

Systemy wizyjne w robotach przemysłowych na przykładzie FANUC iRVision*

Arkadiusz Mielczarek, Robert Muszyński[†]

Laboratorium Robotyki
Wydział Elektroniki
Politechnika Wrocławska

Spis treści

1	Cel ćwiczenia	1
2	Wymagania wstępne	2
3	System wizyjny iRVision	2
3.1	Połączenie	2
3.2	Procesy wizyjne	3
3.3	Nauka rozpoznawania detalu (lokalizator)	5
3.4	Ustalanie Reference Position	7
3.5	Użycie systemu wizyjnego programie	7
3.5.1	VISION RUN_FIND '...'	7
3.5.2	VISION GET_OFFSET '...' VR[...] JMP LBL[...]	7
3.5.3	VOFFSET,VR[...]	7
3.6	VR[...] .MODEL_ID	8
4	Uwagi odnośnie wykonywanych zadań	8
5	Zadania do wykonania	8
6	Sprawozdanie	9

1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie z możliwościami programowania robotów przemysłowych w oparciu o wizyjne sprzężenie zwrotne. System wizyjny używany w ćwiczeniu to FANUC iRVision dla robotów przemysłowych FANUC LR Mate 200iC.

W przypadku realizacji zadania w formie zdalnej należy je wykonać w środowisku ROBOGUIDE wykorzystując komórkę CELL_Viz umieszczoną na stronie kursu. Elementy

*Ćwiczenie laboratoryjne przeznaczone do realizacji w ramach kursu Robotyka (3) – data ostatniej modyfikacji: 26 września 2022.

[†]Katedra Cybernetyki i Robotyki

instrukcji różniące się w przypadku stacjonarnego i zdalnego trybu zostały oznaczone [STAC] i [ZDAL].

2 Wymagania wstępne

Przed przystąpieniem do realizacji ćwiczenia należy:

- zapoznać się z treścią instrukcji *Instrukcja bezpieczeństwa i higieny pracy w Laboratorium Robotyki* [2],
- (FANUC) zapoznać się z treścią instrukcji *Obsługa i programowanie robota FANUC LR Mate 200iC/200iD* [3],
- przypomnieć podstawowe pojęcia związane z robotami manipulacyjnymi, w tym:
 - układ współrzędnych,
 - przestrzeń konfiguracyjna,
 - przestrzeń robocza,
 - łańcuch kinematyczny,
 - kinematyka prosta,
 - kinematyka odwrotna,
- zapoznać się z niniejszą instrukcją.
- [ZDAL] zapoznać się z instrukcją *Programowanie robota FANUC w środowisku symulacyjnym Roboguide* [1].

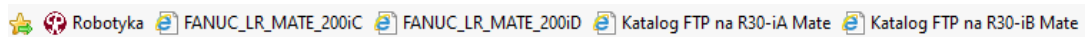
3 System wizyjny iRVision

iRVision jest komponentem robota pozwalającym na ekstrakcję i wykorzystanie informacji zawartych w obrazie z kamery. Interfejsem systemu iRVision jest przeglądarka internetowa.

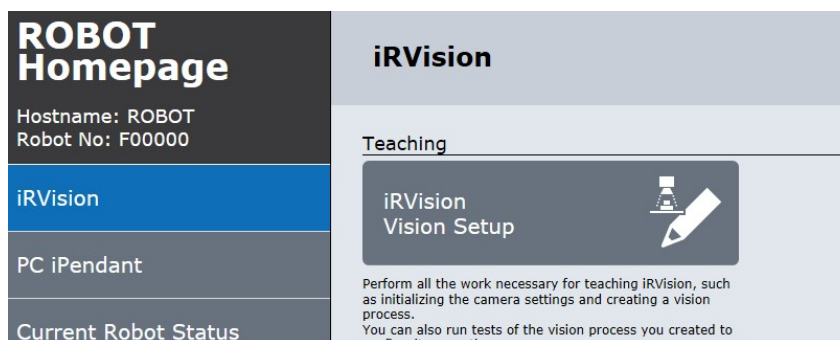
3.1 Połączenie

Aby połączyć się z systemem iRVision należy:

- [STAC] uruchomić przeglądarkę Internet Explorer, a następnie wybrać zakładkę FANUC_LR_MATE_iC.



- [ZDAL] w programie ROBOGUIDE wejść w menu Robot a następnie Web Browser. W lewej części przeglądarki wejść w iRVision.



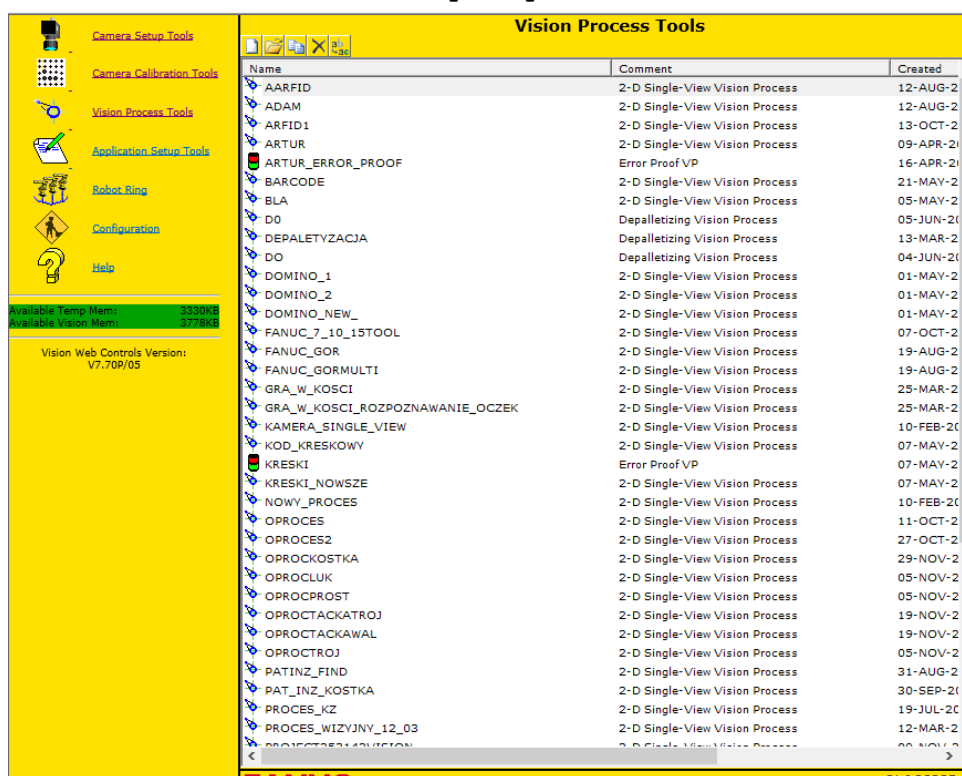
3.2 Procesy wizyjne

Proces wizyjny służy do pobierania pewnego typu informacji ze zdjęcia oraz zapisaniu go w odpowiednim rejestrze robota. Rodzajów dostępnych procesów wizyjnych jest sporo. Na potrzeby zadania odpowiednim procesem jest identyfikacja położenia detalu w środowisku. Aby utworzyć nowy proces wizyjny należy:

- [STAC] Na stronie robota, w części iRVision, należy wybrać link Vision Setup
- [ZDAL] Na stronie robota, w przeglądarce ROBOGUIDE, należy wybrać zakładkę Teaching a następnie iRVision Vision Setup.

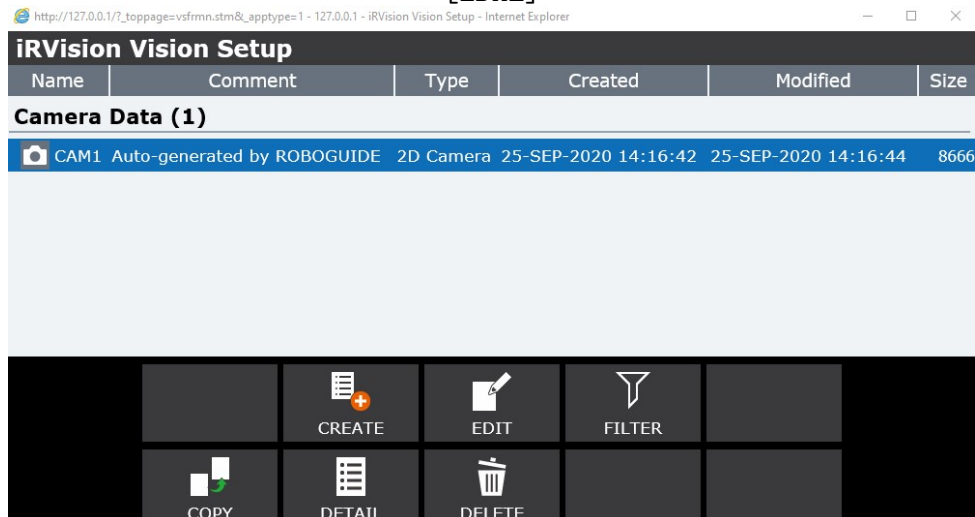
Na tej stronie znajdują się kamery podłączone do systemu, ich kalibracje oraz istniejące procesy wizyjne.

[STAC]



Name	Comment	Created
AARFID	2-D Single-View Vision Process	12-AUG-2
ADAM	2-D Single-View Vision Process	12-AUG-2
ARFID1	2-D Single-View Vision Process	13-OCT-2
ARTUR	2-D Single-View Vision Process	09-APR-2
ARTUR_ERROR_PROOF	Error Proof VP	16-APR-2
BARCODE	2-D Single-View Vision Process	21-MAY-2
BLA	2-D Single-View Vision Process	05-MAY-2
DO	Depalletizing Vision Process	05-JUN-2
DEPALETYZACJA	Depalletizing Vision Process	13-MAR-2
DO	Depalletizing Vision Process	04-JUN-2
DOMINO_1	2-D Single-View Vision Process	01-MAY-2
DOMINO_2	2-D Single-View Vision Process	01-MAY-2
DOMINO_NEW	2-D Single-View Vision Process	01-MAY-2
FANUC_7_10_15TOOL	2-D Single-View Vision Process	07-OCT-2
FANUC_GOR	2-D Single-View Vision Process	19-AUG-2
FANUC_GORMULTI	2-D Single-View Vision Process	19-AUG-2
GRA_W_KOSCI	2-D Single-View Vision Process	25-MAR-2
GRA_W_KOSCI_ROZPOZNAWANIE_OCZEK	2-D Single-View Vision Process	25-MAR-2
KAMERA_SINGLE_VIEW	2-D Single-View Vision Process	10-FEB-2
KOD_KRESKOWY	2-D Single-View Vision Process	07-MAY-2
KRESKI	Error Proof VP	07-MAY-2
KRESKI_NOWSZE	2-D Single-View Vision Process	07-MAY-2
NOWY_PROCES	2-D Single-View Vision Process	10-FEB-2
OPROCES	2-D Single-View Vision Process	11-OCT-2
OPROCES2	2-D Single-View Vision Process	27-OCT-2
OPROCKOSTKA	2-D Single-View Vision Process	29-NOV-2
OPROCLUK	2-D Single-View Vision Process	05-NOV-2
OPROCPROST	2-D Single-View Vision Process	05-NOV-2
OPROCTACKATROJ	2-D Single-View Vision Process	19-NOV-2
OPROCTACKAWAL	2-D Single-View Vision Process	19-NOV-2
OPROCTROJ	2-D Single-View Vision Process	05-NOV-2
PATINZ_FIND	2-D Single-View Vision Process	31-AUG-2
PAT_INZ_KOSTKA	2-D Single-View Vision Process	30-SEP-2
PROCES_KZ	2-D Single-View Vision Process	19-JUL-2
PROCES_WIZYJNY_12_03	2-D Single-View Vision Process	12-MAR-2

[ZDAL]



Name	Comment	Type	Created	Modified	Size
CAM1	Auto-generated by ROBOGUIDE	2D Camera	25-SEP-2020 14:16:42	25-SEP-2020 14:16:44	8666

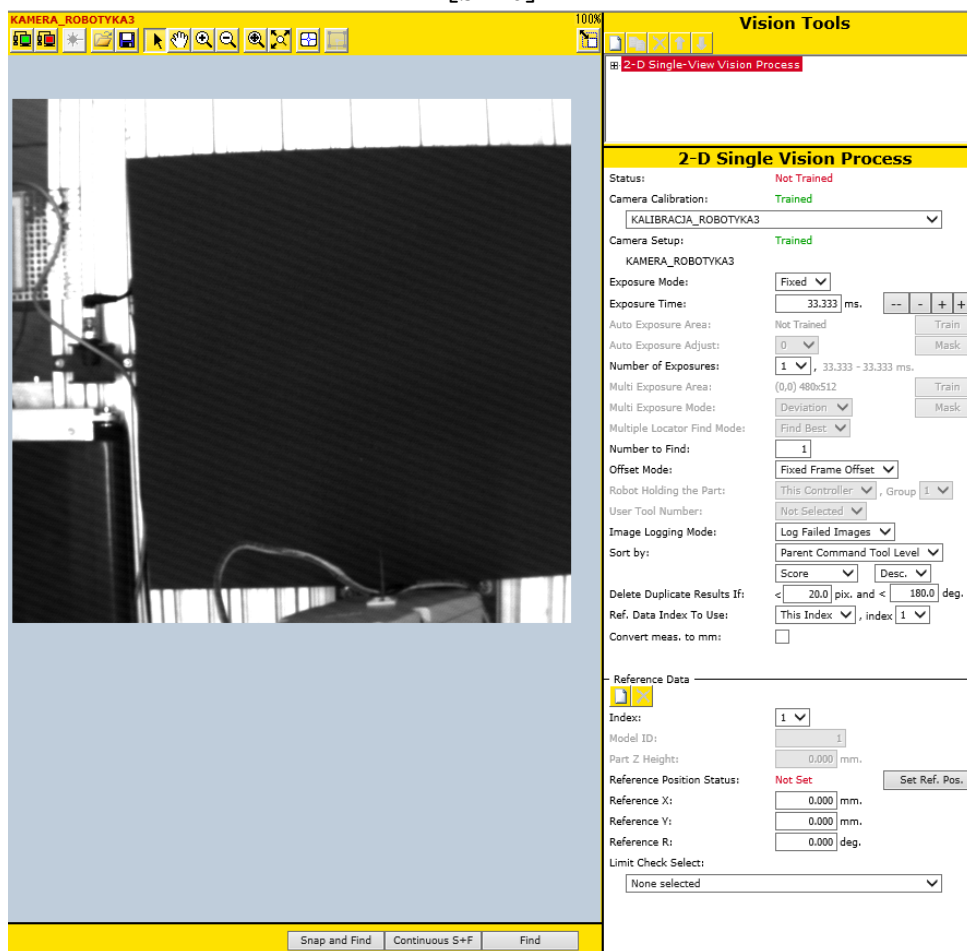
Aby utworzyć nowy proces wizyjny należy nacisnąć ikonę nowego procesu [STAC] lub CREATE [ZDAL]. Następnie wpisać nazwę, wybrać typ 2-D Single-View Vision Process i zatwierdzić. Klikając dwukrotnie na nazwie [STAC] lub EDIT [ZDAL] należy rozpocząć dostosowywanie parametrów procesu.

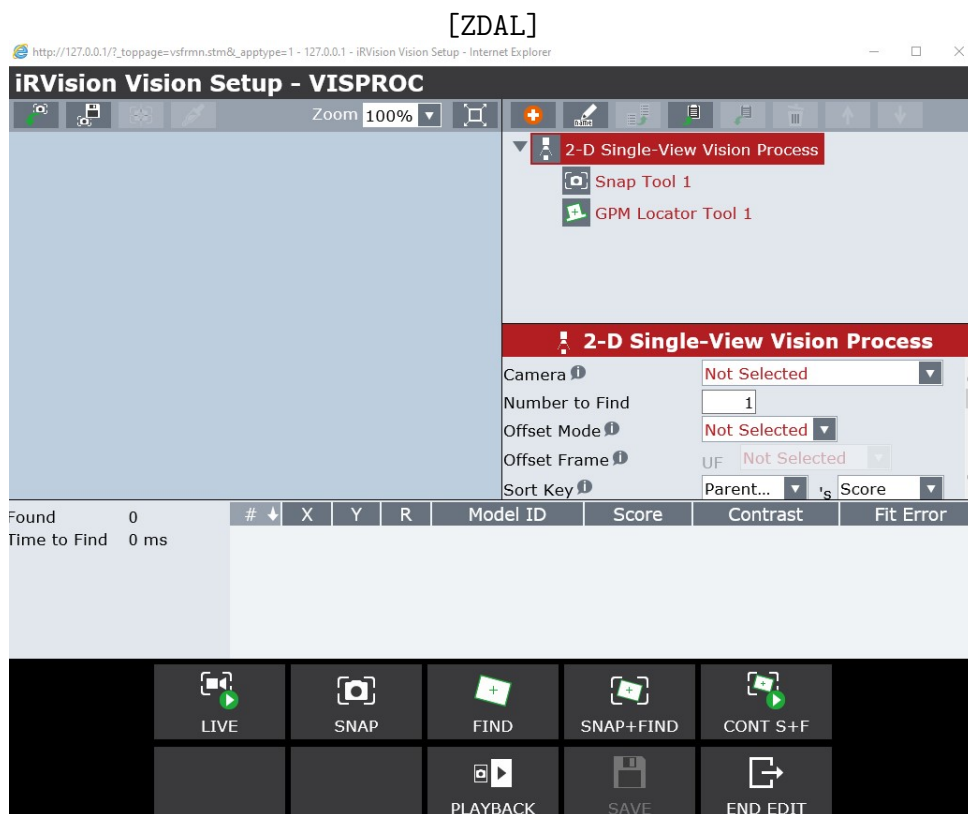
Parametry jakie należy wybrać to:

Opcja	[STAC]	[ZDAL]
Camera	KAMERA_ROBOTYKA3	jedyna istniejąca
Camera Calibration	KALIBRACJA_ROBOTYKA3	—
Offset Mode	Fixed Frame Offset	Fixed Frame Offset
Part Z Height	—	wysokość detalu

Reference Position Status – położenie ”bazowe” detalu. Będzie to położenie wobec którego proces wizyjny będzie zwracał lokalizacje. Położenie to można zdefiniować dopiero po zdefiniowaniu pierwszego lokalizatora detalu i opisano to w punkcie 3.4.

[STAC]



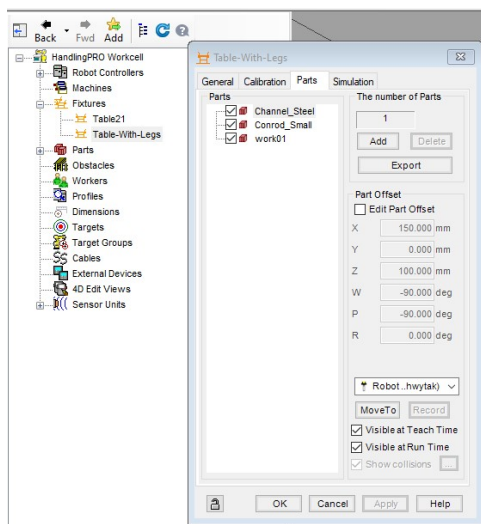


3.3 Nauka rozpoznawania detalu (lokalizator)

Do rozpoznawania detalu służy GPM Locator Tool. Aby nauczyć system nowego detalu należy:

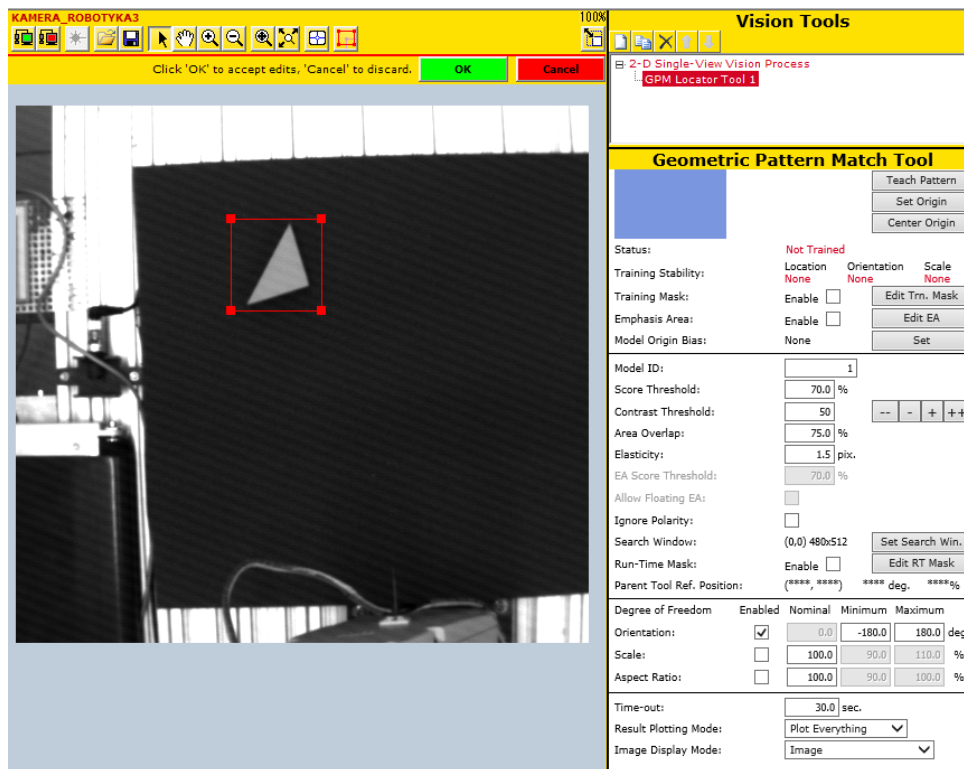
1. umieścić ramię robota poza obszarem kamery.
2. umieścić w wybranym miejscu bazowym detal którego chcemy nauczyć system, inne usunąć z obszaru kamery.

[ZDAL] Aby wykonać to w symulatorze należy rozwinąć listę Fixtures w drzewie projektu po lewej stronie. Kliknąć prawym przyciskiem na Table-With-Legs i wybrać Properties. Wejść w zakładkę Parts i odznaczyć opcje Visible at Teach Time oraz Visible at Run Time dla detali których obecnie nie chcemy na stole.

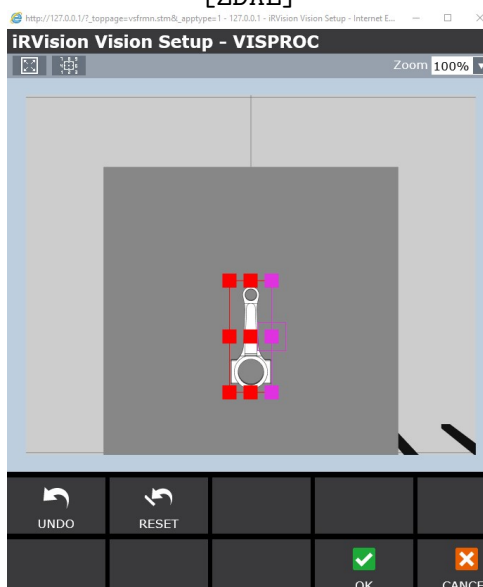


3. przejść do zakładki GPM Locator Tool.
4. zrobić zdjęcie kamerą za pomocą przycisku SNAP.
5. nacisnąć przycisk Teach Pattern [STAC] lub Teach [ZDAL]. Zaznaczyć detal i zaakceptować.

[STAC]



[ZDAL]



Można ręcznie zedytować powstałą maskę detalu poprzez Edit Trn. Mask [STAC] lub GEdit [ZDAL].

Opcjami Orientation, Scale i Aspect można określić wykrywanie detali obróconych o dany kąt, przeskalowanych (przybliżonych, oddalonych) oraz zniekształconych (np. poprzez nieznaczne obrócenie w innej osi niż Z lub zniekształcenie przez soczewkę kamery).

Po ustawieniu pożądanych opcji należy kliknąć **SAVE**.

Aby znaleźć zapamiętany element należy nacisnąć **SNAP+FIND**.

W przypadku, gdy znaleziono więcej niż jeden pasujący detal, można podnieść próg akceptacji **Score Threshold**.

Mając zdefiniowany detal można dokończyć konfigurację procesu wizyjnego o znalezienie bazowego położenia.

3.4 Ustalenie Reference Position

Aby wybrać położenie bazowe należy najpierw zdefiniować lokalizator **GPM Locator Tool 1**. Po zdefiniowaniu należy kliknąć **SNAP+FIND** (będąc w **2-D Single-View Vision Process**, nie w **GPM Locator Tool**) a następnie wybrać zlokalizowany detal. [STAC] W przypadku rzeczywistego robota, po zlokalizowaniu detalu, a przed zapisaniem pozycji należy ręcznie ustawić manipulator w pozycji pozwalającej na pobranie/oddanie detalu znajdującego się w pozycji referencyjnej i zapamiętanie jej w programie jako punkt lub rejestr pozycyjny. [UWAGA] Należy upewnić się, że na manipulatorze wybrane są odpowiednie **TOOL_FRAME** (**TOOL_FRAME1 : suction_thingy**) oraz **USER_FRAME** (**USER_FRAME9 : ROB3_VIZ**).

3.5 Użycie systemu wizyjnego programie

Poniżej opisano trzy funkcje możliwe do wywołania z poziomu programu.

3.5.1 VISION RUN_FIND '...'

Funkcja jest do znalezienia w **INST** → **VISION**. Argumentem jest proces wizyjny. Służy do wykonania zdjęcia przez kamerę dla danego procesu.

3.5.2 VISION GET_OFFSET '...' VR[...] JMP LBL[...]

Funkcja jest do znalezienia w **INST** → **VISION**. Argumentami są:

'...' proces wizyjny,

VR[...] rejestr wizyjny do którego należy wpisać wynik działania,

LBL[...] etykieta (label) do której program ma skoczyć jeśli nie udało się otrzymać wyniku.

Funkcja służy do określania przesunięcia względem pozycji bazowej elementu rozpoznawanego przez proces wizyjny.

3.5.3 VOFFSET,VR[...]

Opcja związana z instrukcją ruchu **POINT**. Argumentem jest numer rejestru wizyjnego. Do punktu określonego w instrukcji ruchu (w tym przypadku punktu referencyjnego procesu wizyjnego) dodaje offset odczytany z rejestru wizyjnego. Sposób użycia i działanie analogiczne jak dla zwykłej opcji offsetu instrukcji ruchu z rejestru pozycyjnego **PR**.

3.6 VR[...].MODEL_ID

Funkcja jest do znalezienia w INST \rightarrow VISION. Funkcja zwraca MODEL_ID elementu znalezionej przez proces wizyjny używający danego rejestru wizyjnego. MODEL_ID elementów można ustawić w opcjach lokalizatora GPM Locator Tool.

4 Uwagi odnośnie wykonywanych zadań

- manipulatorem należy poruszać z małymi prędkościami,
- w każdym programie należy na początku określić układy: narzędzia (UTOOL_NUM=1 : `suction_thingy`) i użytkownika (UFRAME_NUM=9 : ROB3_VIZ). Układy powinny się pokrywać z aktualnymi (SHIFT + COORD). Na potrzeby zadania zostało zdefiniowane narzędzie (TOOL_FRAME1 : `suction_thingy`) oraz układ współrzędnych użytkownika (USER_FRAME9 : ROB3_VIZ).
- punkty zapamiętane za pomocą instrukcji `Point` pamiętają do jakiego User Frame się odnoszą. Inaczej jest w przypadku rejestrów pozycyjnych. W przypadku użycia takich w programie, należy upewnić się, że program jest uruchamiany z odpowiednim User Frame.
- [ZDAL] na potrzeby zadania zostały zdefiniowane podprogramy umożliwiające chwytanie (GRASP) i puszczenie (RELEASE) elementu. Takie podprogramy są zdefiniowane dla każdego detalu osobno.
- [ZDAL] W komórce CELL_Viz na stole odkładczym detale są widoczne w położeniu docelowym.

5 Zadania do wykonania

1. Na podstawie swobodnego ruchu manipulatorem określić w przybliżeniu położenie i orientację używanego układu współrzędnych użytkownika (User Frame).
2. Zdefiniować proces wizyjny pozwalający na zidentyfikowanie oraz lokalizację detalu wskazanego przez prowadzącego.
3. Napisać program pozwalający na odkładanie detalu losowo położonego na blacie w określone miejsce.
4. Uogólnić powyższy program na przypadek z kilkoma różnymi detalami.
- 5.* Uogólnić powyższy program dla przypadku więcej niż jednego detalu takiego samego rodzaju.

Powyższa lista zadań jest propozycją, ostateczny zakres zadań do wykonania zostanie podany przez prowadzącego na zajęciach.

6 Sprawozdanie

Sprawozdanie z przebiegu ćwiczenia powinno zawierać:

- Imię i nazwisko autora, numer i termin grupy, skład grupy, temat ćwiczenia, datę wykonania ćwiczenia, datę dostarczenia sprawozdania.
- Cel ćwiczenia.
- Opis zrealizowanych zadań:
 - co dane zadanie miało na celu,
 - sposób przeprowadzenia zadania,
 - opracowanie otrzymanych wyników,
 - uzyskany rezultat – wnioski.
- Wnioski końcowe.

Literatura

- [1] Mirela Kaczmarek. *Programowanie robota FANUC w środowisku symulacyjnym Roboguide*. Katedra Cybernetyki i Robotyki, Politechnika Wrocławska, Programowanie robota FANUC w środowisku symulacyjnym Roboguide, 2020.
- [2] Marek Wnuk. *Zasady bezpiecznej pracy w Pracowni Robotyki - s. 010 C-3*. Katedra Cybernetyki i Robotyki, Politechnika Wrocławska, Zasady bezpiecznej pracy w Pracowni Robotyki - s. 010 C-3.
- [3] Katarzyna Zadarnowska, Jacek Jagodziński, Arkadiusz Mielczarek. *Obsługa i programowanie robota FANUC LR Mate 200iC*. Katedra Cybernetyki i Robotyki, Politechnika Wrocławska, Obsługa i programowanie robota FANUC LR Mate 200iC, 2018.