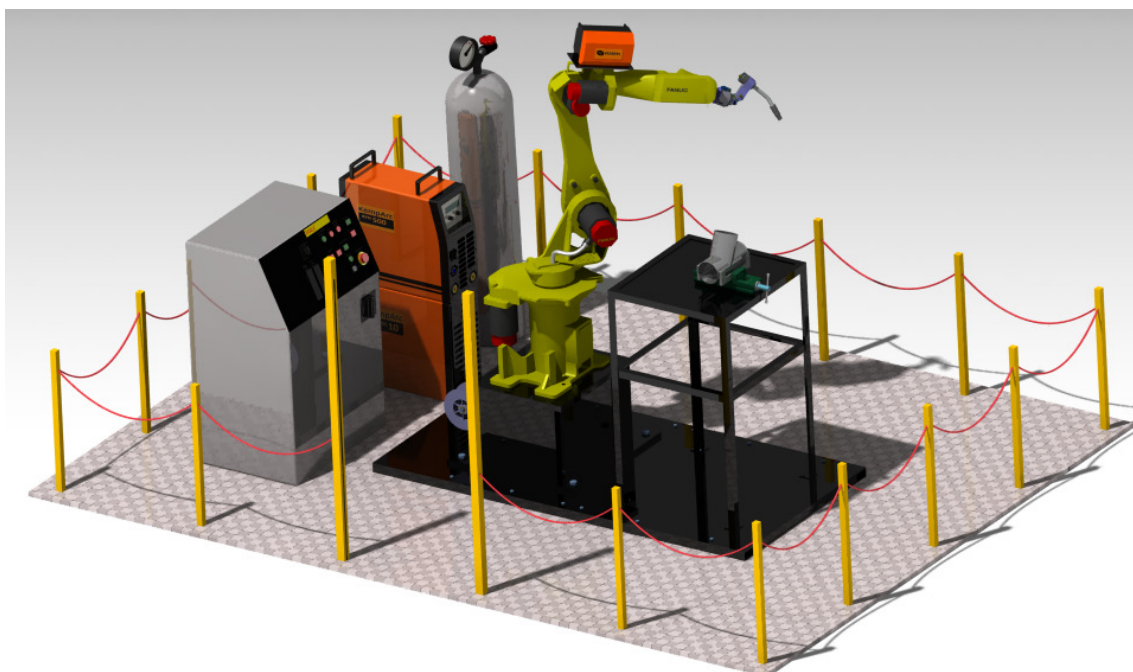
Do zamodelowania stanowiska zostały wykorzystane modele 3D dostępne w bibliotekach *Delmii* (*Fanuc ArcMate 100i*), modele udostępnione przez firmę *Abicor Binzel* zajmującą się produkcją komponentów do stanowisk spawalniczych (kołnierz mocujący, złącze antykolizyjne oraz palnik). Pozostałe komponenty zostały zamodelowane w programie *Catia V5* z wykorzystaniem techniki modelowania bryłowego oraz powierzchniowego. Całość stanowiska złożono w programie *Delmia V5*. Zostały określone wzajemne położenia pomiędzy poszczególnymi elementami stanowiska. Na rysunku 9 pokazano widok modelu palnika oraz jego połączenia z końcówką robota. Rysunek 10 przedstawia umieszczenie podajnika drutu spawalniczego. Widok modelu całego stanowiska widoczny jest na rysunku 11.



Rys. 9 Zdefiniowane narzędzie robota



Rys. 10 Sposób zamocowania podajnika drutu



Rys. 11 Zrobotyzowane stanowisko spawalnicze

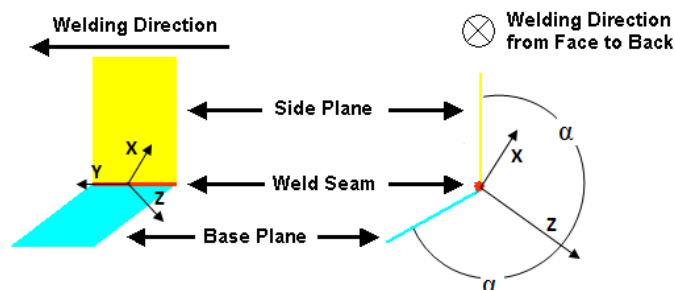
Na podstawie parametrów dostarczonych przez producenta (odległość od osi Z końcówki robota, przesunięcie w osi Y, kąt zakrzywienia palnika) zdefiniowano punkt *TCP* narzędzia.

3.2 Definiowanie zadania robota i symulacja

Realizowane na stanowisku zadanie to operacja spawania połączenia rurowego. Dwie rury o średnicach 110 [mm] są wstępnie połączone ze sobą pod kątem 67 stopni. Przedmiot jest zamocowany w imadle wyposażonym w specjalnie zaprojektowane do tego celu szczęki. Zadanie składa się z następujących etapów:

- szybki ruch ustawczy do punktu w pobliżu spoiny,
- dojazd do spoiny,
- wykonanie pierwszej części spoiny,
- odjazd od spoiny w celu przeorientowania narzędzia,
- ponowny dojazd do spoiny,
- wykonanie drugiej części spoiny,
- odjazd od spoiny i powrót do pozycji wyjściowej.

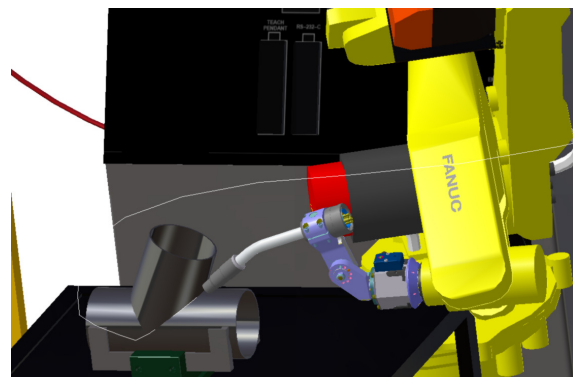
Operacje ruchu robota zdefiniowano poprzez określenie *tagów*, do których robot porusza się punktem *TCP*. Tagi do ruchów ustawczych zostały zdefiniowane przez „uczenie”. Metoda ta polega na ustawieniu punktu *TCP* robota w żądanym położeniu i orientacji (poprzez sterowanie poszczególnymi osiami, bądź też przez sterowanie przemieszczeniami wzdłuż osi kartezjańskiego układu współrzędnych) oraz zapamiętanie tego położenia w postaci *taga*. Zdefiniowanie *tagów* do wykonania spoiny polegało na wskazaniu powierzchni, pomiędzy którymi ma być wykonana spoina. *Tagi* tworzone są na krzywej przecięcia się wskazanych powierzchni. Orientację *tagów* pokazano na rysunku 12 – oś Z jest ustawiona wzdłuż dwusiecznej kąta pomiędzy powierzchniami, oś Y jest styczna do krzywej, natomiast oś X jest zdefiniowana w ten sposób, by układ był prawoskrętny.



Rys. 12 Orientacja tagów wygenerowanych przy pomocy narzędzi modułu Arc Welding

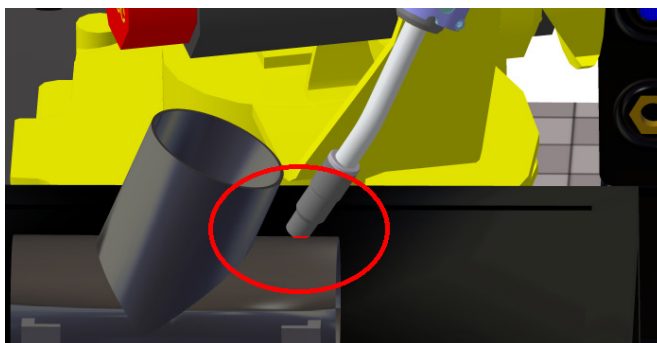
Dla ruchów ustawczych przyjęto maksymalną prędkość ruchu robota. Prędkość ruchu dojazdowego do spoiny wynosi 25% prędkości maksymalnej, a prędkość spawania ustawiono na 30 [mm/s]. Poza prędkościami konieczne jest też odpowiednie zdefiniowanie typów ruchu oraz sposobu osiągnięcia punktów. Jeśli trajektoria punktu *TCP* pomiędzy punktami nie ma znaczenia, można skorzystać z ruchu typu *joint*. Ze względu na konieczność dokładnego odwzorowania krzywizny spoiny, do jej wykonania został wykorzystany ruch typu *circular*. Ostatnim ważnym elementem jest sposób osiągnięcia kolejnych punktów. Do wyboru dostępna jest opcja *Fine* – robot zatrzymuje się po osiągnięciu pozycji, a następnie rusza do kolejnego punktu oraz *CNT* – robot zwalnia przy dojeździe do punktu pośredniego i płynnie przechodzi do punktu docelowego (*CNT 50* oznacza, że robot zwolni do 50% aktualnej prędkości). W przypadku zadania spawania konieczne jest zastosowanie opcji *CNT*.

Po zdefiniowaniu całości zadania przeprowadzona zostaje symulacja w celu jego weryfikacji (ścieżka ruchu, prędkości, kolizje). Podczas wykonywania symulacji warto zwrócić uwagę na dwie bardzo przydatne funkcje. Pierwsza z nich – *TCP Trace* – pozwala na wygenerowanie podczas symulacji trajektorii, po jakiej porusza się punkt *TCP*. Funkcja ta jest bardzo pomocna w przypadkach, gdy istotną rolę odgrywa dokładne odwzorowanie założonej krzywej. Ma to miejsce w przypadku spawania. Porównując wygenerowaną ścieżkę z oczekiwaną ścieżką, możliwa jest ocena poprawności wykonania zadania (rysunek 13). W przypadku rozbieżności można edytować zadanie poprzez zmianę typu ruchu, czy też wstawianie dodatkowych punktów pośrednich.



Rys. 13 Wykorzystanie funkcji TCP trace

Kolejną funkcją jest wykrywanie kolizji. Funkcja ta (*Clash detection STOP*) pozwala na wykrycie kolizji oraz zatrzymanie wykonywania symulacji. Miejsce wystąpienia kolizji oznaczane jest kolorem czerwonym (rysunek 14). Funkcja ta jest szczególnie przydatna podczas symulacji rozbudowanych stanowisk, gdzie można przeoczyć niewielką kolizję. Wykorzystanie tego narzędzia zapobiega uszkodzeniu sprzętu na stanowisku.



Rys. 14 Przykład kolizji na stanowisku spawalniczym

Istotne jest także poprawne ustawienie zmiennych systemowych *UTOOL* oraz *UFRAME*. Pierwsza z nich odpowiada za punkt *TCP* robota, natomiast druga określa układ współrzędnych, w jakim porusza się robot. Ważne jest, aby takie same zmienne były ustawione w rzeczywistym kontrolerze robota oraz na stanowisku wirtualnym. Zmienne te definiowane są w module *Robot OLP*. Również w tym samym module wygenerowany zostaje kod sterujący robotem.

Ze względu na specyfikę wykorzystywanego kontrolera *RJ2*, wygenerowany w *Delmii* kod w postaci pliku tekstowego **.LS* został przekonwertowany do formatu pliku binarnego **.TP*. W tym celu skorzystano z programu *Fanuc RJ Program Translator*. Do skopiowania programu do kontrolera wykorzystano komputer połączony z kontrolerem przez port *RS232C* oraz emulator stacji dyskiety – program *KFLOPPY*. Po przesłaniu programu do kontrolera należy jeszcze przetestować go na stanowisku w trybie pracy krokowej. Gdy wszystko działa jak należy, można uruchomić robota w trybie pracy automatycznej.

4. Podsumowanie

Programowanie off-line robotów przemysłowych jest metodą doskonale sprawdzającą się zarówno przy wdrażaniu nowych stanowisk zrobotyzowanych, jak również przy modernizacji stanowisk już istniejących. Może być realizowane równoległe z budową nowego stanowiska. W przypadku modernizacji stanowisk produkcyjnych zostaje znacznie skrócony czas wyłączenia stanowiska z produkcji. Do niezbędnego minimum ogranicza się czas przebywania programisty w nierzadko szkodliwym środowisku hali produkcyjnej.

Dostępność rozbudowanej biblioteki robotów oraz możliwości korzystania z baz danych producentów oprzyrządowania, jak również możliwość samodzielnego modelowania komponentów stanowiska, pozwala na zbudowanie wirtualnego modelu odzwierciedlającego istotne cechy stanowiska. Dzięki temu wiele testów można przeprowadzić w środowisku wirtualnym. Zmniejsza to liczbę testów wykonywanych na stanowisku fizycznym, co pozwala zaoszczędzić produkty, zasoby oraz bardzo cenny czas. Na stanowisku fizycznym wykonywane są tylko testy

końcowe, które mogą wykazać błędy nie wykryte przez symulację, spowodowane niezgodnością modelu ze stanowiskiem rzeczywistym.

W branży krążą prawdziwe historie związane z błędnie wykonanymi instalacjami i projektami. Problemem jest konieczność zmiany położenia robota, bądź też korekta wymiarów lub kształtu końcówki chwytnej. O wiele gorsza w skutkach jest konieczność wymiany robota lub też zmiana koncepcji całego stanowiska. Dzięki możliwościom analizy i optymalizacji stanowiska w środowisku wirtualnym, ryzyko popełnienia takiego błędu ulega istotnemu zmniejszeniu.

Wykorzystywany w pracy system *Delmia* udostępnia funkcje ułatwiające weryfikację działania stanowiska. Istotne, przedstawione w tekście pracy funkcjonalności, to: wykrywanie kolizji, wyznaczanie przestrzeni roboczej, sprawdzanie zasięgu. Dodatkowo możliwe jest np.: wyznaczenie czasu trwania zadania, obliczenie mas i momentów bezwładności brył obciążających ramię robota.

W trakcie pracy z systemem *Delmia* natknięto się na kilka błędów. W ocenie autorów przyczyną ich występowania jest fakt, że jest to oprogramowanie bardzo rozbudowane i stosowane od niedawna. Zwiększająca się liczba użytkowników oraz ciągły rozwój tego oprogramowania z pewnością będą służyć jego doskonaleniu.

5. Literatura

- [1] E. Dobaj: *Maszyny i urządzenia spawalnicze*, Warszawa 2005, WNT
- [2] A. Słota, T. Więk: *Application of Catia V5 for robots modeling and analysis*, 3rd International Conference Optimization of the Robots and Manipulators, OPTIROB 2008, Predal, 30 May – 1 June 2008
- [3] Z Series Kawasaki Robot Instalation and Connection Manual: instrukcja dostarczona przez producenta
- [4] http://ol.cadfamily.com/delmia/online/DELMIA_default.htm