

Politechnika Wrocławskiego
Wydział Elektroniki, Fotoniki i Mikrosystemów

KIERUNEK: Automatyka i Robotyka (AIR)
Specjalność: Robotyka

**PRACA DYPLOMOWA
INŻYNIERSKA**

Tytuł pracy:
Sterowanie manipulatorem FANUC
w oparciu o system wizyjny iRVision
i środowisko FANUC Roboguide

AUTOR:
Wojciech Kierbić

PROMOTOR:
Dr inż. Katarzyna Zadarnowska

Spis treści

1 Wprowadzenie	3
1.1 Cel i zakres pracy	4
2 Cela w środowisku FANUC Roboguide	5
2.1 Manipulatory	6
2.1.1 LR Mate 200iC	6
2.1.2 LR Mate 200iD/4S	7
2.2 System wizyjny iRVision	8
2.3 Środowisko symulacyjne FANUC Roboguide	8
2.4 Zebranie informacji o celi	9
2.5 Cela w FANUC Roboguide	9
2.5.1 Tworzenie nowej celi	9
2.5.2 Projektowanie celi w FANUC Roboguide	18
2.5.3 Manipulatory - UserFrame i UToolFrame	24
2.5.4 Kamery w FANUC Roboguide	26
2.5.5 Projektowanie celi - efekt	27
3 FANUC iRVision	29
3.1 Konfiguracja kamery w FANUC Roboguide	29
3.2 Konfiguracja kamery w iRVision	32
3.3 Kalibracja kamery w iRVision	33
3.4 Tworzenie procesu wizyjnego	37
3.5 Ustawienie pozycji referencyjnej obiektu	41
4 Program testowy	42
4.1 Program dla LR Mate 200iC w FANUC Roboguide	42
4.2 Program dla rzeczywistego LR Mate 200iC	44
4.3 Program dla symulatora i rzeczywistego manipulatora - różnice	45
4.4 Opis szczegółowy programu	46
4.4.1 Inicjalizacja programu	46
4.4.2 Główna część programu	46
4.5 Efekty pracy programu	48
4.6 Rozszerzenie programu testowego	49
5 Podsumowanie	54
Spis rysunków	55
Literatura	57

Rozdział 1

Wprowadzenie

Pierwszą wzmianką dotyczącą wprowadzenia robotów do codziennego użytkowania człowieka było stworzenie przez arabskiego naukowca Al-Jazariego na początku XIII wieku mechanicznej łodzi z czterema robotami imitującymi muzyków, które za cel miały zabawianie gości dźwiękami podczas królewskich uroczystości [1]. Jak pokazuje historia, takie prymitywne wynalazki mogą być podstawą do zainspirowania innych inżynierów do ulepszenia bądź stworzenia czegoś zupełnie nowego, własnego. Skutkiem takich działań były rewolucje przemysłowe. Człowiek, aby siebie odciążyć, konstruował urządzenia napędzane również ludzką pracą, lecz o wiele lżejszą. Następnie zastąpiono go mechanizmami zasilanymi silnikami parowymi, aby ostatecznie dojść do ery elektryczności [2]. Szukano rozwiązań tańszych, szybszych, bardziej precyzyjnych i bezpiecznych.

Jednym z etapów trwającej od końca II wojny światowej do dziś rewolucji przemysłowej była robotyzacja produkcji. Pierwszy prosty manipulator zbudowała firma Unimation. Nazywał się Unimate i jego zadaniem było powtarzanie niebezpiecznych zadań na linii produkcyjnej General Motors [3]. Manipulatory na początku wprowadzano do produkcji po to, aby przenosić obiekty z jednego punktu do drugiego z jak największą precyzją wykorzystując napędy hydrauliczne. Dopiero w 1969 roku manipulator sześciociosiowy Stanford Arm zbudowano tak, by jego wszystkie osie były napędzane elektrycznie. Zaraz po tym, zaczęto wykorzystywać manipulatory w zadanach takich jak montaż czy spawanie. Późniejsze lata 70. poprzedniego stulecia były czasem rozwитku firm zajmujących się robotyką przemysłową m.in. General Motors, która wraz z japońską firmą FANUC utworzyła spółkę joint venture - FANUC Robotics.

Wraz z rozwojem przemysłu pojawiła się możliwość uczenia manipulatorów rozpoznawania obiektów za pomocą systemów wizyjnych. Umożliwiło to rozszerzenie zakresu zadań robotów w fabrykach o np. sprawdzanie wykonanych przez robota spawów czy usuwanie z linii produkcyjnej wadliwych produktów. Firma FANUC Robotics posiada swój dedykowany system wizyjny zwany iRVision. Rozwinęła się również możliwość projektowania cel produkcyjnych zdalnie za pomocą symulatorów tzw. *OFFLINE*, co znacznie ułatwiło pracę robotykom. Dzięki temu, przed uruchomieniem programów na rzeczywistych manipulatorach sprawdza się, jak roboty zachowają się w celach symulowanych. Praktyka ta znacznie zmniejsza ryzyko uszkodzenia elementów pobocznych czy też samego manipulatora. FANUC Robotics również oferuje takie środowisko symulacyjne - FANUC Roboguide.

1.1 Cel i zakres pracy

Celem pracy inżynierskiej jest odtworzenie w symulatorze FANUC Roboguide celi odzwierciedlającej rzeczywiste stanowisko robotyczne znajdujące się w Laboratorium Robotyki Politechniki Wrocławskiej (sala 010, budynek C-3). Dalej, z poziomu środowiska FANUC Roboguide należy skalibrować i wykorzystać system wizyjny iRVision w celu zaimplementowania procesu rozpoznawania elementów znajdujących się na stole roboczym. W symulatorze zostanie utworzony i przetestowany program wykorzystujący funkcjonujący w systemie iRVision proces rozpoznawania elementów do ich segregacji. Dalej nastąpi przeniesienie programu na rzeczywisty manipulator i sprawdzenie poprawności odwzorowania celi, wykorzystania systemu wizyjnego iRVision oraz zadania realizowanego przez manipulator. Dodatkowo, zaimplementowane zostaną programy dla dwóch współpracujących robotów, umożliwiające ich integrację. Zadaniem manipulatorów będzie rozpoznawanie obiektów, przekazywanie ich sobie nawzajem i wreszcie paletyzacja pobieranych elementów.

Układ pracy jest następujący. Rozdział drugi opisuje proces implementacji rzeczywistego stanowiska robotycznego w postaci celi w środowisku symulacyjnym Roboguide. Trzeci rozdział poświęcony jest systemowi iRVision - jego konfiguracji, kalibracji, procesowi rozpoznawania obiektów. W rozdziale czwartym zaprezentowany zostanie przykładowy program stworzony w środowisku FANUC Roboguide, działający na rzeczywistych manipulatorach, wykorzystujący system iRVision w celu rozpoznawania elementów.

Rozdział 2

Cela w środowisku FANUC Roboguide

Stanowisko zrobotyzowane (rys. 1), które będzie odwzorowywane w niniejszej pracy znajduje się w laboratorium 010 budynku C-3 Politechniki Wrocławskiej. Jego wyposażenie składa się z dwóch robotów z serii LR Mate wyprodukowanych przez firmę FANUC Robotics: LR Mate 200iC oraz LR Mate 200iD. Oprócz manipulatorów w skład celi będą wchodzić:

- **Metalowe stoły** - na nich umieszczone są manipulatory. Przykręcione są do ziemi co zapewnia zarówno bezpieczeństwo, jak i dokładność pracy robotów.
- **Sterowniki robotów** - R-30iA dla LR Mate 200iC oraz R-30iB Mate dla LR Mate 200iD/4S. Umieszczone są na półkach pod blatem metalowych stołów.
- **Taśmociąg** - umożliwia zintegrowanie pracy manipulatorów. Działa, dzięki zastosowaniu sterownika PLC Siemens S7-1200. Ma zapewnione dwa tryby pracy: ręczny (za pomocą przycisków) oraz automatyczny (za pomocą wyjść cyfrowych).
- **Szafy na elementy RLC** - znajdują się tam rezystory, kondensatory i cewki o różnych wartościach.
- **Sprężarka powietrza** - zapewnia działanie chwytaków ssących stanowiących efektory manipulatorów.
- **Kamery** - dwie kamery monochromatyczne marki Sony typu XC-56. Służą jako system wizyjny manipulatorów.
- **Stanowisko komputerowe.**
- **Klimatyzator.**
- **Stół roboczy.**
- **Barierki bezpieczeństwa** - zapewniają bezpieczeństwo i ograniczają cele.
- **Ściany** - ograniczają cele.

2. Cela w środowisku FANUC Roboguide



Rysunek 1: Stanowisko zrobotyzowane w sali 010

2.1 Manipulatory

Roboty z serii LR Mate są na tyle wszechstronne, że znacznie zwiększały produktywność poleconych im zadań [4]. Wielkość manipulatorów można porównać do rozmiaru ludzkiego ramienia. Poszczególne egzemplarze mogą różnić się od siebie wagą, zasięgiem, udźwigiem czy prędkością w przegubach. Wykorzystywane są w przemyśle spożywczym i metalowym. LR Mate oferuje szeroki zakres usług dodatkowych m.in. rozpoznawanie obiektów dzięki systemowi wizyjnemu iRVision czy uwzględnianie siły reakcji dzięki czujnikowi siły. Dodatkowo możliwe jest zamontowanie na robocie różnego rodzaju efektorów/końcówek roboczych: do spawania, malowania, podnoszenia czy cięcia.

2.1.1 LR Mate 200iC

FANUC LR Mate 200iC (rys. 2) to 6 osiowy manipulator, który może pracować na dużej prędkości [5]. Działa dzięki sterownikowi R-30iA Mate. Jego zasięg to 704 mm. Może podnieść obiekt ważący nawet do 5kg, sam waży przy tym 27 kg. Jest bardzo podobny do innego z tej serii robota 200iD, lecz wygrywa z nim pod względem prędkości pracy i dokładności.

2. Cela w środowisku FANUC Roboguide



Rysunek 2: LR Mate 200iC

2.1.2 LR Mate 200iD/4S

FANUC LR Mate 200iD/4S (rys. 3) różni się od typu 200iC możliwością udźwigu, gdyż potrafi podnieść do 4 kg obciążenia [6]. Jego zasięg jest krótszy o 154 mm, a waga mniejsza o 7 kg, co czyni go jeszcze bardziej kompaktowym manipulatorem. Natomiast łączy je ta sama liczba osi. Jego sterowanie zapewnia kontroler R-30iB Mate.



Rysunek 3: LR Mate 200iD/4S

2. Cela w środowisku FANUC Roboguide

2.2 System wizyjny iRVision

Zastosowanie systemu wizyjnego iRVision umożliwia wykrywanie zarówno dwu-, jak i trójwymiarowo [7]. Należy wspomnieć, iż system iRVision współpracuje z każdym manipulatorem FANUC niezależnie od jego masy, rozmiaru, ilości osi czy udźwigu. Istnieje kilka funkcjonalności związanych z systemem wizyjnym:

- **2D vision** - wykrywanie obiektów w dwóch wymiarach.
- **2 $\frac{1}{2}$ D vision** - zbliżone do samego 2D, lecz umożliwia wykrywanie obiektów warstwowo.
- **3D laser vision** - detekcja obiektów w 6 stopniach swobody.
- **3D Are Sensor and 3D Vision Sensor** - Podobnie do zwykłej detekcji 3D laser vision, lecz używa się jej w trudniejszych warunkach np. elementy są zabrudzone lub zardzewiałe.
- **iRPickTool** - używane, kiedy elementy znajdują się na ruchomych platformach.
- **iRCalibration** - ułatwia kalibrowanie systemów wizyjnych.
- **iRVision Weld Tip Inspection/iRTorchMate** - kontrola elementów po spawaniu.

2.3 Środowisko symulacyjne FANUC Roboguide

FANUC Roboguide [8] to bardzo popularne środowisko symulacyjne służące do programowania robotów offline, czyli bez kontaktu z rzeczywistym sprzętem. Pakiety FANUC Roboguide umożliwiają użytkownikom projektowanie, oprogramowanie i testowanie zrobotyzowanych cel/komórek bez ponoszenia kosztów związanych z fizycznymi stworzeniem stanowiska. Największą zaletą używania FANUC Roboguide jest możliwość wcześniejszego przetestowania napisanego programu, a przy tym zmniejszenie ryzyka wystąpienia jakiejkolwiek kolizji robota z innym robotem lub otoczeniem. Symulator zaopatruje użytkownika w dodatkowe narzędzia, takie jak:

- **HandlingPRO** - najpopularniejsze, służące do załadunku/rozładunku, pakowania, montażu i usuwania obiektów
- **PaintPRO** - malowanie elementów
- **PalletPRO** - aplikacje paletyzujące
- **WeldPRO** - spawanie łukowe.

W pracy inżynierskiej wykorzystano wersję 9 symulatora (komputer własny) oraz wersję 8.1 (komputer w Laboratorium Robotyki Politechniki Wrocławskiej).

2. Cela w środowisku FANUC Roboguide

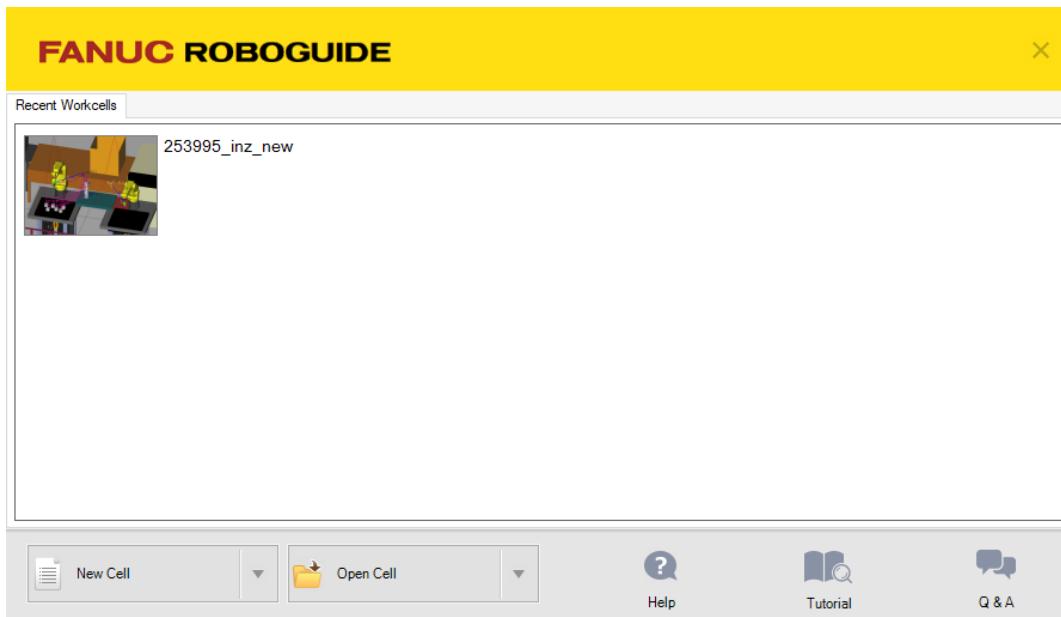
2.4 Zebranie informacji o celi

Przed przystąpieniem do budowania celi w FANUC Roboguide należy zebrać o niej podstawowe informacje. Jedną z nich są elementy znajdujące się w obrębie odwzorowywanego stanowiska (str. 5). Pełnią one zarówno funkcje użytkowe, jak i kosmetyczne. Należy bardzo dokładnie zmierzyć części będące w otoczeniu robota, do których ma on zasięg. Bywa to na tyle istotne, że programista musi zapewnić bezkolizyjność robota podczas pracy. Nie może również zabraknąć najważniejszego, czyli manipulatorów (str. 7). Każdy manipulator charakteryzuje się innymi wymiarami, dlatego należy wiedzieć jakie modele znajdują się w obrębie zrobotyzowanego stanowiska. Oczywiście, programu nie uruchamia się w trybie automatycznym od razu, bo zawsze istnieje możliwość pomyłki, a co za tym idzie - korekcja tych błędów. Wszystkie powyższe informacje zebrano i uwzględniono w projekcie komórki.

2.5 Cela w FANUC Roboguide

2.5.1 Tworzenie nowej celi

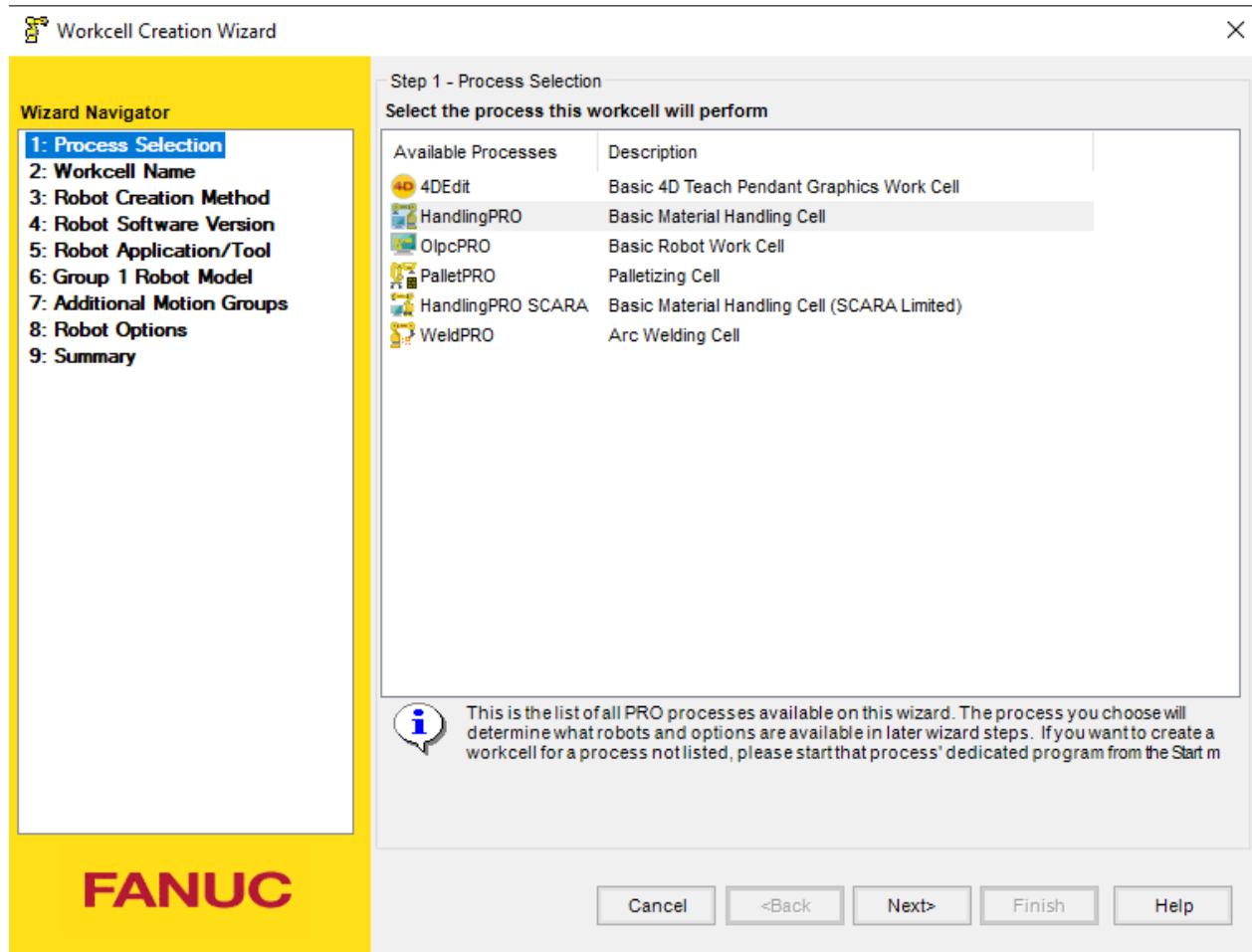
Projektowanie nowej celi w programie FANUC Roboguide rozpoczyna się od uruchomienia programu z poziomu komputera. Następnie na ekranie można zobaczyć okno startowe (rys. 4), w którym widnieją utworzone już wcześniej projekty z możliwością ich otwarcia lub stworzenia nowego. Dodatkowo producent zapewnił opcję samouczka dla początkujących lub kontaktu w razie problemów.



Rysunek 4: FANUC Roboguide - ekran startowy

2. Cela w środowisku FANUC Roboguide

Będąc na tym etapie należy wybrać opcję *New Cell*. Rozpocznie ona proces tworzenia nowej zrobotyzowanej komórki w symulatorze (patrz rys. 5):

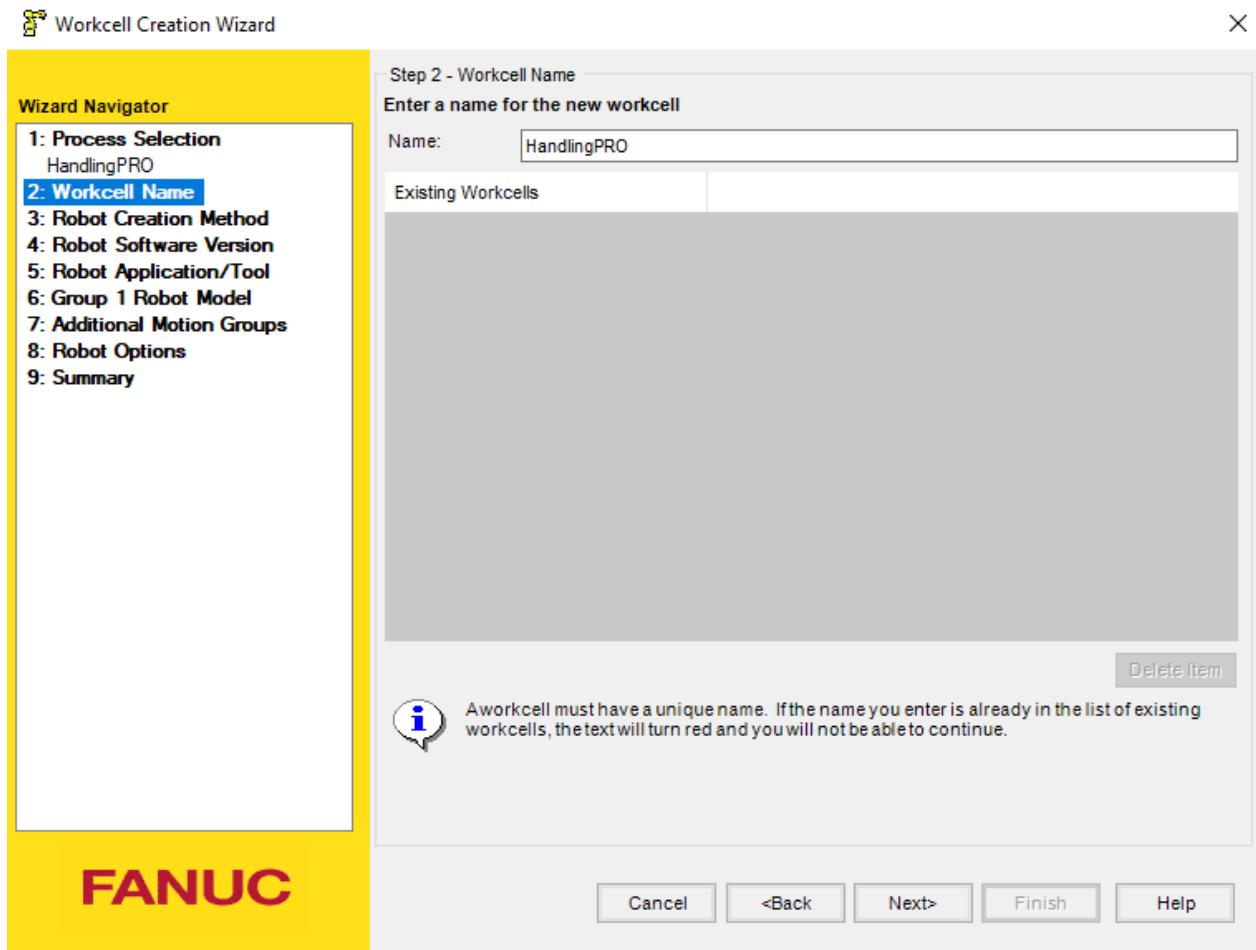


Rysunek 5: FANUC Roboguide - wybór procesu

Na powyższym rysunku (rys. 5) przedstawiono procesy oferowane przez użytkowaną wersję zainstalowanego symulatora. Wybór należy dostosować do potrzeb wykonywanego przez manipulator zadania. Przykładowo, jeżeli robot będzie wykonywał spawanie, wybór pada na proces WeldPRO o czym wspomniano na str. 8. Najczęściej wybierana jest opcja HandlingPRO, ponieważ przenoszenie elementów to powszechnie zadanie dla robotów. W pracy inżynierskiej również taki proces został wybrany. Po dokonaniu selekcji procesu, należy nacisnąć przycisk *Next*, w celu dalszego tworzenia komórki symulacyjnej.

2. Cela w środowisku FANUC Roboguide

Kolejną częścią tworzenia nowej celi jest ustalenie nazwy projektu (rys. 6). Bywa to na tyle istotne, iż zazwyczaj proces produkcyjny składa się z więcej niż jednej komórki robotycznej i każda z nich powinna być konkretnie nazwana przy ewentualnych korektach. Podstawowa nazwa nowego projektu zaczyna się od wybranego na poprzednim etapie procesu i ewentualnej cyfry, jeżeli nomenklatura będzie się powtarzać.



Rysunek 6: FANUC Roboguide - nazwa celi

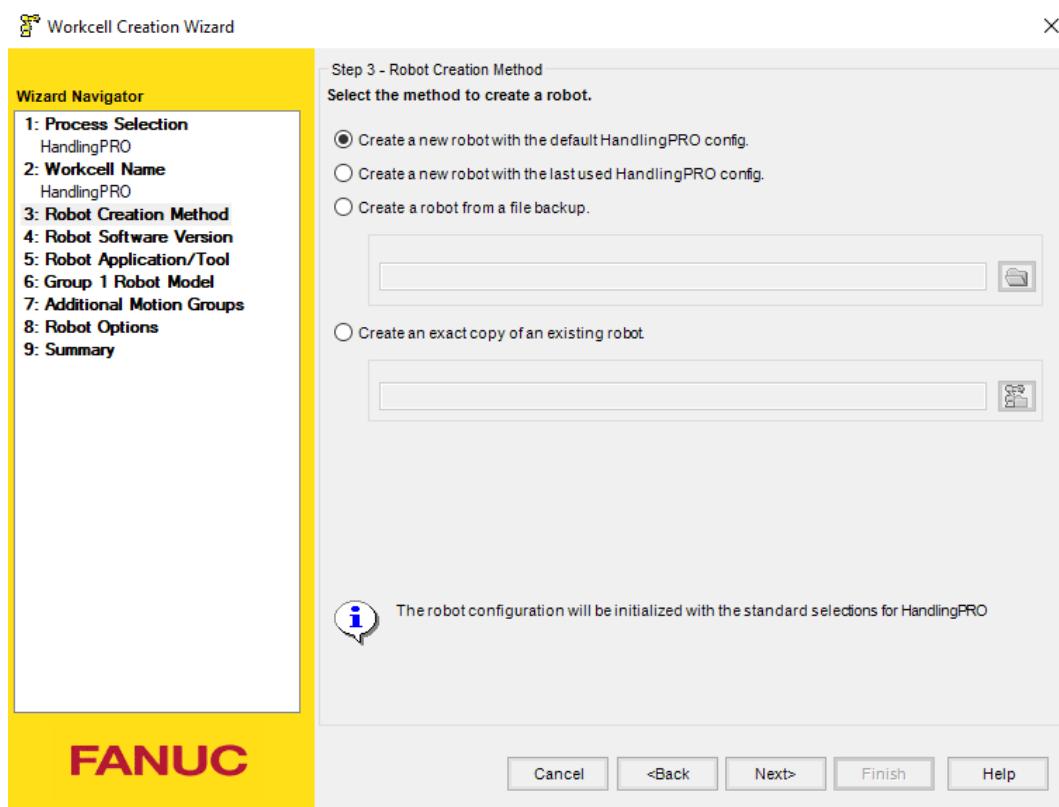
Po nadaniu charakterystycznej nazwy dla danej celi, należy nacisnąć przycisk *Next*, aby przejść dalej.

2. Cela w środowisku FANUC Roboguide

FANUC Roboguide oferuje trzy możliwości dodania robota do danej celi (rys. 7):

- **Create a new robot with the default HandlingPRO config** - stworzenie całkowicie nowej konfiguracji dla robota w celi, w tym efektor, dodatki czy oprogramowanie.
- **Create a new robot with the last used HandlingPRO config** - dodanie robota do celi wykorzystując poprzednio użytą konfigurację. Bardzo przydatne, kiedy dodawane manipulatory są identyczne lub podobne konfiguracyjnie.
- **Create a robot from a file backup** - manipulatory mogą stworzyć plik z ich własną kopią zapasową. Wtedy taki plik z rozszerzeniem *.dt* należy wybrać i można robota dodać do celi symulacyjnej.
- **Create an exact copy of an existing robot** - tzw. *RobotNeighbourhood*. Możliwe jest dodanie manipulatora bezpośrednio do projektu, jeżeli jest on połączony do tej samej sieci, co komputer, na którym znajduje się FANUC Roboguide.

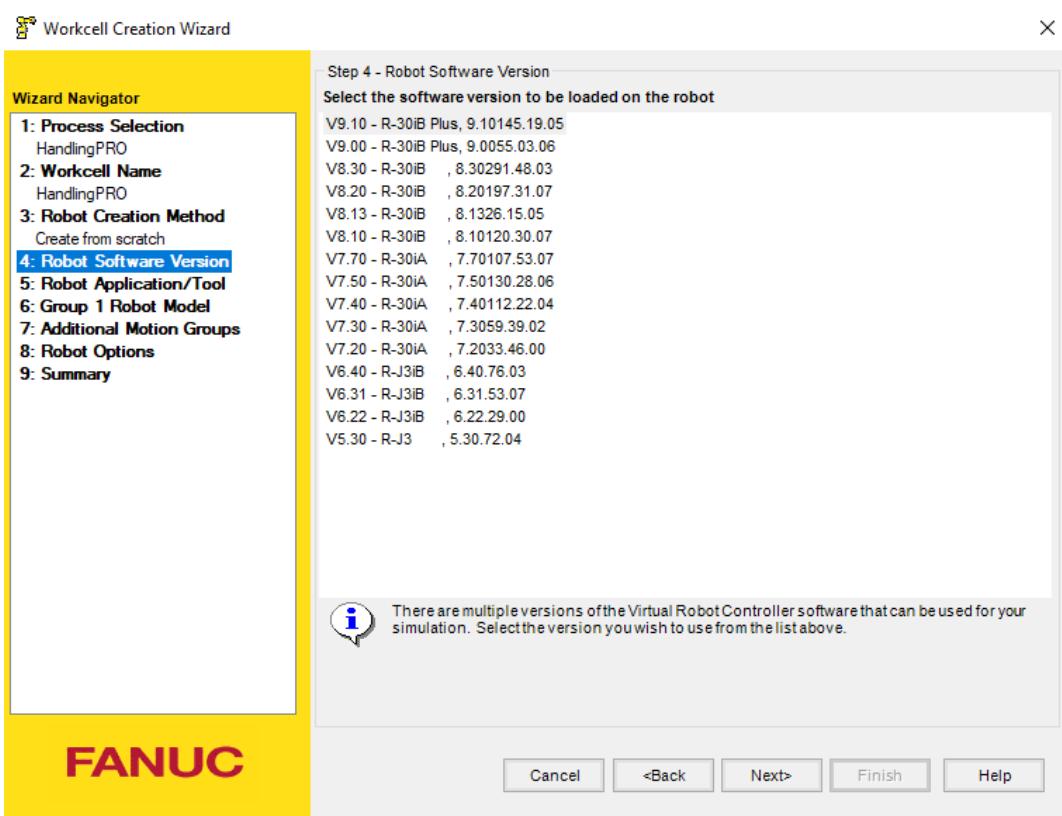
Używając dwóch ostatnich opcji tj. dodanie robota z kopii zapasowej i z *RobotNeighbourhood* nie można edytować dalszych narzędzi. Należy przejść wtedy od razu do podsumowania.



Rysunek 7: FANUC Roboguide - metody dodawania robota

2. Cela w środowisku FANUC Roboguide

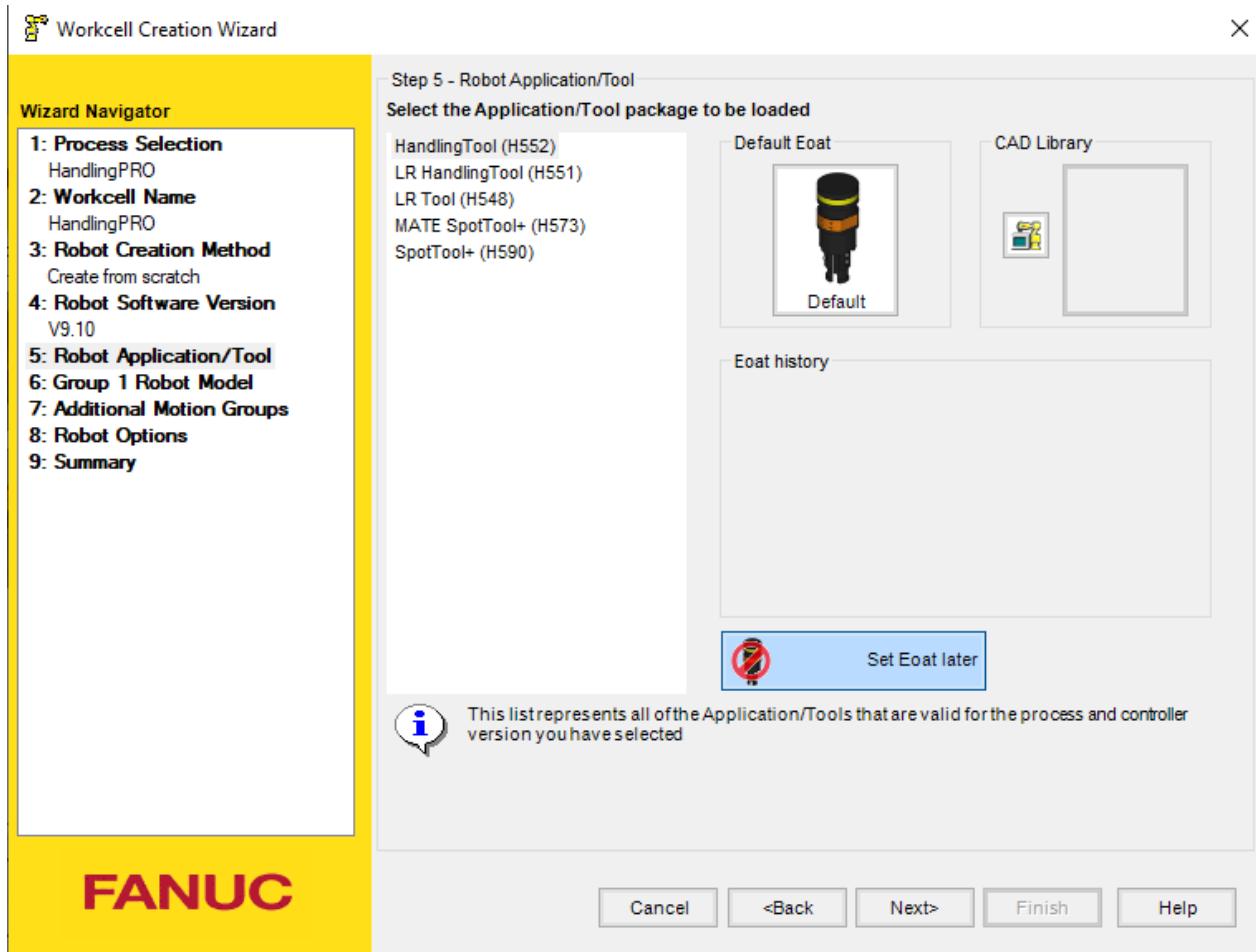
Następna zakładka (rys. 8) odpowiada za wybór wersji oprogramowania, a co za tym idzie - rodzaju sterownika, który ma zarządzać pracą danego manipulatora. W laboratorium robotyki manipulator LR Mate 200iD/4S posiada kontroler w wersji R-30iB, dlatego należałoby wybrać którąś z wersji: V8.30, V8.20, V8.13 lub V8.10. Jednak dzięki zastosowaniu *RobotNeighbourhood* lub kopii zapasowej nie trzeba się tym przejmować, gdyż FANUC Roboguide sam przyporządkuje poprawną wersję. Po wybraniu sterownika należy nacisnąć przycisk *Next* i przejść dalej.



Rysunek 8: FANUC Roboguide - wybór oprogramowania robota

Manipulatorowi w kolejnym etapie można przyporządkować efektor (rys. 9). Baza biblioteki CAD w FANUC Roboguide jest bogata w różnego rodzaju końcówki robocze, od zwykłych chwytaków palczastych po chwytaki ssące z kilkoma przyssawkami. Jeżeli żadna z opcji nie będzie odpowiednia, zawsze można dodać własny model efektora. Ponadto, na tym etapie simulator nie wymaga od użytkownika założenia jakiegokolwiek wykończenia na manipulator i można to zrealizować później.

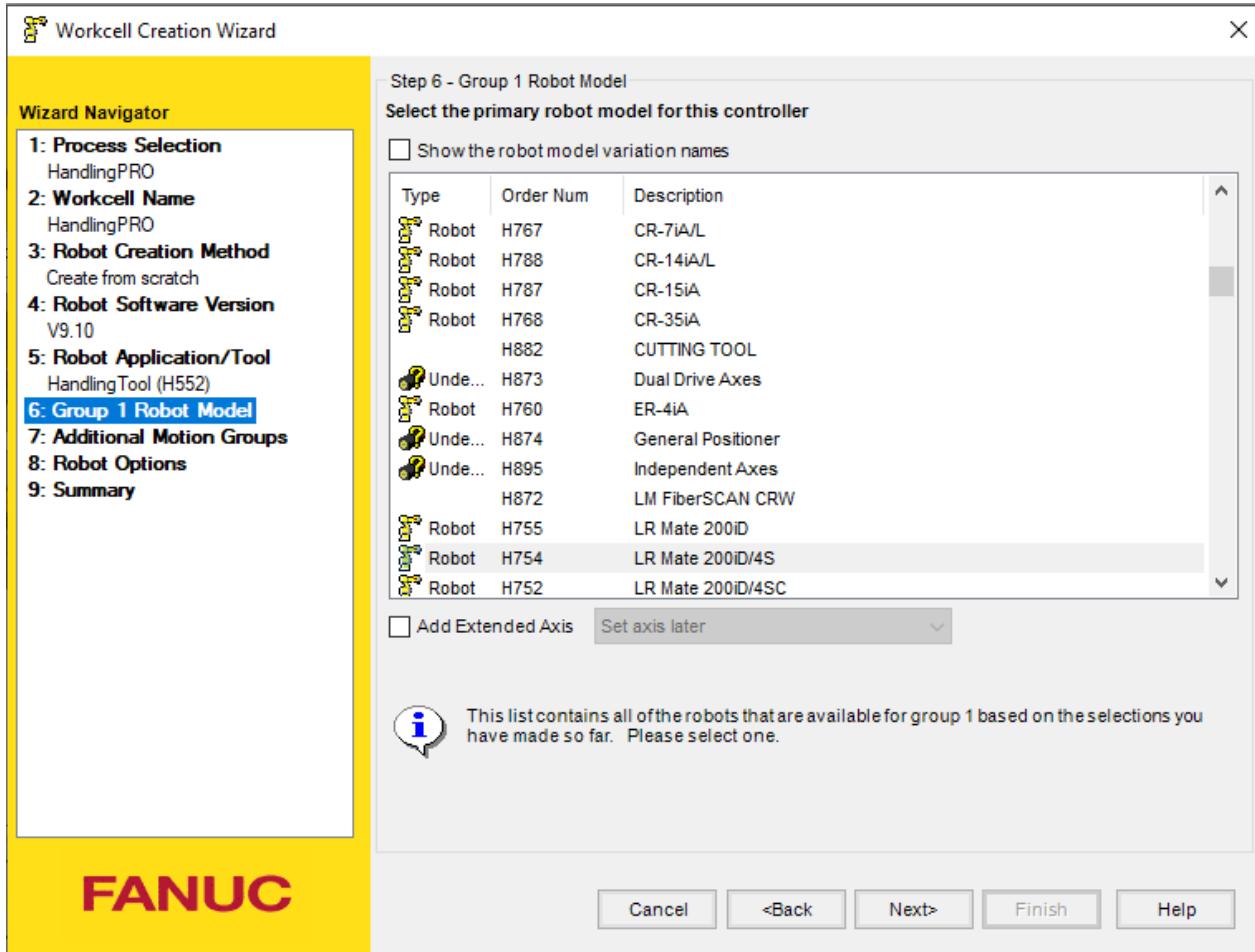
2. Cela w środowisku FANUC Roboguide



Rysunek 9: FANUC Roboguide - wybór efektora robota

W drugiej zakładce została wybrana opcja dodania nowej konfiguracji manipulatora. Następna (rys. 10) będzie umożliwiała wybór robota, który ma zostać dodany do celi. Ich liczba w symulatorze jest bardzo duża. Roboty FANUC różnią się od siebie wielkością, zasięgiem czy udźwigiem. Tak bogaty wachlarz możliwości, który oferuje producent, pozwala na dobranie do celi odpowiednich modeli, co jest kolejną zaletą stosowania symulatora - możliwość przetestowania manipulatorów pod względem wydajności przy realizacji tego samego zadania.

2. Cela w środowisku FANUC Roboguide

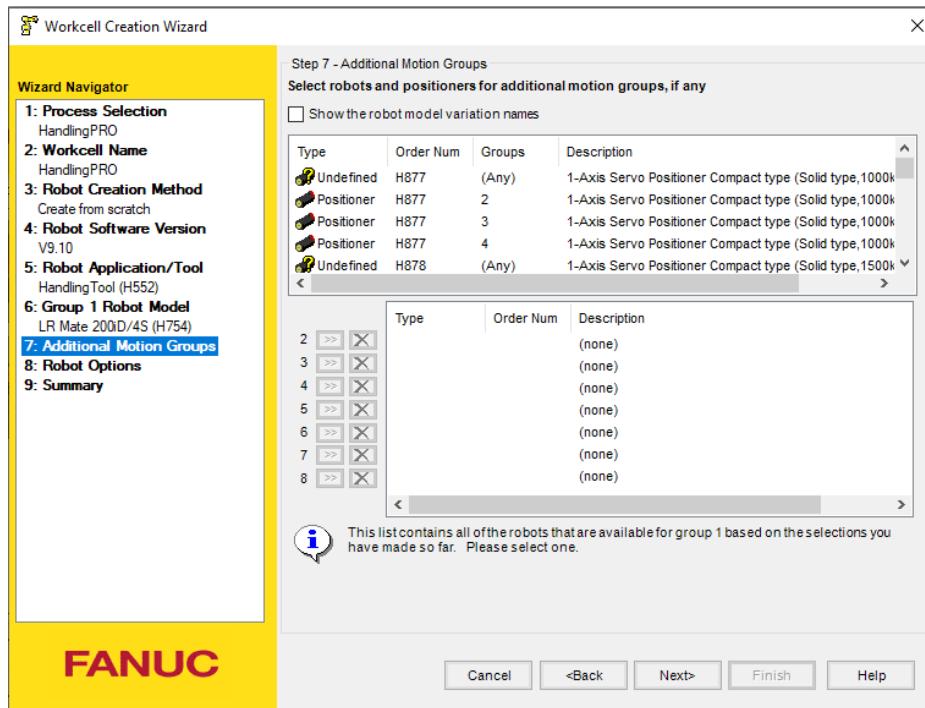


Rysunek 10: FANUC Roboguide - wybór robota

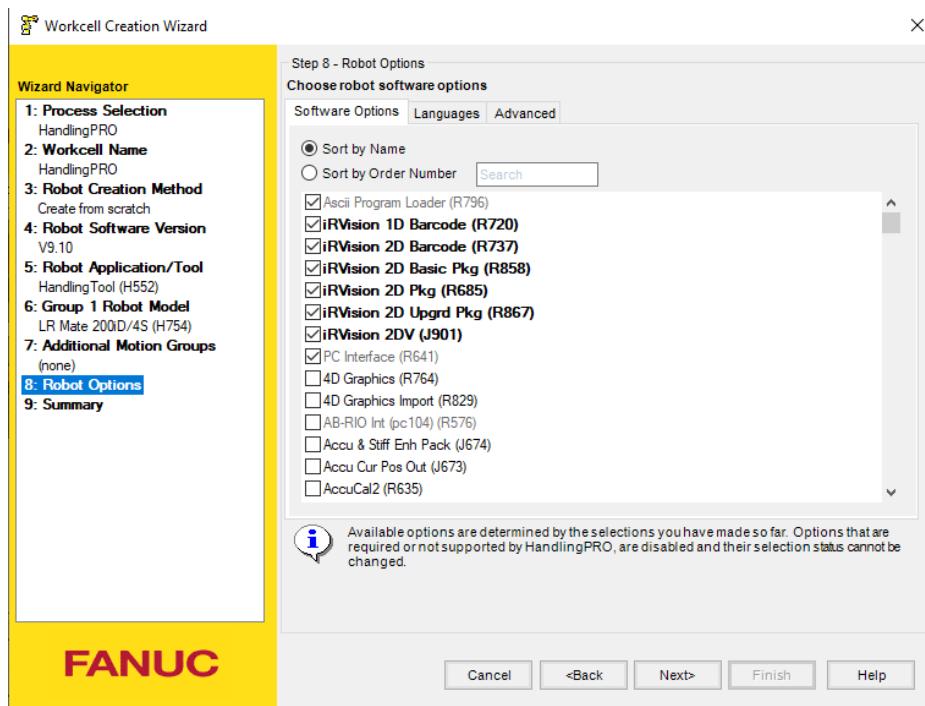
Zakładka 7. podczas tworzenia nowej celi umożliwia dodanie dodatkowych grup ruchu. Pojedyncza grupa składa się z zestawu silników i osi, które są połączone w celu wykonywania ruchów [9]. Roboty marki FANUC są przykładem zazwyczaj grupy 1. Dodatkowe grupy ruchu to osie sterujące na elemencie wyposażenia innym niż robot. Maksymalna liczba, którą można dodać w tej zakładce (rys. 11) to 8.

Natomiast ostatnia już część tworzenia nowej celi (rys. 12), z możliwością wprowadzania własnych ustawień to opcje robota. Są to wszystkie dodatkowe elementy oprogramowania, które manipulatory marki FANUC mogą posiadać. W pracy inżynierskiej jednym z celów jest implementacja systemu wizyjnego, w związku z czym rysunek 12 prezentuje wszystkie niezbędne części do jego realizacji (zaznaczone odpowiednie pola wyboru).

2. Cela w środowisku FANUC Roboguide



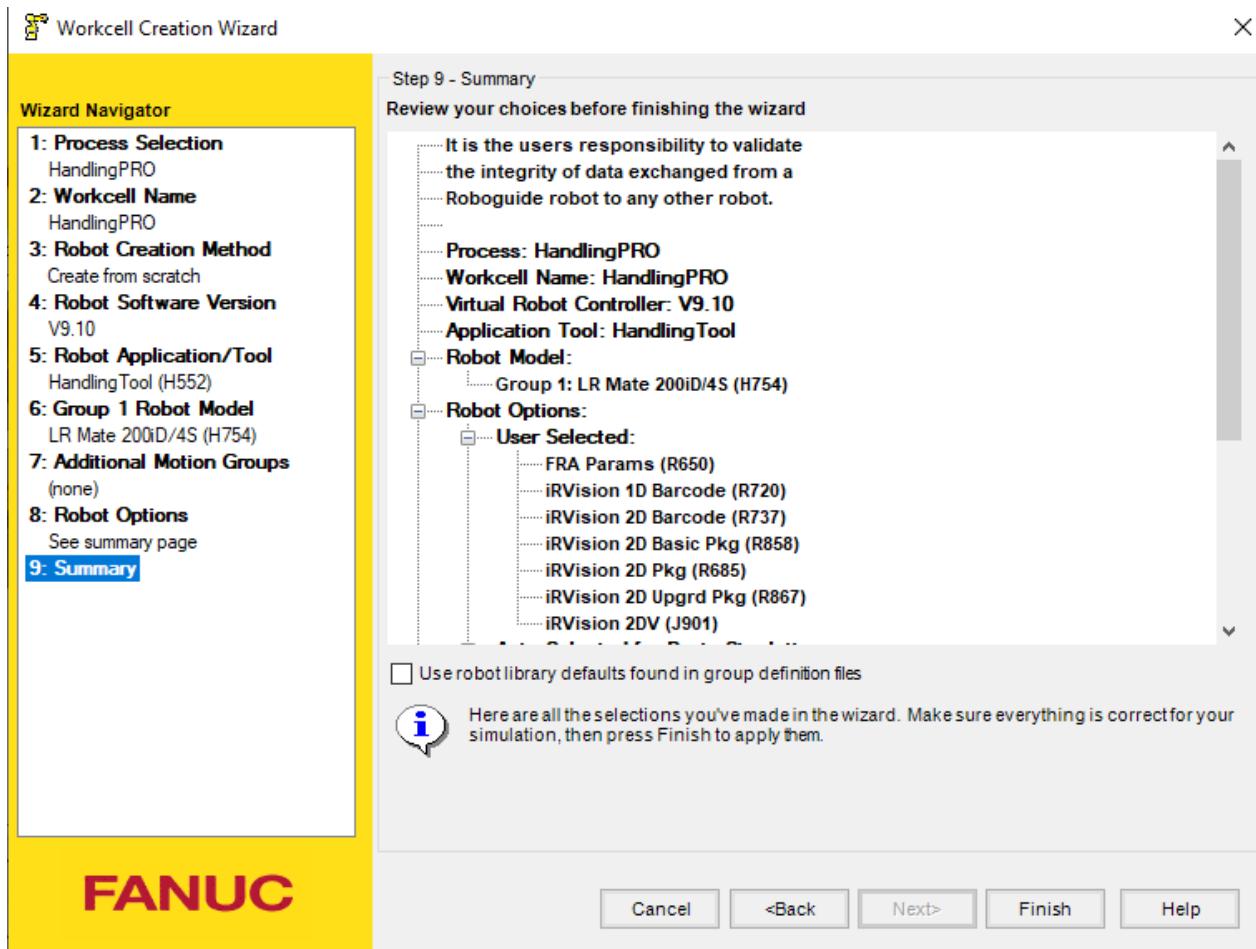
Rysunek 11: FANUC Roboguide - dodatkowe grupy ruchów



Rysunek 12: FANUC Roboguide - dodatkowe opcje robota

2. Cela w środowisku FANUC Roboguide

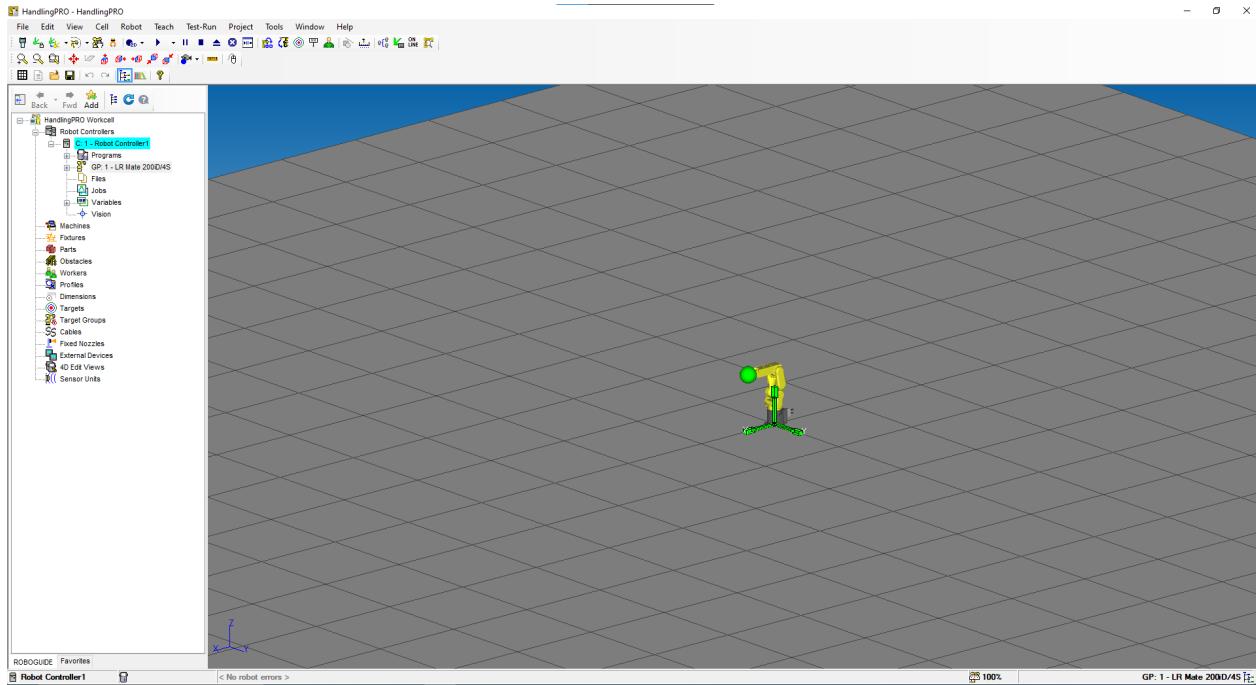
Po dokonaniu wszystkich wyborów pojawia się okno z podsumowaniem (rys. 13). W nim powinny znajdować się wszystkie zaznaczone wcześniej opcje: proces, nazwa celi, wersja kontrolera, wybrany efektor, wybrany robot i jego dodatki. Należy sprawdzić czy wszystkie zaznaczone opcje są poprawne i zakończyć proces tworzenia nowego projektu przyciskiem *Finish*.



Rysunek 13: FANUC Roboguide - Podsumowanie

Kiedy zakończy się proces tworzenie nowej celi, oczom użytkownika powinna ukazać się pusta cela (rys. 14) z wybranym przez programistę robotem.

2. Cela w środowisku FANUC Roboguide

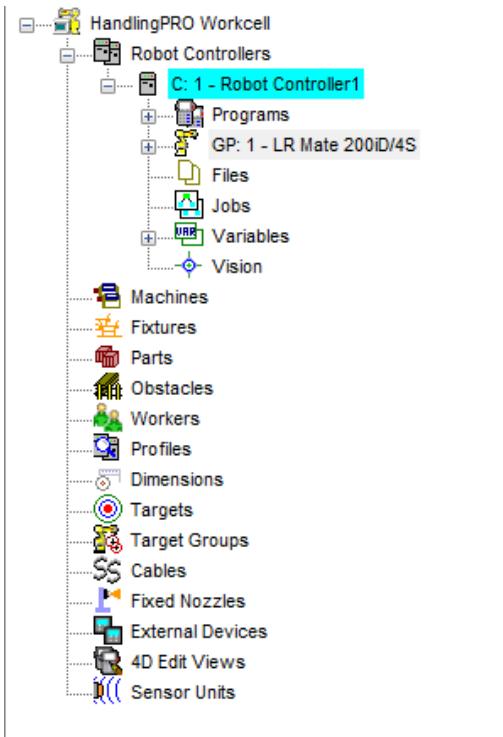


Rysunek 14: FANUC Roboguide - Pusta cela po stworzeniu projektu

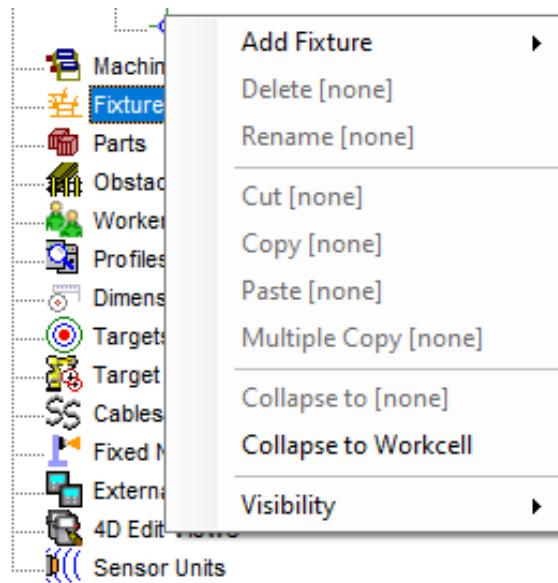
2.5.2 Projektowanie celi w FANUC Roboguide

Po utworzeniu nowego projektu następnym krokiem jest zwizualizowanie celi za pomocą modeli 3D. Wbudowana biblioteka CAD zapewnia użytkownikowi pokaźną liczbę elementów, które można zamieścić na planszy testowej. Oprócz gotowych elementów stworzonych w przeznaczonych do tego programach, FANUC Roboguide zapewnia także proste bryły: prostopadłościan, sześcian, kulę. Po lewej stronie znajduje się zakładka (rys. 15) z członami odpowiadającymi elementom celi. Nowa komórka z samym robotem będzie miała tylko jeden element w grupie *Robot Controllers*. Obiekty wypełniające przestrzeń roboczą znajdują się pod zakładką *Fixtures* i aby je dodać, należy nacisnąć na nie prawym przyciskiem oraz wybrać *Add fixture* (rys. 16).

2. Cela w środowisku FANUC Roboguide



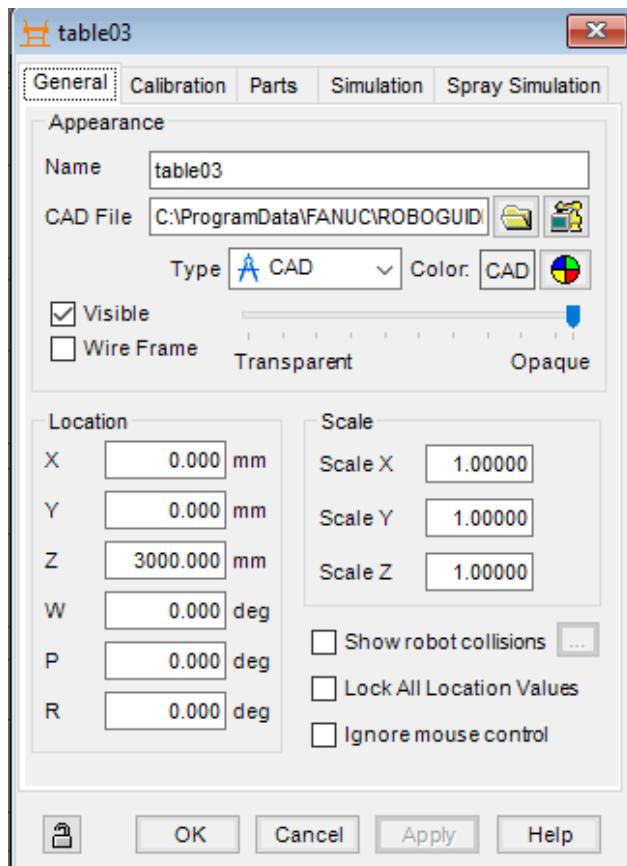
Rysunek 15: FANUC Roboguide - Zakładki w projekcie



Rysunek 16: FANUC Roboguide - Dodawanie obiektu

2. Cela w środowisku FANUC Roboguide

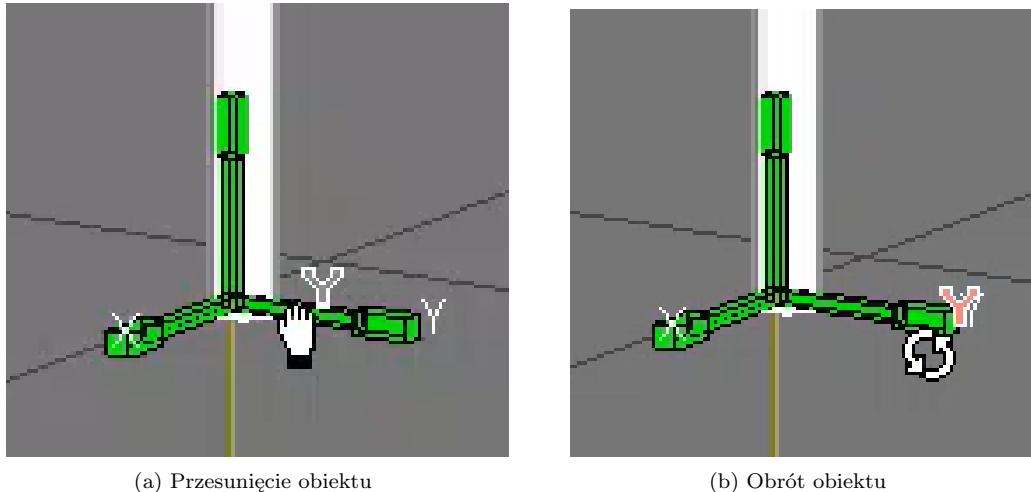
Wraz z dodaniem nowego przedmiotu, ukazuje się okno z jego właściwościami (rys. 17). Istnieje w nim opcja zmiany nazwy obiektu, lokalizacji w przestrzeni, wymiarów, widoczności czy kolizyjności z manipulatorem. Jeżeli obiekt jest z biblioteki CAD, to użytkownik może tylko przeskalać rozmiar, lecz nie będzie w stanie ustalić konkretnego. Z kolei proste bryły (prostopadłościan, sześcian, kula) mają taką możliwość.



Rysunek 17: FANUC Roboguide - Zmiana właściwości obiektu

Kiedy projektant zna dokładną pozycję lub orientację, na której chce, aby obiekt się znajdował, może wartości te wpisać w rubryki odpowiadające przemieszczeniu i obrotowi wzdłuż/wokół danych wektorów. FANUC Roboguide umożliwia programistę również przemieszczanie (rys. 18a) i obracanie ręczne (rys. 18b). Wykonuje się je wybierając wskaźnikiem konkretną oś układu współrzędnych obiektu.

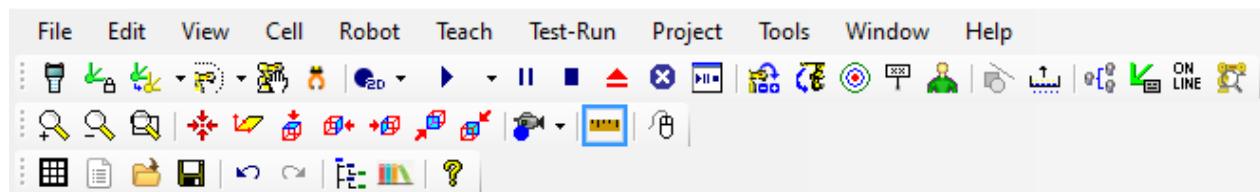
2. Cela w środowisku FANUC Roboguide



Rysunek 18: FANUC Roboguide - Ręczne przesunięcie i obrót obiektu

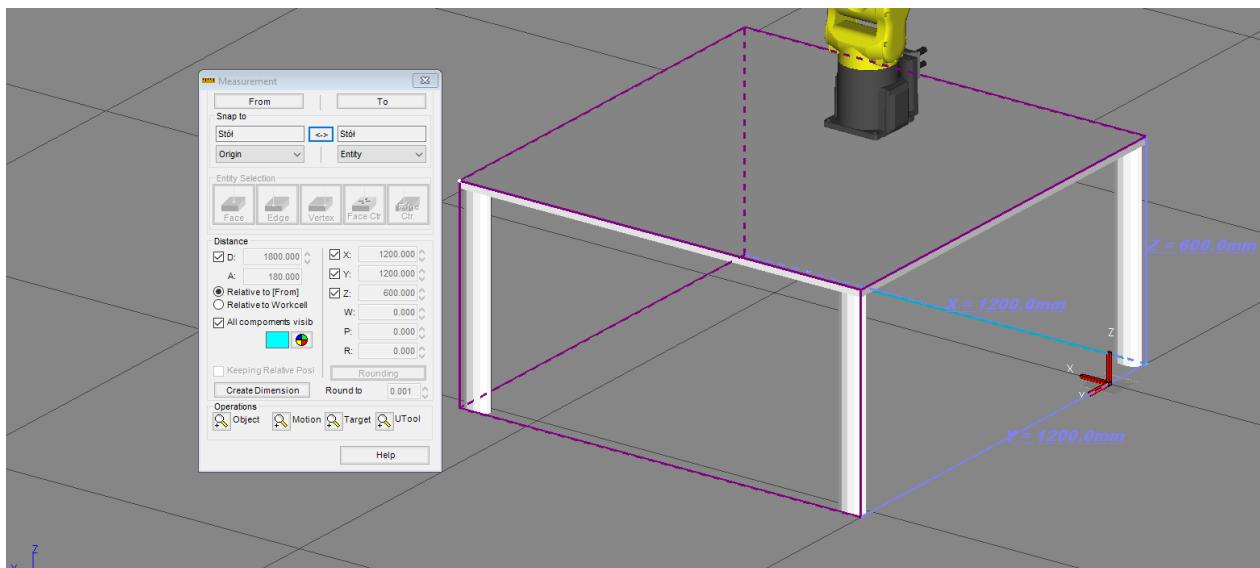
Jednym z udogodnień programu jest możliwość mierzenia przedmiotów. Przy odzwierciedlaniu fizycznego stanowiska zrobotyzowanego użycie tej opcji zdaje się być nieuniknione. Dzięki niej można przeskalać modele CAD do potrzebnych wymiarów. Dodatkowo, ułatwi to ustalenie obiektów w zmierzonych w celu odległościach pomiędzy przedmiotami. Warunki rzeczywiste weryfikują, że obiekty nie są tak idealne jak w symulatorze. Żeby perfekcyjnie odwzorować celę należałoby każdy fizyczny element zaprojektować, a następnie dodać do FANUC Roboguide. Jednak przy tak małych wymogach dokładności obiektów z biblioteki CAD były wystarczające.

Opcję mierzenia znaleźć można w narzędziach w górnej części symulatora (rys. 19, zaznaczony niebieskim kwadratem). Należy wybrać tam punkt początkowy *From* i końcowy mierzenia *To*. Oprócz pojedynczych punktów, może to być powierzchnia, krawędź, narożnik czy środek powierzchni. Po zmierzeniu ukażą się odległości wzdłuż wszystkich osi (rys. 20).



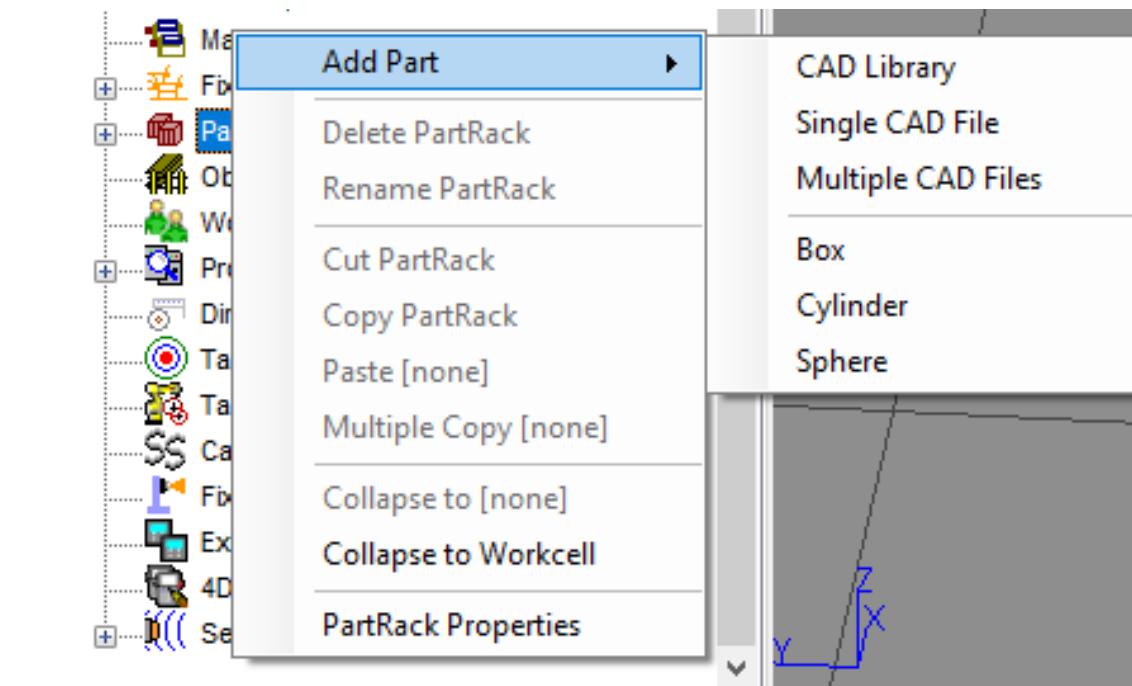
Rysunek 19: FANUC Roboguide - Wybór opcji mierzenia

2. Cela w środowisku FANUC Roboguide



Rysunek 20: FANUC Roboguide - Mierzenie obiektu

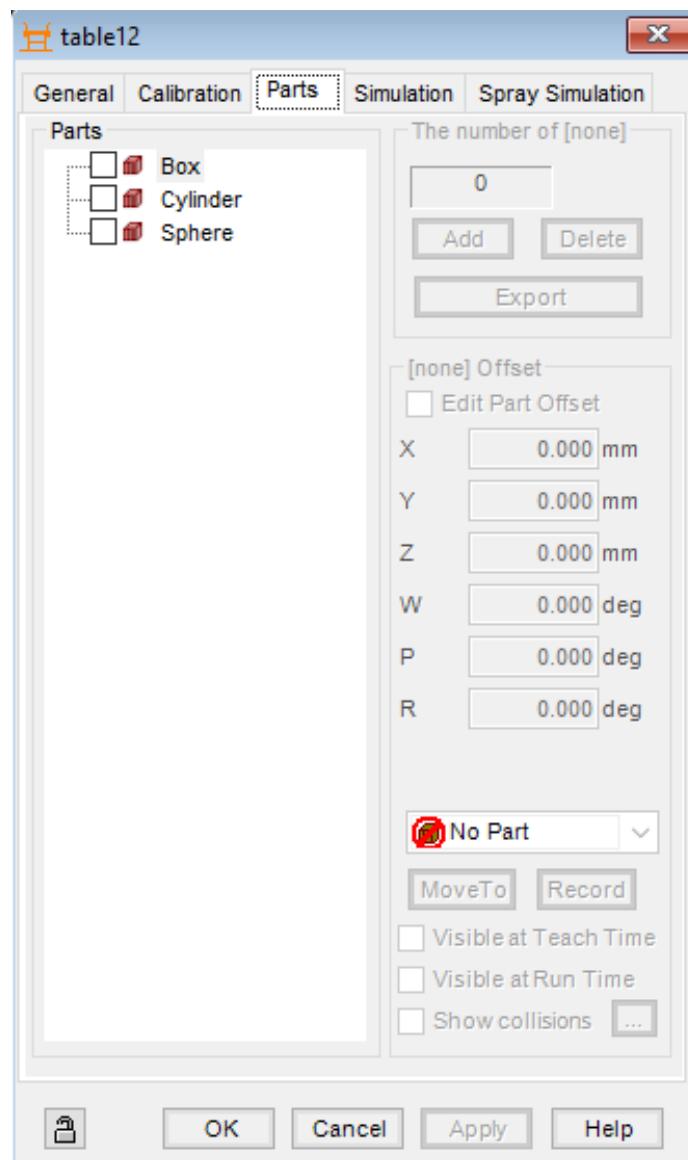
Oprócz dodawania zwykłego otoczenia istnieje zakładka *Parts* (rys. 21), której używa się do tworzenia animacji pracy celi. Działa ona w podobny sposób do opcji *Fixtures*. Tak samo można używać biblioteki CAD i prostych brył.



Rysunek 21: FANUC Roboguide - Dodawanie części

2. Cela w środowisku FANUC Roboguide

Każda taka część, która będzie brała udział w symulacji przenoszenia, spawania, malowania musi zostać przypisana danemu obiekowi (rys. 22). Przykładowo, mając element do przeniesienia z taśmociągu i odłożenia do pudełka, część trzeba zadeklarować w każdym miejscu, które bierze udział w procesie paletyzacji. Obiekt powinien zostać przypisany zarówno taśmociągowi, chwytkowi robota i pudełku. FANUC Roboguide umożliwi wówczas użytkownikowi stworzenie programu symulacyjnego do podnoszenia, przenoszenia i odkładania danego elementu.



Rysunek 22: FANUC Roboguide - Przypisanie części do obiektu

2. Cela w środowisku FANUC Roboguide

2.5.3 Manipulatory - UserFrame i UToolFrame

Koniecznym elementem programowania jest definiowanie układów współrzędnych, tzw. UserFrame i UToolFrame. UserFrame to układ współrzędnych definiowany bezpośrednio przez użytkownika. Dzięki tym ramkom tworzenie trajektorii manipulatora odbywa się bardziej intuicyjnie [10]. W warunkach rzeczywistych stworzenie UserFrame polega na zdefiniowaniu układu współrzędnych za pomocą jednej z trzech metod:

- **Three Point** - należy podjechać manipulatorem w trzy różne punkty na płaszczyźnie definiującą początek układu współrzędnych, kierunek i orientację osi X oraz kierunek i orientację osi Y.
- **Four Point** - bardzo zbliżony do metody Three Point. Dodatkowo umożliwia umieszczenie UserFrame w dowolnym miejscu w przestrzeni za pomocą punktu System Origin.
- **Direct Entry** - bezpośrednie zadawanie układów współrzędnych (za pomocą wartości liczbowych definiujących kierunki i orientacje poszczególnych osi)

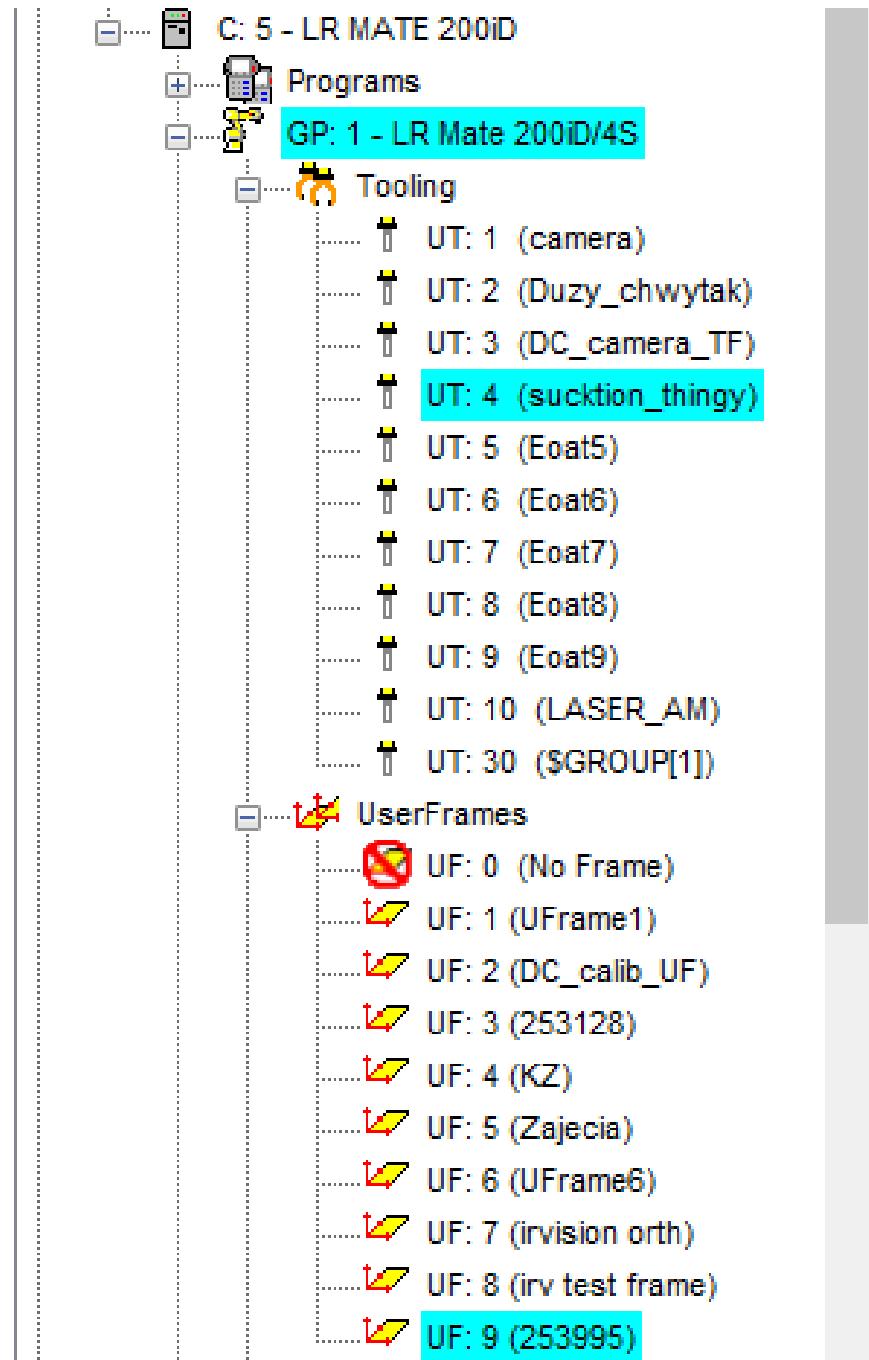
Natomiast UToolFrame jest układem współrzędnych efektora robota, czyli oznaczenie miejsca, gdzie znajduje się jego punkt pracy. Definicja UToolFrame umożliwia precyzyjną pracę manipulatora. W warunkach rzeczywistych układ współrzędnych narzędzia określa się za pomocą metod:

- **Three Point** - polega na precyzyjnym ustaleniu efektora w 3 różnych konfiguracjach
- **Six Point (XZ)** - metoda, w której również precyzyjnie ustawa się efektor w 3 różnych konfiguracjach, lecz dodatkowo istnieje możliwość zorientowania narzędzia poprzez podanie kierunków prowadzenia osi - w tym przypadku są to osie X oraz Z.
- **Six Point (XY)** - identycznie jak w przypadku Six Point (XZ), lecz różnica znajduje się w podawanych osiach. W tym przypadku są to X oraz Y.
- **Two Point + Z** - podobna metoda do Three Point, lecz dodatkowo zadawana jest os Z
- **Direct Entry** - bezpośrednie określenie UToolFrame, poprzez podanie konkretnych wartości liczbowych definiujących położenie i orientację narzędzia.

W FANUC Roboguide dzięki *RobotNeighbourhood* można skorzystać z już wcześniej definiowanych przez innych użytkowników układów współrzędnych UserFrame i ToolFrame (rys. 23). Jest to na tyle korzystne, że kiedy np. dla danego systemu wizyjnego jest skonfigurowany UserFrame, można z niego bezpiecznie korzystać ze świadomością tego, iż będzie poprawnie

2. Cela w środowisku FANUC Roboguide

działać. ToolFrame ułatwi dobranie odpowiedniego rozmiaru efektora w środowisku symulacyjnym. W pracy inżynierskiej takie praktyki zostały zastosowane dla obu manipulatorów.

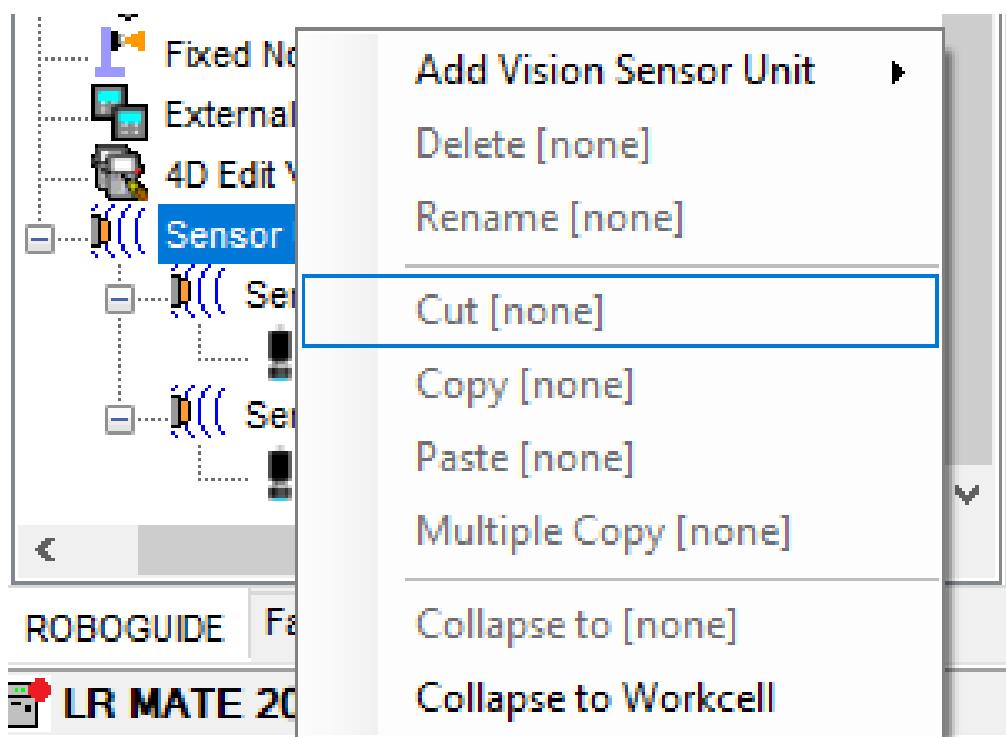


Rysunek 23: FANUC Roboguide - ToolFrame i UserFrame po zastosowaniu *RobotNeighbourhood*

2. Cela w środowisku FANUC Roboguide

2.5.4 Kamery w FANUC Roboguide

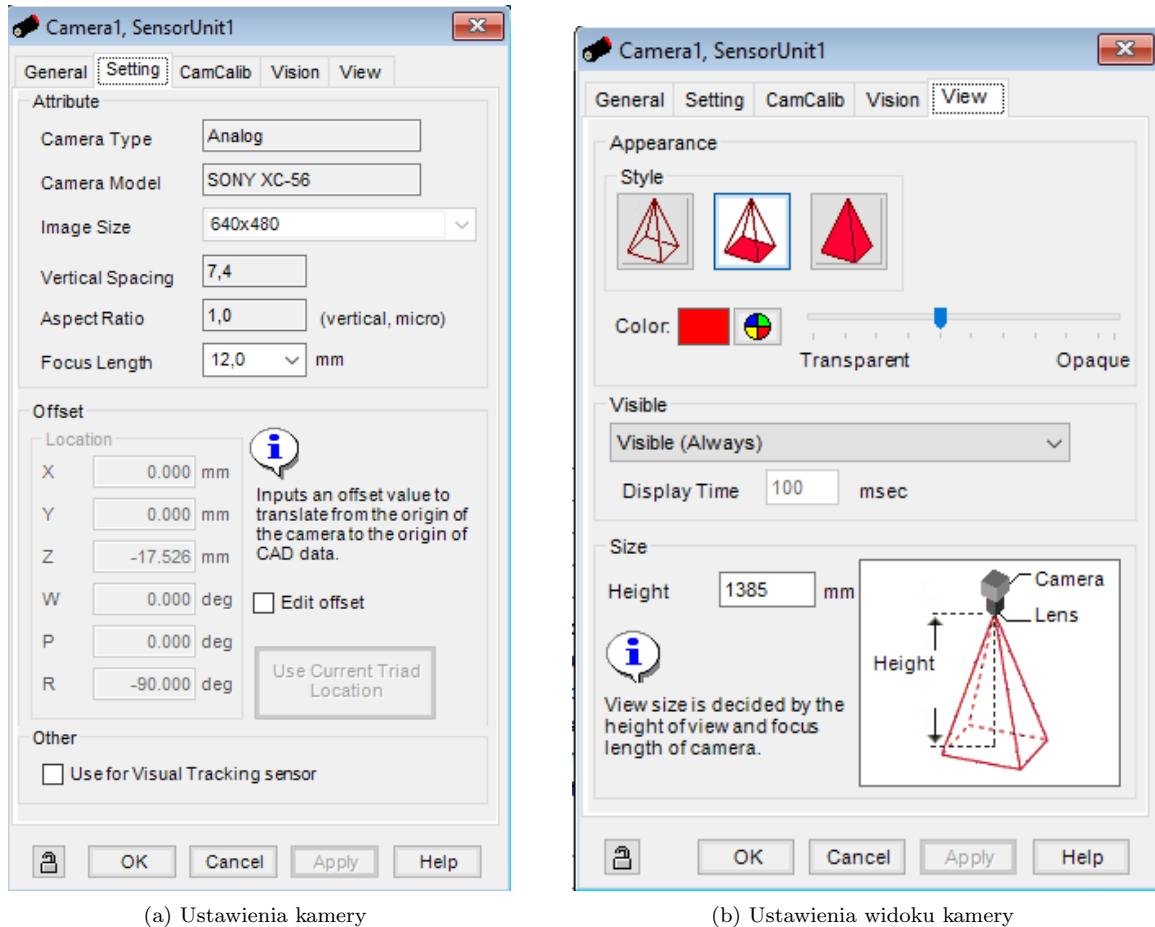
W FANUC Roboguide znajduje się zakładka SensorUnit. Należy nadmienić, iż w starszych wersjach programu opcja ta jest niedostępna. Wersja FANUC Roboguide, która znajduje się w laboratorium, nie posiada tej opcji, dlatego rekomenduje się zastosowanie nowszych wersji oprogramowania. SensorUnit charakteryzuje się możliwością dodania czujników związanych z wizją. W celu zainstalowania kamery należy kliknąć prawym przyciskiem myszy na zakładkę, następnie przyciskając *Add Vision Sensor Unit* (rys. 24) i wybrać z bogatej oferty kamerę - od monochromatycznych 2D po kolorowe 3D.



Rysunek 24: FANUC Roboguide - Dodanie kamery

Kamera, tak samo jak *Fixtures* czy *Parts* jest obiektem 3D. Można zmieniać jego rozmiar i położenie. Istnieją też opcje dotyczące ustawień kamery tj. zmiana ogniskowej, ustawienie odległości kamery czy dostosowanie rozdzielczości kamery (rys. 25a). Model w FANUC Roboguide pokazuje kolorową poświatę, oznaczającą obszar obejmowany przez kamerę - przydatne przy m.in. kalibrowaniu systemu wizyjnego (rys. 25b).

2. Cela w środowisku FANUC Roboguide



Rysunek 25: FANUC Roboguide - kamera

2.5.5 Projektowanie celi - efekt

Korzystając z zamieszczonych dotychczas informacji można stworzyć zrobotyzowaną celę w FANUC Roboguide. Kolejnym etapem po utworzeniu celi jest napisanie programu testowego, sprawdzającego wielkość przestrzeni roboczej manipulatora. W tym celu napisano program przemieszczający manipulator, na wysokości 2 cm, pomiędzy kolejnymi krawędziami przestrzeni roboczej. Po niewielkich korekcjach, można było zatwierdzić pozytywne wykonanie celi. Jej wygląd prezentuje się jak na rys. 26.

2. Cela w środowisku FANUC Roboguide



Rysunek 26: Efekt projektowania celi

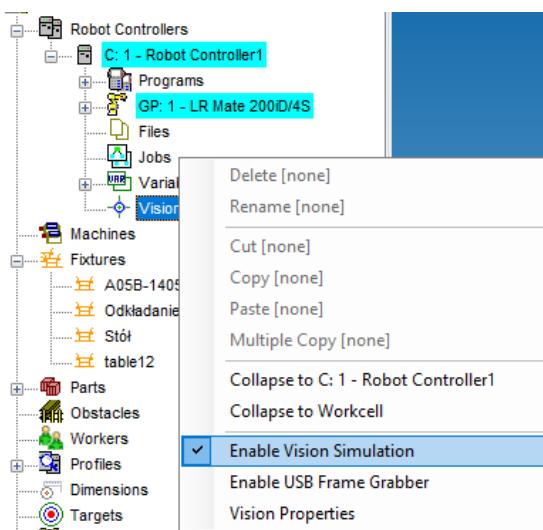
Rozdział 3

FANUC iRVision

Dedykowany system wizyjny dla marki FANUC pozwala nauczyć robota rozpoznawania elementów w przestrzeni obejmowanej przez kamerę. Taki dedykowany system wizyjny został zaimplementowany w wersjach FANUC Roboguide powyżej 9.1. Należy nadmienić, że dzięki metodzie *Robot Neighbourhood* (str. 12) można pobrać wszystkie konfiguracje iRVision bezpośrednio z robota. Wszystkie przedstawione poniżej funkcje można wykonać zarówno pod FANUC Roboguide jak i bezpośrednio na robocie, lecz z powodu braku dokładnych modeli CAD rzeczywistych obiektów i możliwości importu własnych ustawień iRVision do manipulatora, konfiguracji/kalibracji i nauczania nie przeprowadzono w symulatorze, a skorzystano z bezpośrednich możliwości oferowanych w ustawieniach kamery (z poziomu przeglądarki).

3.1 Konfiguracja kamery w FANUC Roboguide

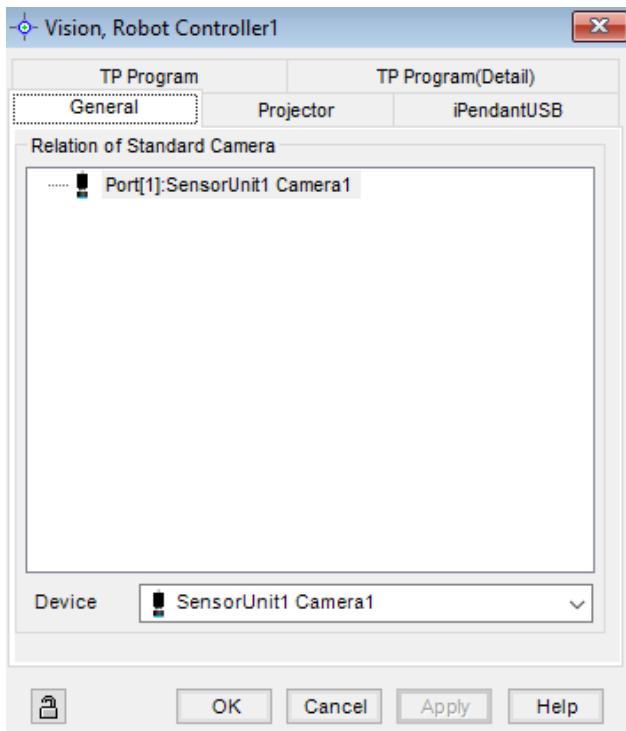
Kiedy w celi znajduje się chociaż jedna kamera, można rozpoczęć proces jej konfigurowania w FANUC Roboguide. Obligatoryjne jest włączenie systemu wizyjnego dla danego manipulatora, poprzez kliknięcie prawym przyciskiem na zakładkę *Vision*, a następnie *Enable Vision Simulation*. Po poprawnym włączeniu powinna pojawić się parafka obok tej opcji (rys. 27).



Rysunek 27: FANUC Roboguide - Włączenie systemu wizyjnego w manipulatorze

3. FANUC iRVision

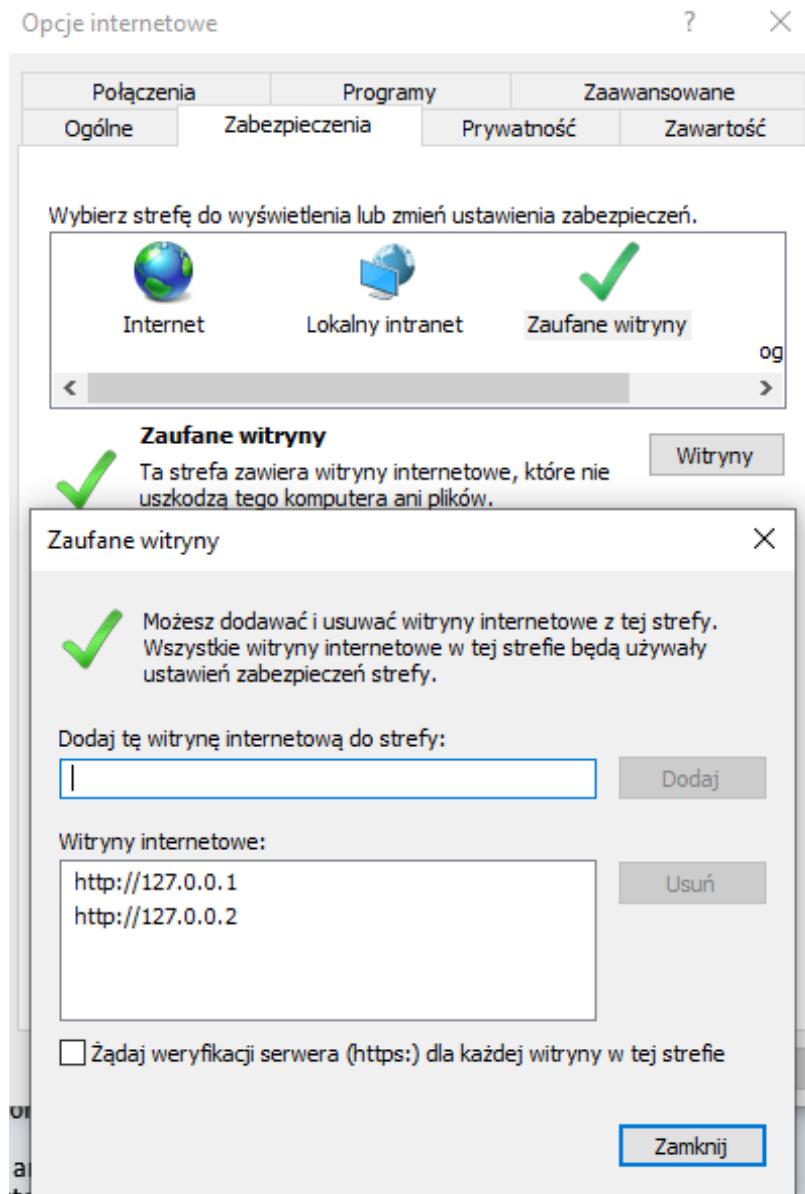
Jeszcze przed przejściem do konfiguracji iRVision należy we właściwościach wybrać port, do którego podłączona jest kamera (rys. 28). W tym celu również prawym przyciskiem myszy należy kliknąć na zakładkę *Vision*, a następnie *Vision Properties*. W opcjach ogólnych istnieje możliwość podłączenia kamery do portu. Po dokonaniu tych czynności można przejść do konfigurowania iRVision.



Rysunek 28: FANUC Roboguide - Dodanie kamery do portu

W celu wejścia do ustawień kamery można wykorzystać Teach Pendant lub przeglądarkę internetową. Na panelu należy wybrać opcje *MENU → IRVISION → VISION SETUP*. W FANUC Roboguide pozwala na to zakładka *Robot* znajdująca się w górnym pasku. Następnie należy kliknąć *Internet Explorer*. W pasku adresowym ukaże się adres IP robota. Korzystając jedynie z manipulatora zaimplementowanego w symulatorze należy zostawić podstawowy adres - 127.0.0.1. Natomiast istnieje możliwość połączenia się z rzeczywistym manipulatorem poprzez FANUC Roboguide. Wówczas wystarczy w pasku adresowym wpisać adres IP rzeczywistego robota. Następnie należy wybrać opcję *Vision Setup*. W niektórych wersjach manipulatorów mogą pojawić się problemy z otwarciem ustawień wizyjnych. Często wystarczy dodać adres IP robota do zaufanych witryn. W tym celu w opcjach internetowych przeglądarki Internet Explorer istnieje zakładka *Zabezpieczenia*, gdzie poprzez przycisk *Witryny* można dodać adres do zaufanych (rys. 29).

3. FANUC iRVision



Rysunek 29: Dodanie adresu IP robota do zaufanych witryn

Po wykonaniu tych czynności, w zależności od wersji robota, ukaże się system iRVision. Jeżeli dalej pojawią się problemy z uruchomieniem *Vision Setup*, mogą wynikać one z kompatybilności starszej wersji przeglądarki. Wtedy rozwiązańe znajduje się w opcji ustawień przeglądarki: *Ustawienia widoku zgodności*.

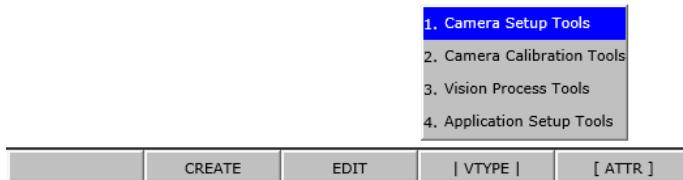
3.2 Konfiguracja kamery w iRVision

Rozpocząć konfigurację należy poprzez uruchomienie z poziomu przeglądarki *Internet Explorer*, a następnie wybranie opcji *Vision Setup* (rys. 30).



Rysunek 30: Wybór opcji Vision Setup

Po kliknięciu oprogramowanie przeniesie użytkownika do okna, w którym znajdują się wszystkie elementy związane z systemem wizyjnym: ustawienia, kalibracje i procesy. Domyslnie na początku uruchamia się okno związane z ustawieniem kamery. Jeżeli natomiast tak się nie stało, w opcji *VTYPE* należy wybrać *Camera Setup Tools* (rys. 31) i stworzyć nowe ustawienie za pomocą przycisku *Create*.



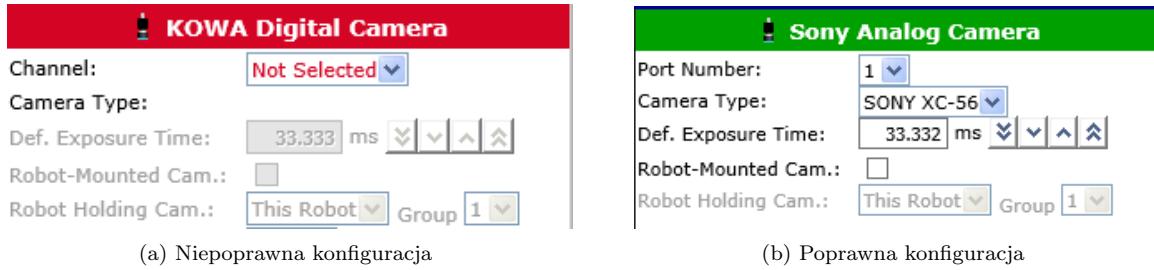
Rysunek 31: Wybór opcji ustawień kamery

Po wybraniu typu kamery, nadaniu nazwy i dodaniu ewentualnego komentarza można przejść do konfiguracji ustawień:

- **Port Number** - wybór portu, do którego przypisana jest kamera (str. 31)
- **Camera type** - wybór typu używanej kamery
- **Def. Exposure Time** - zmiana czasu naświetlania kamery. Im mniejsza wartość, tym ciemniejszy widok z kamery
- **Robot-Mounted Cam.** - opcja używana, gdy robot ma zainstalowaną kamerę na sobie.

3. FANUC iRVision

Jeżeli wszystko zostało skonfigurowane poprawnie oraz obraz widoczny jest zadowalający, okno ustawień zmieni swój kolor z czerwonego (rys. 32a) na zielony (rys. 32b) i można zakończyć definiowanie ustawień kamery.



Rysunek 32: Konfiguracja kamery w iRVision

3.3 Kalibracja kamery w iRVision

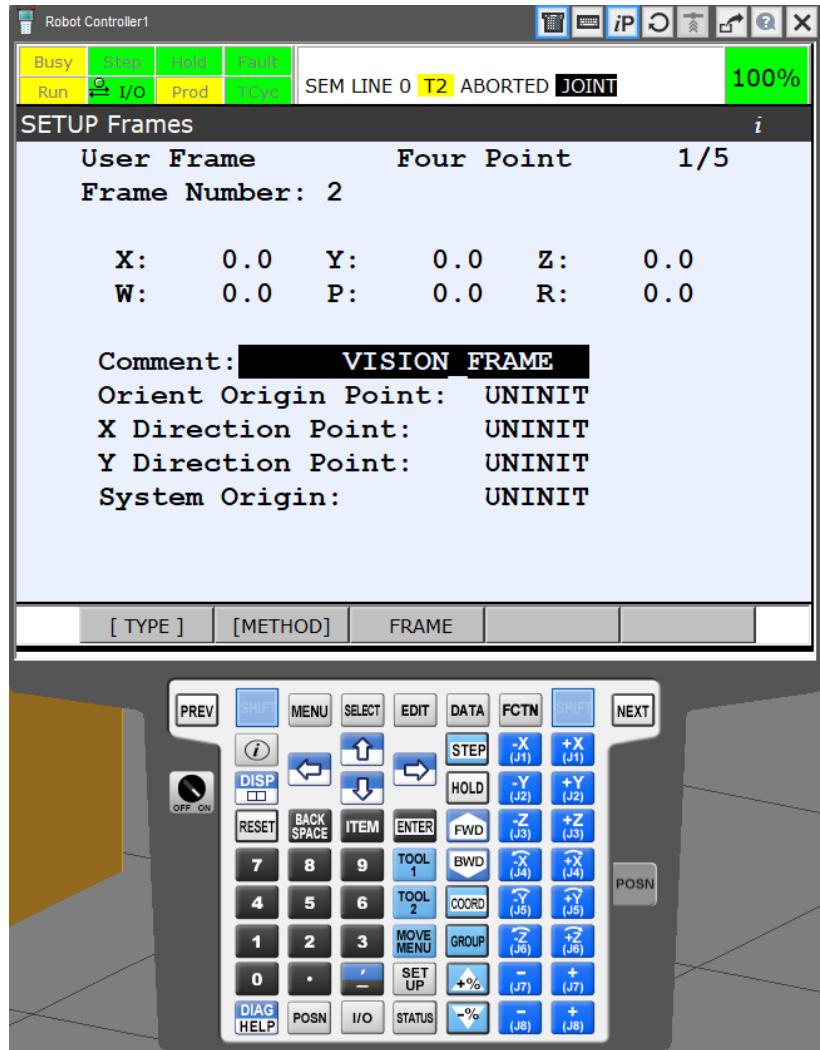
Kolejnym krokiem jest kalibracja kamery. W systemie iRVision jest to osobna zakładka, do której dostać się można również za pomocą *VTYPE* i wybrać *Camera Calibration Tools*. Tak samo jak w ustawieniach kamery należy nadać nazwę, opcjonalnie komentarz, ale najistotniejsza jest tutaj metoda kalibracji. Do wyboru jest kalibracja za pomocą *Siatki kalibracyjnej* lub *Siatki kalibracyjnej generowanej przez robota*. W FANUC Roboguide oraz Laboratorium Robotyki można znaleźć takie siatki, dlatego w pracy inżynierskiej kalibracja odbyła się za pomocą pierwszej metody. W symulatorze znajdują się one w zakładce *Fixtures* jako obiekty CAD. Przy ustawianiu kalibracji występują 3 istotne kwestie:

1. **Wszystkie punkty siatki kalibracyjnej muszą znajdować się w polu widzenia kamery.**
2. **Manipulator musi mieć możliwość dojazdu do każdego z punktów siatki.**
3. **Jeżeli to możliwe, siatka powinna zajmować jak największy obszar widzenia kamery.**

O powyższych zasadach należy pamiętać przy doborze rozmiarów planszy kalibracyjnej czy wysokości kamery nad stołem roboczym. Obraz kamery można podglądać w czasie rzeczywistym za pomocą opcji *Live*. W tym momencie, posiadając dobrze skonfigurowaną kamerę oraz odpowiednio położoną siatkę kalibracyjną, należy przerwać proces kalibracji w celu utworzenia wizyjnego układu współrzędnych UserFrame.

3. FANUC iRVision

Na panelu sterowania manipulatorem należy wybrać *MENU* → *SETUP* → *FRAMES* i dzięki opcji *[OTHER]* wejść do tworzenia własnego UserFrame'u. W systemach wizyjnych wykorzystuje się metodę czterech punktów (ang. Four Point).



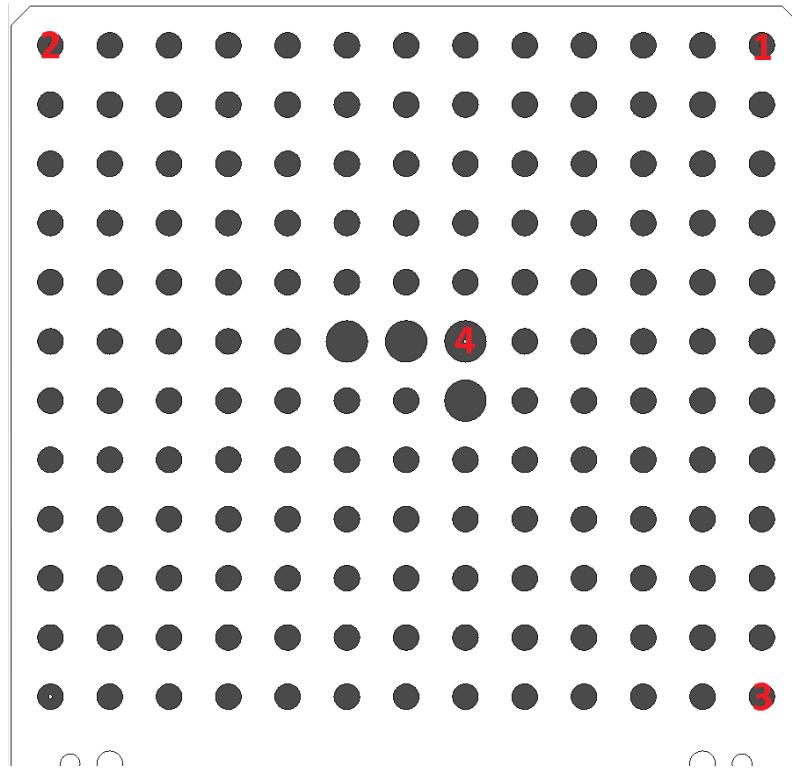
Rysunek 33: Definiowanie UserFrame'u dla systemu wizyjnego

Rysunek 33 pokazuje, że do wykreowania układu współrzędnych użytkownika za pomocą tej metody, potrzebne są 4 punkty:

1. **Orient Origin Point** - początek układu współrzędnych
2. **X Direction Point** - definiowanie kierunku osi X
3. **Y Direction Point** - definiowanie kierunku osi Y
4. **System Origin** - umiejscowienie UserFrame w przestrzeni.

3. FANUC iRVision

Do każdego z tych punktów należy podjechać efektorem manipulatora i zapisać koordynaty za pomocą metody *SHIFT + RECORD*. FANUC Roboguide umożliwia dojazd efektorem do konkretnego punktu za pomocą kombinacji *CTRL + SHIFT + LPM*, co jest znacznym ułatwieniem w tej kwestii. Na rys. 34 przedstawiony jest wygląd siatki kalibracyjnej i zaznaczonymi na niej punktami, do których trzeba podjechać. Trzy większe kropki pośrodku definiują kierunek prowadzenia osi X, a dwie - osi Y. Każda z takich siatek ma także istotny parametr, jakim jest rozstaw kropek, który będzie uwzględniony podczas kalibracji.



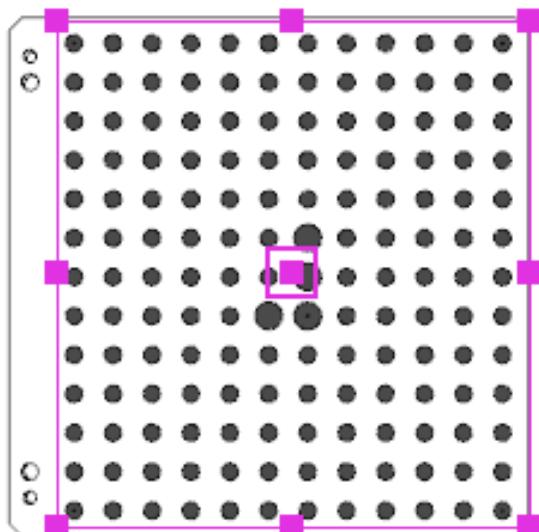
Rysunek 34: Siatka kalibracyjna

Posiadając zdefiniowany UserFrame, można powrócić do kalibrowania systemu wizyjnego. Każdy z podpunktów w *Grid Pattern Calibration Tool* należy dostosować do swojej kalibracji:

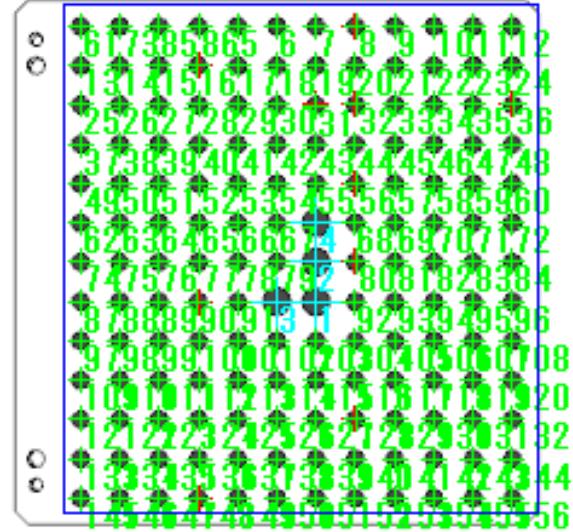
- **Robot to be offset** - opcja wykorzystywana, jeżeli pojawiają się dodatkowe grupy ruchów. Nominalnie robot posiada tylko grupę pierwszą
- **Application Frame** - tutaj należy ustawić UserFrame, który zdefiniowano dla systemu wizyjnego wcześniej
- **Camera** - wybór konfiguracji kamery

- **Exposure Time** - czas naświetlania, wpływa na jasność obrazu
- **Grid Spacing** - odległość pomiędzy kropkami na siatce kalibracyjnej
- **Number of Planes** - 1 - wybierane, gdy kamera i siatka kalibracyjna nie znajdują się na manipulatorze. 2 - powinno się wybrać, kiedy kamera lub siatka kalibracyjna są umieszczone na robocie
- **Robot-Held Cal. Grid** - czy manipulator trzyma siatkę kalibracyjną
- **Cal. Grid Frame** - tutaj należy ustawić wcześniej zdefiniowany UserFrame
- **Projection** - tryb perspektywiczny wybrany jako podstawowy. Natomiast jeżeli kamera posiada obiektyw telecentryczny powinno ustawić się tryb ortogonalny
- **Override Focal Dist.** - zmiana odległości ogniskowej.

Po ustawieniu tych wszystkich elementów zgodnie z własnymi potrzebami, należy nacisnąć przycisk *Find* i zaznaczyć obszar siatki kalibracyjnej, obejmując wszystkie punkty (rys. 35a). W kalibracji nie wolno zaznaczyć innych punktów niż czarne duże kropki odpowiadające za poprawne jej efekty (rys. 35b). Aby zakończyć z powodzeniem proces kalibracji systemu wizyjnego, należy kliknąć przycisk *Set*.



(a) Zaznaczenie siatki kalibracyjnej

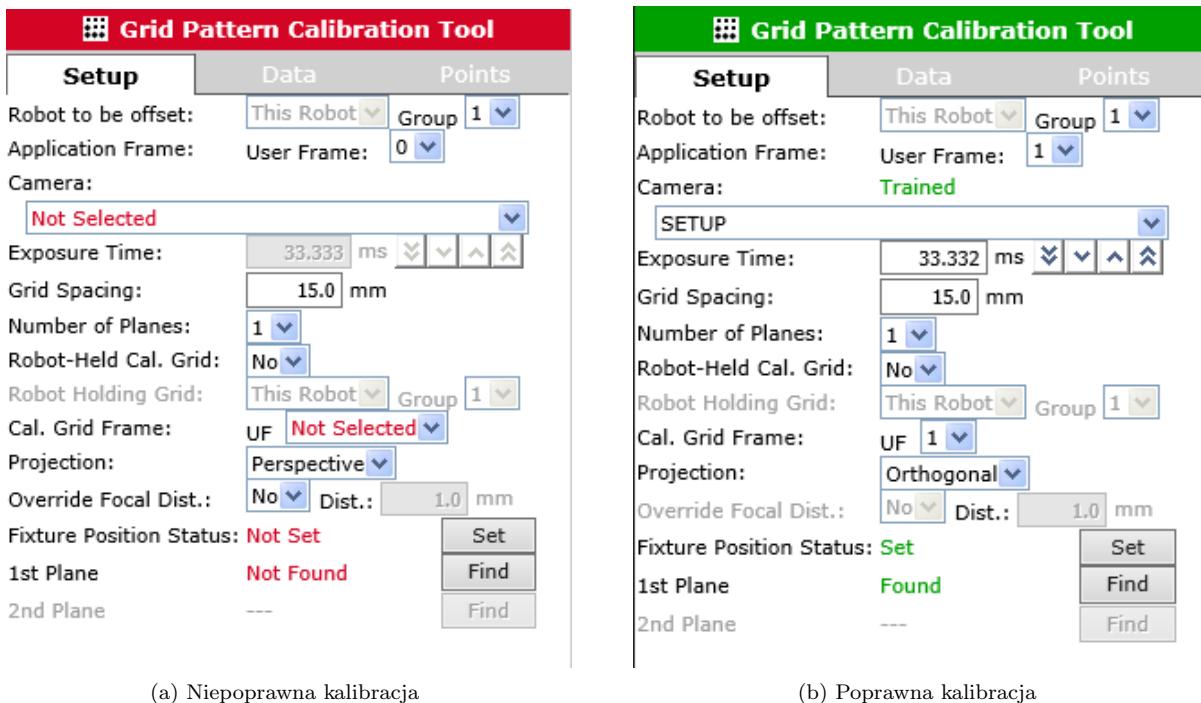


(b) Poprawnie znalezione punkty

Rysunek 35: Zaznaczanie punktów kalibracyjnych

3. FANUC iRVision

Po poprawnym wykonaniu wszystkich czynności kalibracyjnych kolor ustawień powinien zmienić się z czerwonego (rys. 36a) na zielony (rys. 36b). Można w zakładce *Data* znaleźć szczegółowe informacje dotyczące przeprowadzonej kalibracji, a w zakładce *Point* zaobserwować współrzędne punktów, które zostały przez system wizyjny znalezione. Istnieje również opcja usunięcia wadliwego punktu.

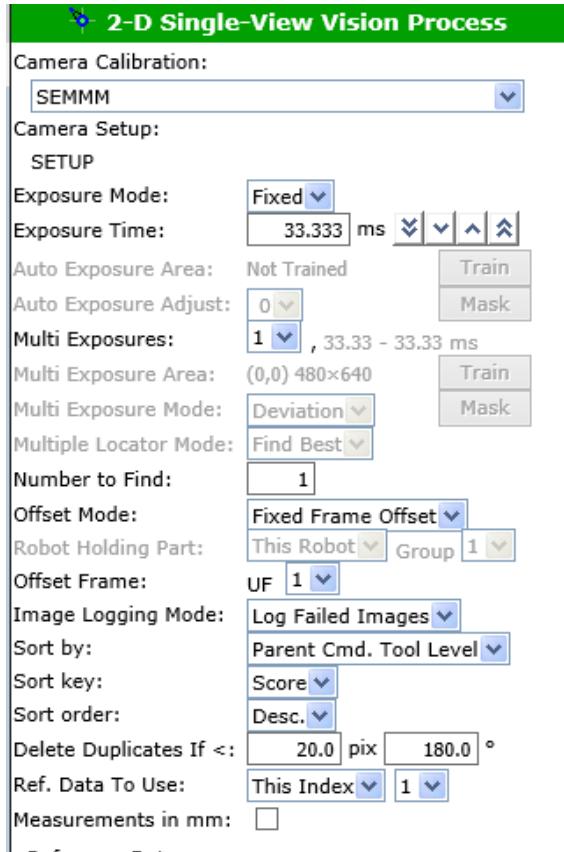


Rysunek 36: Kalibracja kamery w iRVision

3.4 Tworzenie procesu wizyjnego

Całą istotą istnienia systemu wizyjnego jest możliwość nauczania go rozpoznawania obiektów. W tym celu należy wybrać zakładkę *Vision Process Tools* i tam rozpoczęć nowy proces uczenia poprzez nadanie nazwy i dodanie opcjonalnego komentarza. Następnie na ekranie ukaże się okno procesowe. Aby rozpocząć nauczanie systemu, należy ustawić wszystkie opcje w zakładce głównej nauczania: kalibrację i ustawienia kamery, czas naświetlania, ilość obiektów do znalezienia oraz zdefiniowany UserFrame (rys. 37).

3. FANUC iRVision



Rysunek 37: Ustawienia główne uczonego modelu

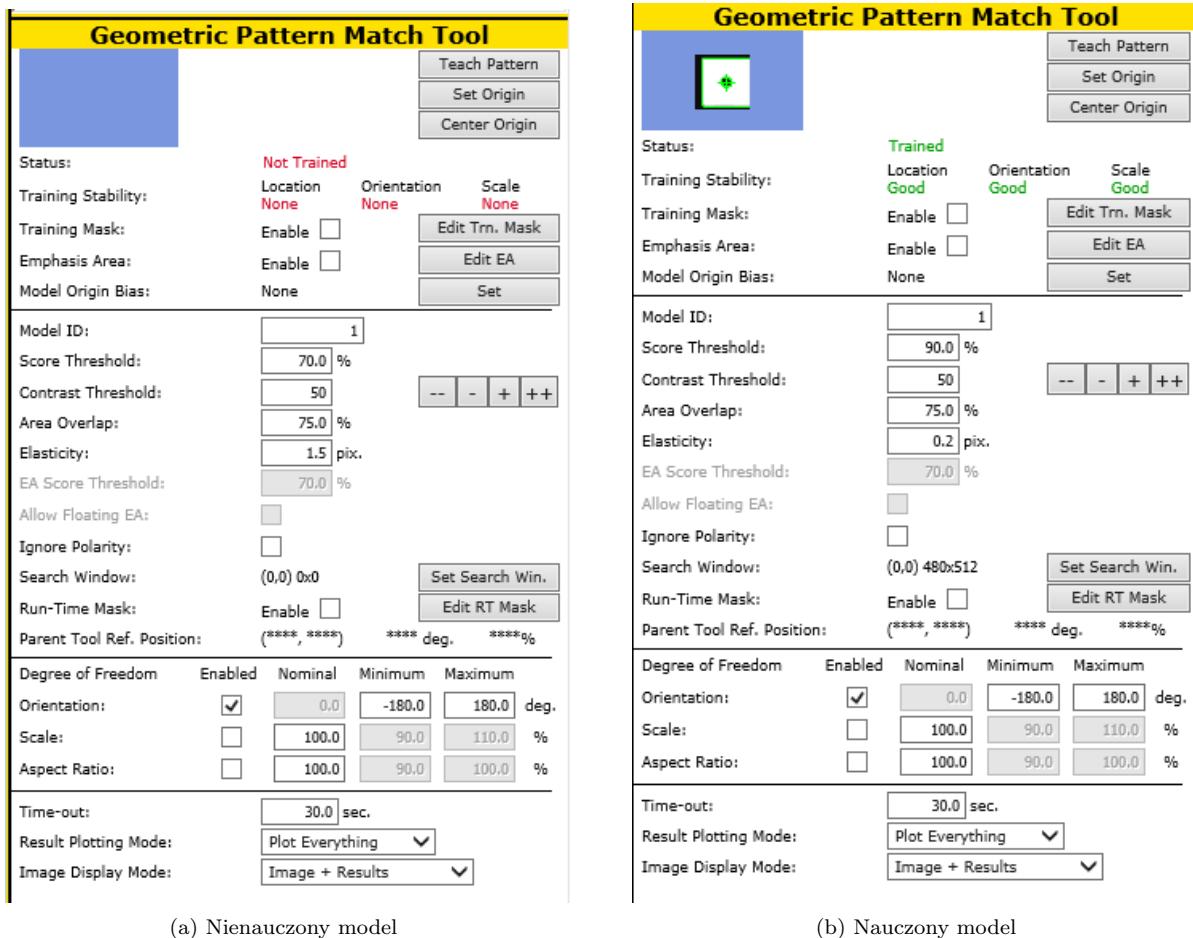
Następnie z drzewka procesowego wybrać *GPM (Geometric Pattern Match) Locator Tool*. Jest to narzędzie, które rozpoznaje obiekt na podstawie jego kształtu. Przed zmianą jakichkolwiek parametrów definiujących dokładność znalezionej obiektu, należy najpierw wskazać jaki kształt powinien być wykrywany. Odbywa się to za pomocą przycisku *Teach Pattern*. Podobnie jak zaznaczanie siatki kalibracyjnej, należy zaznaczyć obiekt. Będzie to podstawa do dalszych edycji uczenia systemu wizyjnego. Bardzo często jest tak, że po nauczeniu wykrywanych jest wiele obiektów na scenie, dlatego iRVision zapewnia użytkownikowi możliwość zmiany parametrów tak, aby tylko ten konkretny, przy wielu podobnych, był wykrywany. Te współczynniki to:

- **Training mask** - umożliwia użytkownikom ręczne usuwanie niechcianych krawędzi trenowanego modelu.
- **Emphasis Area** - używa się, gdy nie można prawidłowo określić pozycji przedmiotu na scenie.

- **Model Origin Bas.** - ta funkcja koryguje znalezione wyniki, dzięki czemu narzędzie generuje te same dane o znalezionej pozycji, co inne narzędzie GPM Locator Tool, które zostało już nauczone.
- **Model ID** - numer identyfikacyjny modelu.
- **Score threshold** - określa próg punktacji dla wzorca, który ma zostać znaleziony. Wzór na obrazie nie został znaleziony, jeśli jego wynik jest niższy niż określony próg. Wartość domyślna to 70 punktów.
- **Contrast Threshold** - określa próg wymaganego kontrastu (różnicy między jasnymi i ciemnymi) obrazu jako cechę charakterystyczną obiektu.
- **Area Overlap** - jeśli wzorce znalezione na obrazie nakładają się na siebie w stopniu większym niż określony, narzędzie GPM Locator Tool pozostawia tylko wzorzec posiadający najwyższy wynik i usuwa pozostałe.
- **Elasticity** - określa górną granicę rozszerzenia nauczonego obiektu w stosunku do wzorca modelu. Dopuszczalna elastyczność musi być określona w pikselach. Wartość domyślna to 1,5 piksela.
- **Ignore Polarity** - można określić czy przeprowadzać wykrywanie obiektu ignorując polaryzację ciemny/jasny wytrenowanego wzorca modelu.
- **Search Window** - określa obszar wykrywania obiektów przez system. Domyślnie przyjmuje wartość pełnej rozdzielczości obrazu.
- **Run Time Mask** - określa obszar okna wyszukiwania, który system wizyjny ma ignorować.

Dodatkowo można jeszcze ustalić czy system wizyjny ma brać pod uwagę lub ignorować: orientację, skalę i aspekt modelu. Podobnie jak z ustawieniem kamery czy jej kalibracji, jeżeli wszystko zostało poprawnie ustalone opcje zmienią kolor z czerwonego (rys. 38a) na zielony (rys. 38b).

3. FANUC iRVision



Rysunek 38: Uczenie rozpoznawania obiektu w iRVision

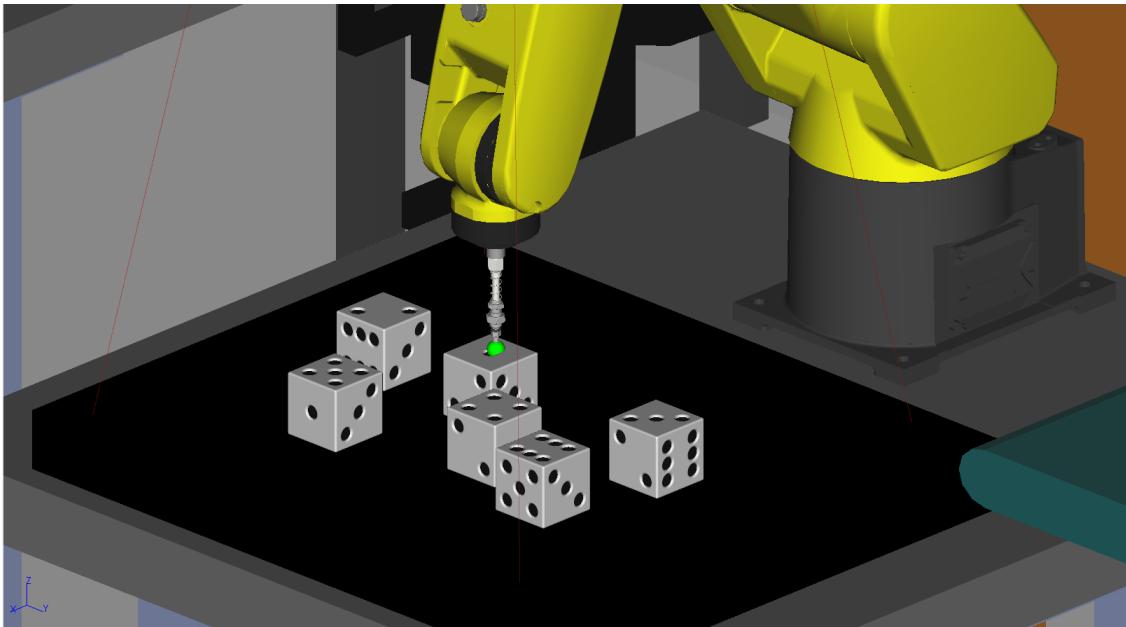
Aby zakończyć proces nauczania modelu, należy jeszcze cofnąć się w drzewku procesu do głównego okna, znaleźć obiekt za pomocą metody *SNAP + FIND* i ustawić pozycję referencyjną przyciskiem *Set* w zakładce *Reference Data* (rys. 39).



Rysunek 39: Ustawienie pozycji referencyjnej obiektu w iRVision

3.5 Ustawienie pozycji referencyjnej obiektu

Poza ustawieniem pozycji referencyjnej w iRVision, którą tak naprawdę system sam sobie wygenerował, należy zapisać tę pozycję w rejestrze pozycyjnym, aby później podczas programowania poszczególnych ruchów manipulatora mieć na uwadze punkt odniesienia (rys. 40).



Rysunek 40: Dojazd manipulatorem do pozycji referencyjnej

Najlepszym sposobem na zapamiętanie tej pozycji jest wykorzystanie *Position Registers* - rejestrów, w których zapisane są dokładne koordynaty. Aby to wykonać, należy dojechać do pozycji, a następnie wejść za pomocą iPendanta w *DATA* → *[OTHER]* → *POSITION REG*. Za pomocą metody *SHIFT + RECORD* wybrać jeden z niezajętych rejestrów i zapisać do niego tak wybraną pozycję robota. Bardzo istotną czynnością jest dodanie komentarza np. *REF_POS_1*, aby później ograniczyć problemy z implementacją kodu.

Rozdział 4

Program testowy

Do przetestowania zbudowanego całego systemu w FANUC Roboguide wymyślono program (rys. 41a i 41b), w którym zadaniem manipulatora było wykonanie paletyzacji na kostkach sześciennych do gry. IRVision rozpoznawał ilość oczek na kostce, a FANUC LR Mate 200iC ustawiał wieżę w wyznaczonym miejscu rozpoczynając od ściany z największą liczbą oczek. Następnie ten sam program z małymi korektami został przesłany do rzeczywistego manipulatora (rys. 42a i 42b) znajdującego się w Laboratorium Robotyki.

4.1 Program dla LR Mate 200iC w FANUC Roboguide

```
1 UFRAME_NUM=9
2 UTOOL_NUM=1
3 J P[1] 100% CNT0
4 LBL [10]
5 VISION RUN_FIND 'WK_253995_6'
6 VISION GET_OFFSET 'WK_253995_6' VR[4] JMP LBL[10]
7 J PR[66:6] 100% FINE Offset ,PR[3:wys_kostki] VOFFSET ,VR[4]
8 J PR[66:6] 100% FINE VOFFSET ,VR[4]
9 CALL PICK_6
10 J PR[66:6] 100% FINE Offset ,PR[39:PICK_DICE] VOFFSET ,VR[4]
11 J PR[69:ODL_6] 100% FINE Offset ,PR[3:wys_kostki]
12 J PR[69:ODL_6] 100% FINE
13 CALL DROP_6
14 J PR[69:ODL_6] 100% FINE Offset ,PR[3:wys_kostki]
15
16
17
18
19 LBL [11]
20 VISION RUN_FIND 'WK_253995_5'
21 VISION GET_OFFSET 'WK_253995_5' VR[5] JMP LBL[11]
22 J PR[65:5] 100% FINE Offset ,PR[3:wys_kostki] VOFFSET ,VR[5]
23 J PR[65:5] 100% FINE VOFFSET ,VR[5]
24 CALL PICK5
25 J PR[65:5] 100% FINE Offset ,PR[39:PICK_DICE] VOFFSET ,VR[5]
26 J P[2] 100% FINE Offset ,PR[39:PICK_DICE]
27 J PR[69:ODL_6] 100% FINE Offset ,PR[3:wys_kostki]
28 CALL DROPS
29 J P[2] 100% FINE Offset ,PR[39:PICK_DICE]
```

4. Program testowy

```
1 LBL [12]
2 VISION RUN_FIND 'WK_253995_4'
3 VISION GET_OFFSET 'WK_253995_4' VR[6] JMP LBL[12]
4 J PR[64:4] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[6]
5 J PR[64:4] 100% FINE VOFFSET,VR[6]
6 CALL PICK4
7 J PR[64:4] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[6]
8 J PR[70:ORIG_ODK] 100% FINE Offset,PR[39:PICK_DICE]
9 J PR[70:ORIG_ODK] 100% FINE Offset,PR[3:wys_kostki]
10 CALL DROP4
11 J PR[71:ODL_3] 100% FINE Offset,PR[39:PICK_DICE]
12
13
14
15
16
17 LBL [13]
18 VISION RUN_FIND 'WK_253995_3'
19 VISION GET_OFFSET 'WK_253995_3' VR[7] JMP LBL[13]
20 J PR[63:3] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[7]
21 J PR[63:3] 100% FINE VOFFSET,VR[7]
22 CALL PICK3
23 J PR[63:3] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[7]
24 J PR[71:ODL_3] 100% FINE Offset,PR[39:PICK_DICE]
25 J PR[71:ODL_3] 100% FINE Offset,PR[3:wys_kostki]
26 CALL DROP3
27 J PR[72:ODL_2] 100% FINE Offset,PR[39:PICK_DICE]
28
29
30
31
32
33 LBL [14]
34 VISION RUN_FIND 'WK_253995_2'
35 VISION GET_OFFSET 'WK_253995_2' VR[8] JMP LBL[14]
36 J PR[62:2] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[8]
37 J PR[62:2] 100% FINE VOFFSET,VR[8]
38 CALL PICK2
39 J PR[62:2] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[8]
40 J PR[72:ODL_2] 100% FINE Offset,PR[39:PICK_DICE]
41 J PR[72:ODL_2] 100% FINE Offset,PR[3:wys_kostki]
42 CALL DROP2
43 J PR[73:ODL1] 100% FINE Offset,PR[39:PICK_DICE]
44
45
46
47
48
49 LBL [15]
50 VISION RUN_FIND 'WK_253995_1'
51 VISION GET_OFFSET 'WK_253995_1' VR[9] JMP LBL[15]
52 J PR[61:1] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[9]
53 J PR[61:1] 100% FINE VOFFSET,VR[9]
```

4. Program testowy

```
1 CALL PICK1
2 J PR[61:1] 100% FINE Offset ,PR[39:PICK_DICE] VOFFSET ,VR[9]
3 J PR[73:ODL1] 100% FINE Offset ,PR[39:PICK_DICE]
4 J PR[73:ODL1] 100% FINE Offset ,PR[3:wys_kostki]
5 CALL DROP1
6 J PR[73:ODL1] 100% FINE Offset ,PR[39:PICK_DICE]
7 J P[1] 100% CNT20
8 J P[3] 100% CNT20
```

Listing 1: Program dla manipulatora LR Mate 200iC w FANUC Roboguide

4.2 Program dla rzeczywistego LR Mate 200iC

```
1 UFRAME_NUM=9
2 UTOOL_NUM=1
3 J P[1] 100% CNT0
4
5 LBL[12]
6 VISION RUN_FIND 'WK_253995_4'
7 VISION GET_OFFSET 'WK_253995_4' VR[6] JMP LBL[12]
8 J PR[64:4] 100% FINE Offset ,PR[3:wys_kostki] VOFFSET ,VR[6]
9 J PR[64:4] 100% FINE VOFFSET ,VR[6]
10 CALL OPEN
11 J PR[64:4] 100% FINE Offset ,PR[39:PICK_DICE] VOFFSET ,VR[6]
12 J PR[70:ORIG_ODK] 100% FINE Offset ,PR[39:PICK_DICE]
13 J PR[70:ORIG_ODK] 100% FINE Offset ,PR[3:wys_kostki]
14 CALL CLOSE
15 J PR[71:ODL_3] 100% FINE Offset ,PR[39:PICK_DICE]
16
17
18
19
20
21 LBL[13]
22 VISION RUN_FIND 'WK_253995_3'
23 VISION GET_OFFSET 'WK_253995_3' VR[7] JMP LBL[13]
24 J PR[63:3] 100% FINE Offset ,PR[3:wys_kostki] VOFFSET ,VR[7]
25 J PR[63:3] 100% FINE VOFFSET ,VR[7]
26 CALL OPEN
27 J PR[63:3] 100% FINE Offset ,PR[39:PICK_DICE] VOFFSET ,VR[7]
28 J PR[71:ODL_3] 100% FINE Offset ,PR[39:PICK_DICE]
29 J PR[71:ODL_3] 100% FINE Offset ,PR[3:wys_kostki]
30 CALL CLOSE
31 J PR[72:ODL_2] 100% FINE Offset ,PR[39:PICK_DICE]
32
33
34
35
36
37 LBL[14]
38 VISION RUN_FIND 'WK_253995_2'
```

4. Program testowy

```
1 VISION GET_OFFSET 'WK_253995_2' VR[8] JMP LBL[14]
2 J PR[62:2] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[8]
3 J PR[62:2] 100% FINE VOFFSET,VR[8]
4 CALL OPEN
5 J PR[62:2] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[8]
6 J PR[72:ODL_2] 100% FINE Offset,PR[39:PICK_DICE]
7 J PR[72:ODL_2] 100% FINE Offset,PR[3:wys_kostki]
8 CALL CLOSE
9 J PR[73:ODL1] 100% FINE Offset,PR[39:PICK_DICE]
10
11
12
13
14
15 LBL[15]
16 VISION RUN_FIND 'WK_253995_1',
17 VISION GET_OFFSET 'WK_253995_1' VR[9] JMP LBL[15]
18 J PR[61:1] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[9]
19 J PR[61:1] 100% FINE VOFFSET,VR[9]
20 CALL OPEN
21 J PR[61:1] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[9]
22 J PR[73:ODL1] 100% FINE Offset,PR[39:PICK_DICE]
23 J PR[73:ODL1] 100% FINE Offset,PR[3:wys_kostki]
24 CALL CLOSE
25 J PR[73:ODL1] 100% FINE Offset,PR[39:PICK_DICE]
26 J P[1] 100% CNT20
27 J P[3] 100% CNT20
```

Listing 2: Program dla rzeczywistego manipulatora LR Mate 200iC

4.3 Program dla symulatora i rzeczywistego manipulatora - różnice

Cała trajektoria wykonywana przez manipulator oraz idea działania programu jest identyczna w obu przypadkach. Istnieją jednak dwie różnice, które musiały zostać uwzględnione przy przenoszeniu programu z symulatora na rzeczywisty manipulator:

1. W Laboratorium Robotyki jest wyposażone w **4 kostki do gry** odpowiednich wymiarów, dlatego program rozpoczynając przeszukiwanie platformy roboczej od kostki z największą ilością oczek, zaczyna od kostki z czterema, a nie jak w przypadku symulatora, z sześcioma oczkami.
2. Polecenia **CALL PICK** oraz **CALL DROP** zostały zastąpione makrami **CALL OPEN** oraz **CALL CLOSE**. Dwa pierwsze zostały stworzone w symulatorze i one tylko imitują przenoszenie elementów przez manipulator. Natomiast makra **OPEN** oraz **CLOSE** odpowiednio włączają i wyłączają chwytkę przyssawkową manipulatora

LR Mate 200iC w pracowni.

4.4 Opis szczegółowy programu

4.4.1 Inicjalizacja programu

Program składa się z podobnych do siebie grup wywoływań ruchów. Każda z nich odpowiada za wykonanie trajektorii dla konkretnej kostki z liczbą oczek na górnej ścianie. Rozpoczyna się inicjalizacją układów współrzędnych oraz pozycji *HOME*.

```
1 UFRAME_NUM=9
2 UTOOL_NUM=1
3 J P[1] 100% CNT0
```

Listing 3: Inicjalizacja programu

- **UFRAME_NUM** - numer układu współrzędnych użytkownika. Skonfigurowany specjalnie na potrzeby systemu wizyjnego dla manipulatora LR Mate 200iC. W tym przypadku parametr przyjmuje wartość 9.
- **UTOOL_NUM** - numer układu współrzędnych narzędzia, który jest efektorem manipulatora. Dla LR Mate 200iC jest to chwytak przyssawkowy przypisany pod wartość 1.
- **J P[1] 100% CNT0** - pozycja home w programie. Ustawiona tak, aby nie zasłaniać elementów umieszczonych na stole roboczym.

4.4.2 Główna część programu

Główna część programu odpowiada za uruchomienie zdefiniowanych procesów wizyjnych w celu odnalezieniu odpowiednich kostek oraz realizację trajektorii manipulatora związanej z ich paletyzacją.

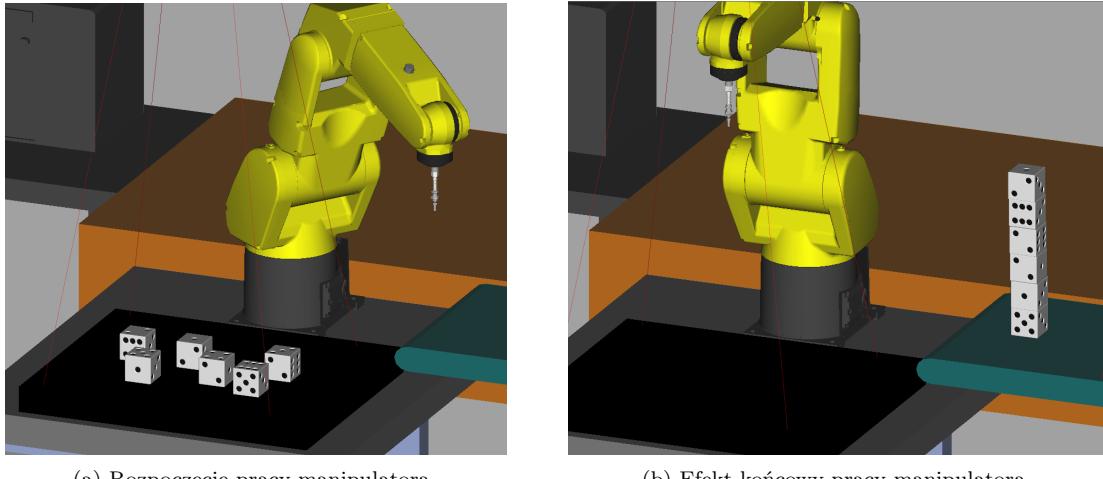
```
1 LBL[11]
2 VISION RUN_FIND 'WK_253995_5'
3 VISION GET_OFFSET 'WK_253995_5' VR[5] JMP LBL[11]
4 J PR[65:5] 100% FINE Offset, PR[3:wys_kostki] VOFFSET, VR[5]
5 J PR[65:5] 100% FINE VOFFSET, VR[5]
6 CALL PICK5
7 J PR[65:5] 100% FINE Offset, PR[39:PICK_DICE] VOFFSET, VR[5]
8 J P[2] 100% FINE Offset, PR[39:PICK_DICE]
9 J PR[69:ODL_6] 100% FINE Offset, PR[3:wys_kostki]
10 CALL DROP5
11 J P[2] 100% FINE Offset, PR[39:PICK_DICE]
```

Listing 4: Przykładowy segment głównej części programu

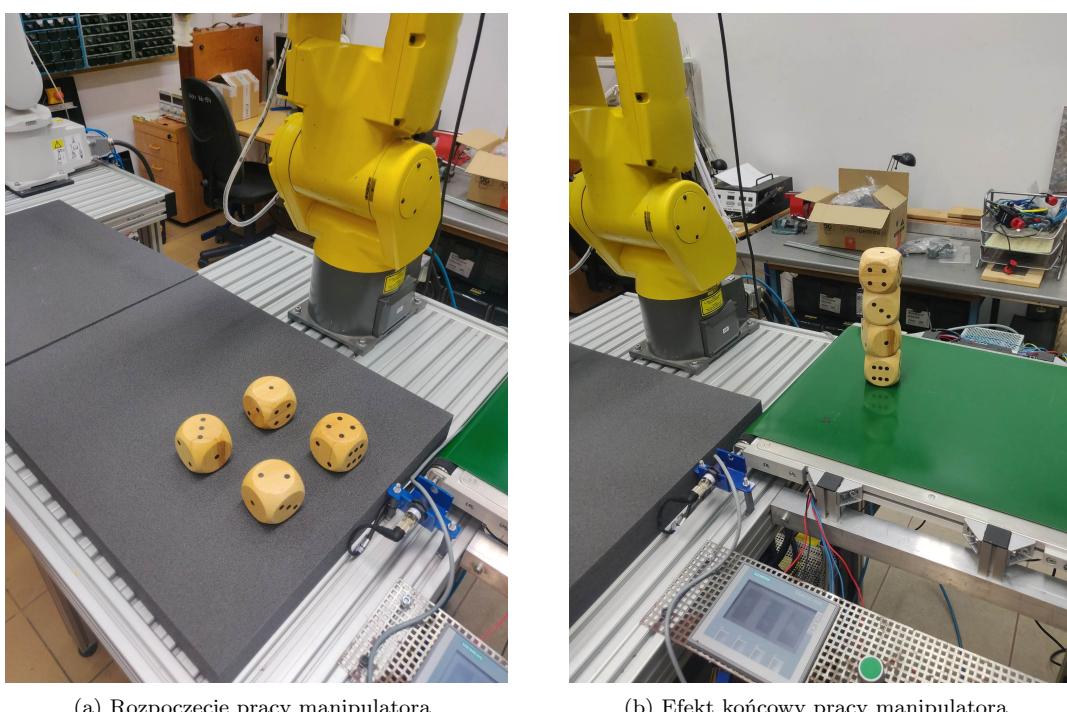
4. Program testowy

- **LBL[11]** - etykieta, do której program będzie powracał, jeżeli obiekt nie zostanie znaleziony przez system iRVision.
- **VISION RUN_FIND 'WK_253995_5'** - uruchamiany jest zdefiniowany wcześniej proces wizyjny o danej nazwie. W tym przypadku nazwa **WK_253995_5** oznacza proces, którego na celu jest odnalezienie kostki z pięcioma oczkami na górnej ścianie
- **VISION GET_OFFSET 'WK_253995_5' VR[5] JMP LBL[11]** - Komenda zwraca przesunięcie obiektu względem środka układu współrzędnych użytkownika i zapisuje je do rejestru **VR[5]**. W przypadku nieodnalezienia obiektu przez system, program przechodzi do etykiety **LBL[11]**.
- **J PR[65:5] 100% FINE Offset, PR[3:wys_kostki] VOFFSET,VR[5]** - manipulator przejeżdża do pozycji referencyjnej **PR[65:5]** z przesunięciem zapisanym w **VR[5]** oraz na wysokość **PR[3:wys_kostki]** nad kostką.
- **J PR[65:5] 100% FINE VOFFSET, VR[5]** - robot ustawia swoją przyssawkę dokładnie nad elementem uwzględniając przesunięcie **VR[5]** względem pozycji referencyjnej **PR[65:5]**.
- **CALL PICK5** - rozpoczęcie symulacji podniesienia kostki z pięcioma oczkami w FANUC Roboguide. W rzeczywistym symulatorze jest tutaj **CALL OPEN** - efektor przyssawkowy zaczyna działać
- **J PR[65:5] 100% FINE Offset, PR[39:PICK_DICE] VOFFSET,VR[5]** - manipulator jedzie z podniesioną kostką na bezpieczną wysokość **PR[39:PICK_DICE]** nad stół roboczy pionowo w góre.
- **J P[2] 100% FINE Offset, PR[39:PICK_DICE]** - przejazd z kostką do miejsca odkładania **P[2]** na wysokość **PR[39:PICK_DICE]**
- **J PR[69:ODL_6] 100% FINE Offset, PR[3:wys_kostki]** - dojazd do pozycji odkładania kostki **PR[69:ODL_6]** o wysokości **PR[3:wys_kostki]**
- **CALL DROP5** - rozpoczęcie symulacji odkładania kostki z pięcioma oczkami w FANUC Roboguide. W rzeczywistym symulatorze jest tutaj **CALL CLOSE** - efektor przyssawkowy kończy działanie
- **J P[2] 100% FINE Offset, PR[39:PICK_DICE]** - przejazd bez kostki nad miejsce odkładania **P[2]** na wysokość **PR[39:PICK_DICE]**

4.5 Efekty pracy programu



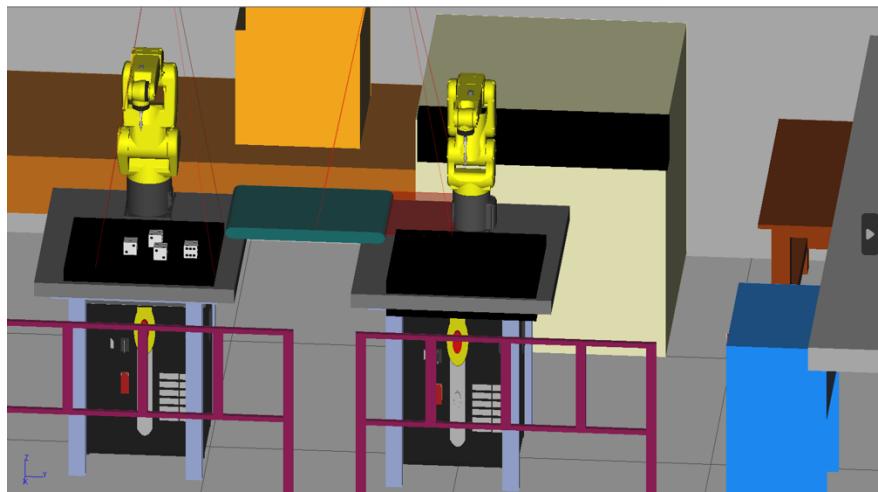
Rysunek 41: FANUC Roboguide - program testowy



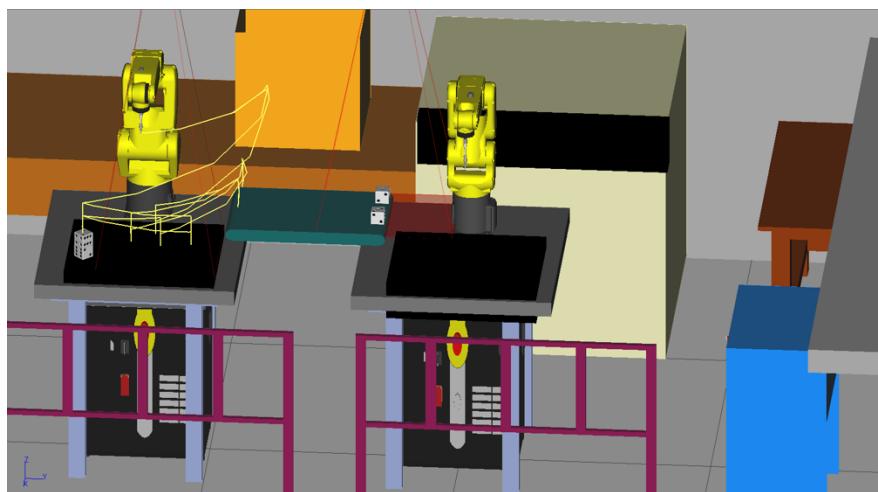
Rysunek 42: Rzeczywisty manipulator - program testowy

4.6 Rozszerzenie programu testowego

Program podstawowy rozszerzono o zintegrowanie pracy dwóch manipulatorów (LR Mate 200iC i LR Mate 200iD). Zadaniem LR Mate 200iC było rozpoznanie za pomocą systemu iRVision i przełożenie na taśmociąg łączący oba manipulatory kostek z jednym oraz trzema oczkami (rys. 43 i 46), a następnie paletyzacja pozostałych kostek. Po zakończeniu pracy przez LR Mate 200iC i przemieszczeniu kostek za pomocą taśmociągu (rys. 44 i 47), robot LR Mate 200iD wyszukiwał, a następnie zabierał wyznaczone elementy z taśmociągu i paletyzował je na środku swojej platformy roboczej (rys. 45 i 48). Program zrealizowany w symulatorze wysłano na rzeczywiste manipulatory, które odwzorowały poprawnie odwzorowały zadanie.

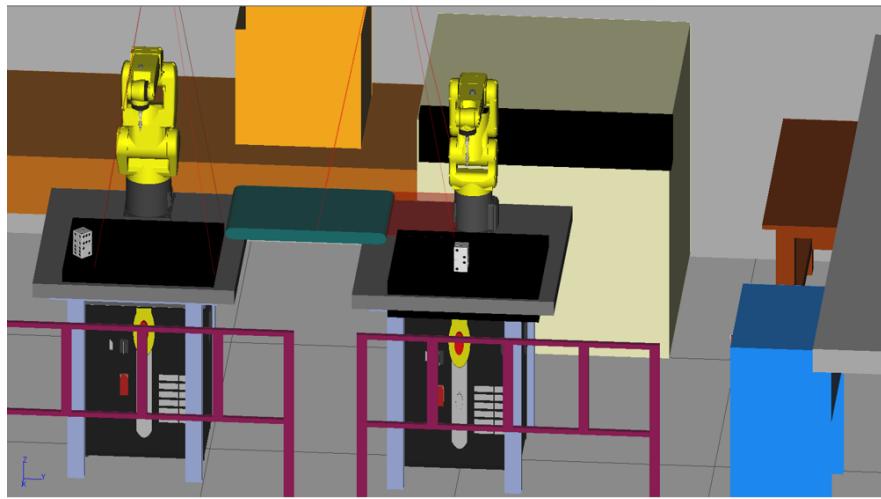


Rysunek 43: Początek – integracja manipulatorów, FANUC Roboguide

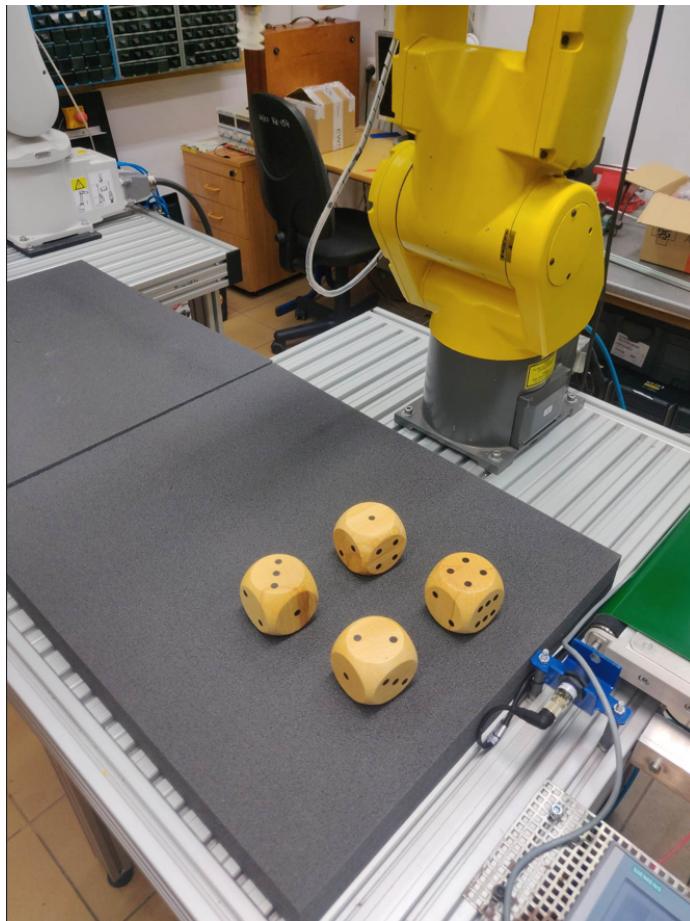


Rysunek 44: Punkt przejściowy – integracja manipulatorów, FANUC Roboguide

4. Program testowy

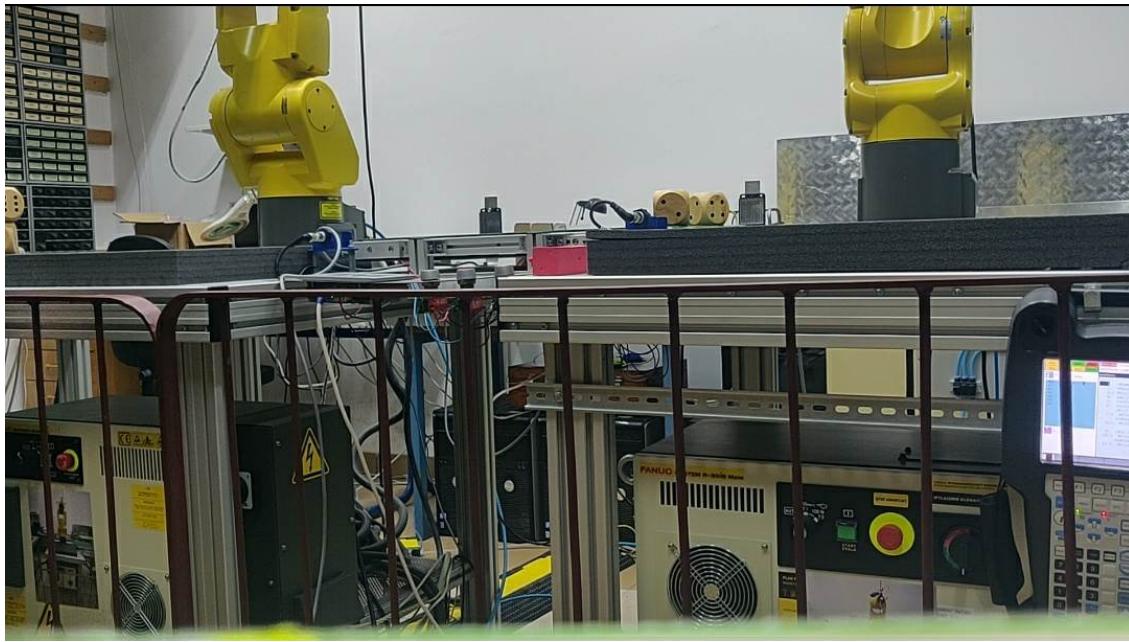


Rysunek 45: Punkt końcowy – integracja manipulatorów, FANUC Roboguide

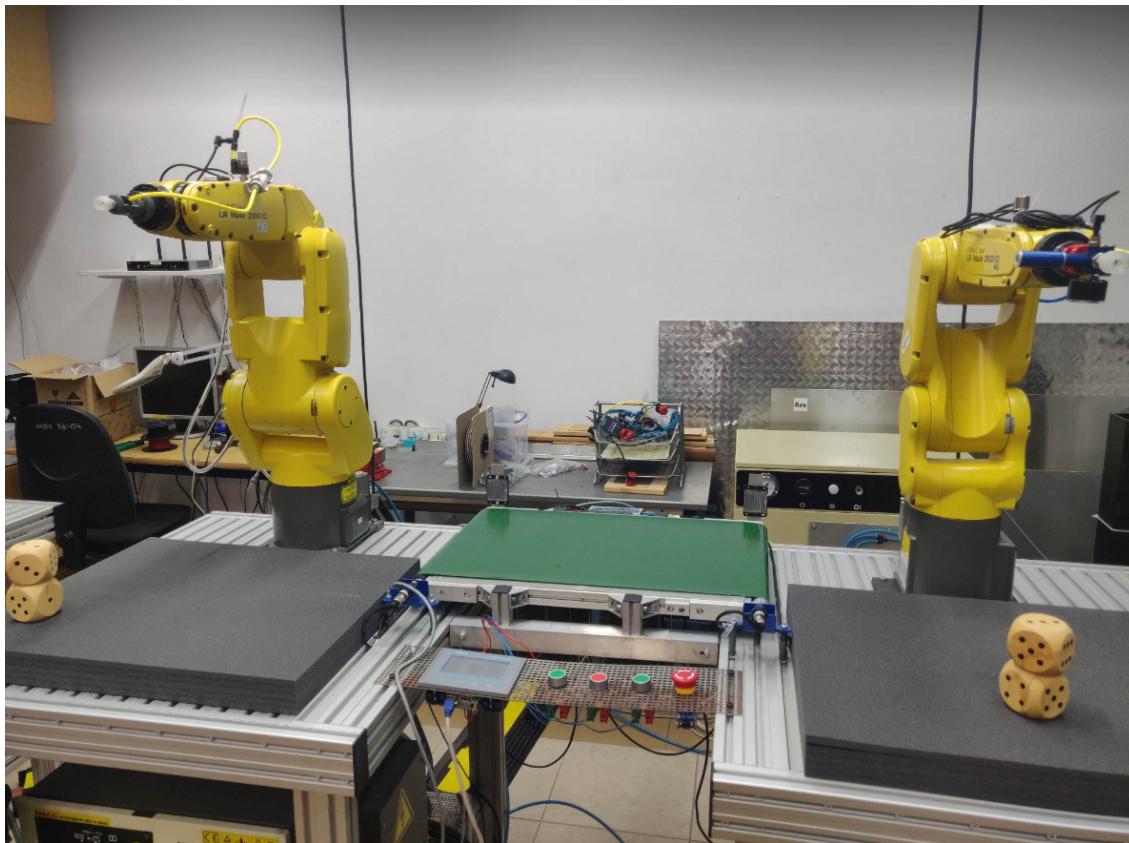


Rysunek 46: Początek – integracja rzeczywistych manipulatorów

4. Program testowy



Rysunek 47: Punkt przejściowy – integracja rzeczywistych manipulatorów



Rysunek 48: Punkt końcowy – integracja rzeczywistych manipulatorów

4. Program testowy

```
1 UFRAME_NUM=8
2 UTOOL_NUM=4
3 J P[1] 100% FINE
4
5 LBL[10]
6 VISION RUN_FIND 'WK_253995_1'
7 VISION GET_OFFSET 'WK_253995_1' VR[1] JMP LBL[10]
8 J PR[56] 100% FINE Offset,PR[55] VOFFSET,VR[1]
9 J PR[56] 100% FINE VOFFSET,VR[1]
10 CALL CLOSE
11 J PR[56] 100% FINE Offset,PR[55] VOFFSET,VR[1]
12 J PR[54] 100% FINE Offset,PR[55]
13 J PR[54] 100% FINE
14 CALL OPEN
15 J PR[54] 100% FINE Offset,PR[55]
16
17
18 LBL[11]
19 VISION RUN_FIND 'WK_253995_3'
20 VISION GET_OFFSET 'WK_253995_3' VR[2] JMP LBL[11]
21 J PR[58] 100% FINE Offset,PR[55] VOFFSET,VR[2]
22 J PR[58] 100% FINE VOFFSET,VR[2]
23 CALL CLOSE
24 J PR[58] 100% FINE Offset,PR[55] VOFFSET,VR[2]
25 J PR[54] 100% FINE Offset,PR[55]
26 J PR[54] 100% FINE Offset,PR[57]
27 CALL OPEN
28 J P[1] 100% FINE
```

Listing 5: Kod programu dla LR Mate 200iD/4S

```
1 UFRAME_NUM=9
2 UTOOL_NUM=1
3 J P[1] 100% CNT100
4
5
6 LBL[13]
7 VISION RUN_FIND 'WK_253995_3'
8 VISION GET_OFFSET 'WK_253995_3' VR[7] JMP LBL[13]
9 J PR[63:3] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[7]
10 J PR[63:3] 100% FINE VOFFSET,VR[7]
11 CALL OPEN
12 J PR[63:3] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[7]
13 J PR[80] 100% FINE Offset,PR[39:PICK_DICE]
14 J PR[80] 100% FINE
15 CALL CLOSE
16 J PR[80] 100% FINE Offset,PR[39:PICK_DICE]
17
18
19 LBL[15]
20 VISION RUN_FIND 'WK_253995_1'
21 VISION GET_OFFSET 'WK_253995_1' VR[9] JMP LBL[15]
22 J PR[61:1] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[9]
```

4. Program testowy

```
1 J PR[61:1] 100% FINE VOFFSET,VR[9]
2 CALL OPEN
3 J PR[61:1] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[9]
4 J PR[81] 100% FINE Offset,PR[39:PICK_DICE]
5 J PR[81] 100% FINE
6 CALL CLOSE
7 J PR[811] 100% FINE Offset,PR[39:PICK_DICE]
8
9
10 LBL[12]
11 VISION RUN_FIND 'WK_253995_4'
12 VISION GET_OFFSET 'WK_253995_4' VR[6] JMP LBL[12]
13 J PR[64:4] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[6]
14 J PR[64:4] 100% FINE VOFFSET,VR[6]
15 CALL OPEN
16 J PR[64:4] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[6]
17 J PR[83] 100% FINE Offset,PR[39:PICK_DICE]
18 J PR[83] 100% FINE
19 CALL CLOSE
20 J PR[83] 100% FINE Offset,PR[39:PICK_DICE]
21
22
23 LBL[14]
24 VISION RUN_FIND 'WK_253995_2'
25 VISION GET_OFFSET 'WK_253995_2' VR[8] JMP LBL[14]
26 J PR[62:2] 100% FINE Offset,PR[3:wys_kostki] VOFFSET,VR[8]
27 J PR[62:2] 100% FINE VOFFSET,VR[8]
28 CALL OPEN
29 J PR[62:2] 100% FINE Offset,PR[39:PICK_DICE] VOFFSET,VR[8]
30 J PR[82] 100% FINE Offset,PR[39:PICK_DICE]
31 J PR[82] 100% FINE
32 CALL CLOSE
33 J PR[73:ODL1] 100% FINE Offset,PR[39:PICK_DICE]
34
35
36 J P[1] 100% CNT20
37 J P[2] 100% CNT20
```

Listing 6: Kod programu dla LR Mate 200iC

Rozdział 5

Podsumowanie

Praca inżynierska dotyczy wykorzystania systemu wizyjnego iRVision oraz środowiska FANUC Roboguide do sterowania rzeczywistymi manipulatorami serii FANUC LR Mate. Praca pokazała, jak zintegrować oba narzędzia. Ponadto, szczegółowo opisano proces konfiguracyjny samego systemu wizyjnego iRVision oraz krok po kroku zaprezentowano proces tworzenia własnej komórki zrobotyzowanej w FANUC Roboguide. Praca zawiera opis przeprowadzonego testu przeniesienia programu demonstracyjnego z symulatora na prawdziwe manipulatory serii FANUC LR Mate 200 znajdujące się w Laboratorium Robotyki Politechniki Wrocławskiej. Pozwoliło to na ocenę skuteczności zastosowanego rozwiązania. Wyniki pracy wskazują, że zastosowanie systemu wizyjnego iRVision w połączeniu z środowiskiem FANUC Roboguide daje możliwość efektywnego sterowania/programowania manipulatorów FANUC. Praca może stanowić źródło do dalszych badań w kierunku automatyzacji procesów produkcyjnych w przemyśle. W pracy zrealizowano następujące cele:

- Zaimplementowano w środowisku symulacyjnym FANUC Roboguide odpowiednie komórki imitujące rzeczywiste stanowisko zrobotyzowane znajdujące się w Laboratorium Robotyki Politechniki Wrocławskiej.
- Zsynchronizowano rzeczywisty manipulator Fanuc LR Mate 200iC oraz system wizyjny iRVision ze środowiskiem symulacyjnym Roboguide.
- Skonfigurowano system wizyjny iRVision oraz zaimplementowano proces uczeniem modeli.
- Sformułowano zadanie programistyczne. Oprogramowano odpowiednie komórki w środowisku symulacyjnym.
- Zrealizowano programy demonstracyjne zarówno w symulatorze FANUC Roboguide jaki i na rzeczywistym stanowisku robotycznym.
- Zaimplementowano zadanie integracji robotów.

Spis rysunków

1	Stanowisko zrobotyzowane w sali 010	6
2	LR Mate 200iC	7
3	LR Mate 200iD/4S	7
4	FANUC Roboguide - ekran startowy	9
5	FANUC Roboguide - wybór procesu	10
6	FANUC Roboguide - nazwa celi	11
7	FANUC Roboguide - metody dodawania robota	12
8	FANUC Roboguide - wybór oprogramowania robota	13
9	FANUC Roboguide - wybór efektora robota	14
10	FANUC Roboguide - wybór robota	15
11	FANUC Roboguide - dodatkowe grupy ruchów	16
12	FANUC Roboguide - dodatkowe opcje robota	16
13	FANUC Roboguide - Podsumowanie	17
14	FANUC Roboguide - Pusta cela po stworzeniu projektu	18
15	FANUC Roboguide - Zakładki w projekcie	19
16	FANUC Roboguide - Dodawanie obiektu	19
17	FANUC Roboguide - Zmiana właściwości obiektu	20
18	FANUC Roboguide - Ręczne przesunięcie i obrót obiektu	21
19	FANUC Roboguide - Wybór opcji mierzenia	21
20	FANUC Roboguide - Mierzenie obiektu	22
21	FANUC Roboguide - Dodawanie części	22
22	FANUC Roboguide - Przypisanie części do obiektu	23
23	FANUC Roboguide - ToolFrame i UserFrame po zastosowaniu <i>RobotNeighbourhood</i>	25
24	FANUC Roboguide - Dodanie kamery	26
25	FANUC Roboguide - kamera	27
26	Efekt projektowania celi	28
27	FANUC Roboguide - Włączenie systemu wizyjnego w manipulatorze	29
28	FANUC Roboguide - Dodanie kamery do portu	30
29	Dodanie adresu IP robota do zaufanych witryn	31
30	Wybór opcji Vision Setup	32
31	Wybór opcji ustawień kamery	32
32	Konfiguracja kamery w iRVision	33
33	Definiowanie UserFrame'u dla systemu wizyjnego	34
34	Siatka kalibracyjna	35

35	Zaznaczanie punktów kalibracyjnych	36
36	Kalibracja kamery w iRVision	37
37	Ustawienia główne uczonego modelu	38
38	Uczenie rozpoznawania obiektu w iRVision	40
39	Ustawienie pozycji referencyjnej obiektu w iRVision	40
40	Dojazd manipulatorem do pozycji referencyjnej	41
41	FANUC Roboguide - program testowy	48
42	Rzeczywisty manipulator - program testowy	48
43	Początek – integracja manipulatorów, FANUC Roboguide	49
44	Punkt przejściowy – integracja manipulatorów, FANUC Roboguide	49
45	Punkt końcowy – integracja manipulatorów, FANUC Roboguide	50
46	Początek – integracja rzeczywistych manipulatorów	50
47	Punkt przejściowy – integracja rzeczywistych manipulatorów	51
48	Punkt końcowy – integracja rzeczywistych manipulatorów	51

Literatura

- [1] Roboty przemysłowe. <https://mlodytechnik.pl/eksperymenty-i-zadania-szkolne/wynalazczosc/30443-roboty-przemyslowe>.
- [2] Rewolucja przemysłowa. https://pl.wikipedia.org/wiki/rewolucja_przemyslowa.
- [3] Historia robotyki. <https://pfitowair.prv.pl/historia-robotyki>.
- [4] Roboty Fanuc z serii LR Mate. <https://www.fanuc.eu/it/en/robots/robot-filter-page/lrmate-series>.
- [5] Robot LR Mate 200iC. <https://robotsdoneright.com/fanuc/lr-mate-series/fanuc-lr-mate-200ic.html>.
- [6] Robot LR Mate 200iD. <https://robotyzacja.pl/produkty/lr-mate-200id4s>.
- [7] System FANUC iRVision. <https://one.fanuc.eu/l/442582/2021-07-22/brqlg9>.
- [8] FANUC Roboguide. <https://www.fanucamerica.com/products/robots/robot-simulation-software-fanuc-roboguide>.
- [9] Additional Motion Groups. <https://www.fanucamerica.com/products/robots/software>.
- [10] UserFrame. <https://motologix.yaskawa.eu.com/lib-1.24.0/features/user-frame/>.