

# Universal Robot UR3\*

Katarzyna Zadarnowska, Joanna Ratajczak<sup>†</sup>  
Laboratorium Robotyki  
Wydział Elektroniki  
Politechnika Wrocławska

## 1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z metodyką pracy z robotem UR3 oraz podstawowymi funkcjonalnościami oferowanymi przez sterownik tego robota.

## 2 Wymagania wstępne

Przed przystąpieniem do realizacji ćwiczenia należy:

- zapoznać się z Zasadami bezpiecznej pracy w Pracowni Robotyki, s.010 C-3,
- znać mechanizmy awaryjnego wyłączenia robota,
- zapoznać się z niniejszą instrukcją.

## 3 Informacje ogólne i opis stanowiska

UR3 jest 6-osiowym robotem przemysłowym posiadającym wbudowany system bezpieczeństwa, dzięki czemu może pracować ramię w ramię z człowiekiem. Posiada udźwig rzędu 3[kg] i zasięg 500[mm], jest wrażliwy na działanie siły i można go ustawić tak, aby natychmiast wstrzymał pracę, kiedy napotka opór rzędu zaledwie 50[N]. To umożliwia pracę z robotem bez konieczności używania drogich osłon bezpieczeństwa. Zakres jego pracy wynosi  $\pm 360^\circ$  na każdym przegubie obrotowym, zaś na ostatnim przegubie: nieskończoność, co umożliwia technologia InfiniteSpin, w którą został wyposażony robot. Maksymalne prędkości dla przegubów nadgarstka wynoszą 360[°/s], natomiast dla pozostałych przegubów 180[°/s]. Narzędzie osiąga prędkość 1[m/s] i 990[mm/s]. Zadania, w których najlepiej sprawdza się model UR3 obejmują: montaż niewielkich przedmiotów, klejenie, wkręcanie i odkręcanie, obsługę narzędzi, lutowanie, malowanie.

W skład stanowiska laboratoryjnego wchodzi: manipulator UR3 z chwytakiem Robotiq 2F-85, sterownik CB3, oraz panel operatora wraz z oprogramowaniem PolyScope (patrz Rys. 1). Manipulator UR3 z chwytakiem i kamerą stanowią wyłącznie układ sensoryczno-wykonawczy,

---

\*Ćwiczenie laboratoryjne przeznaczone do realizacji w ramach kursu Robotyka (3) – data ostatniej modyfikacji dokumentu 17 września 2020

<sup>†</sup>Katedra Cybernetyki i Robotyki

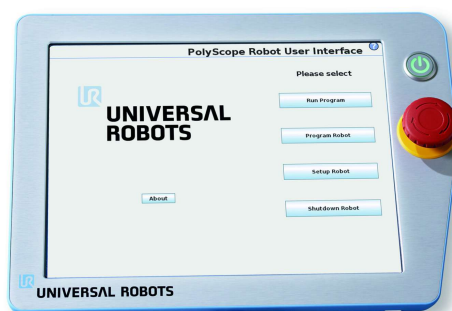
natomiast przetwarzanie percepcji na działanie odbywa się na jednostce centralnej sterownika CB3. To właśnie tam zaimplementowane są algorytmy śledzenia trajektorii, planowania ruchu w przestrzeni konfiguracyjnej i roboczej. Język programowania cobota, omówiony w dalszej części instrukcji, dotyczy głównie komponentów architektury układu sterowania związanych z planowaniem ruchu. Dodatkowo, w sterowniku znajduje się tablica kontroli bezpieczeństwa, układy wejścia/wyjścia oraz mikrokomputer z wejściem USB. Do sterowania jednostką centralną wykorzystuje się panel sterujący teachPendant. TeachPendant jest ręcznym programatorem służącym do poruszania manipulatorem (sterowanie ręczne), wprowadzania i testowania programów oraz sterowania urządzeniami peryferyjnymi. Panel posiada duży, kolorowy wyświetlacz dotykowy, przycisk STOPu awaryjnego oraz włącznik zasilania.



Rysunek 1: Cobot: manipulator UR3, sterownik CB3, panel operatora

### 3.1 Uruchamianie, praca ręczna

W trakcie pracy robota, ze względów bezpieczeństwa, nie wolno przebywać w strefie działania robota. W celu uruchomienia robota należy wcisnąć przycisk zasilania. Pojawi się okno dialogowe z informacją `The Robot Cannot Proceed with Normal Operation`. W celu kontynuowania pracy należy wcisnąć przycisk `Go to initialization screen`. Aby włączyć zasilanie przegubów należy wcisnąć przycisk `ON` oraz dalej przycisk `START` inicjujący pracę robota. Podczas uruchamiania robota wydaje on dźwięk i nieznacznie się porusza w chwili zwolnienia hamulców. Robot jest gotowy do pracy. Wciśnięcie przycisku `OK` spowoduje przejście do głównego menu (patrz Rys. 2). Stąd można programować robota, uruchamiać gotowe programy, zmieniać ustawienia robota oraz wyłączyć robota. Zasilanie ramienia robota można wyłączyć poprzez dotknięcie przycisku `OFF` na ekranie inicjowania. Zasilanie ramienia robota jest także wyłączane automatycznie w chwili wyłączenia skrzynki sterowniczej. Należy pamiętać, że robot jest wyposażony w wiele funkcji właściwych bezpieczeństwu, które można wykorzystać do ograniczenia ruchów przegubów oraz punktu centralnego narzędzia (TCP) robota. System bezpieczeństwa obejmuje dwa konfigurowalne tryby bezpieczeństwa: normalny (domyślnie aktywny) i ograniczony (aktywny kiedy TCP robota znajduje się poza płaszczyzną wyzwania trybu ograniczonego). W przypadku naruszenia granic bezpieczeństwa konieczne jest ponowne uruchomienie systemu bezpieczeństwa. Jeśli w chwili uruchomienia system znajduje się poza granicami bezpieczeństwa, następuje przejście do specjalnego trybu przywracania. W trybie przywracania robot nie może pracować w trybie automatycznym, ale ramię robota można umieścić w trybie pracy ręcznej w bezpiecznym obszarze. Aby to zrobić należy użyć trybu ruchu swobodnego albo zakładki `Move` oprogramowania. W trybie ruchu swobodnego manewrujemy robotem jednocześnie przytrzymując wciśnięty przycisk `FreeDrive`, który znajduje się z tyłu robota (patrz Rys. 3), bądź na ekranie wyświetlacza. Korzystając z zakładki `Move` na panelu operatorskim nawigujemy robotem używając strzałek.



Rysunek 2: Panel operatora – menu główne

Jeśli w trybie ruchu swobodnego ramię robota zbliży się do ustalonych granic bezpiecznej pracy robota, użytkownik odczuje w ruchu przeciwdziałającą siłę, która informuje użytkownika, że bieżąca pozycja lub prędkość są bliskie ograniczeniom. Jeśli jednak użytkownik przyłoży wystarczająco dużą siłę do ramienia robota, taka granica może zostać przekroczona. Wielkość siły zwiększa się w miarę zbliżania się ramienia robota do ograniczenia. Kliknięcie sumy kontrolnej (prawy górny róg) pozwala na wyświetlenie szczegółowych informacji o aktualnie obowiązującej konfiguracji bezpieczeństwa.



Rysunek 3: Panel operatora – przycisk freedrive

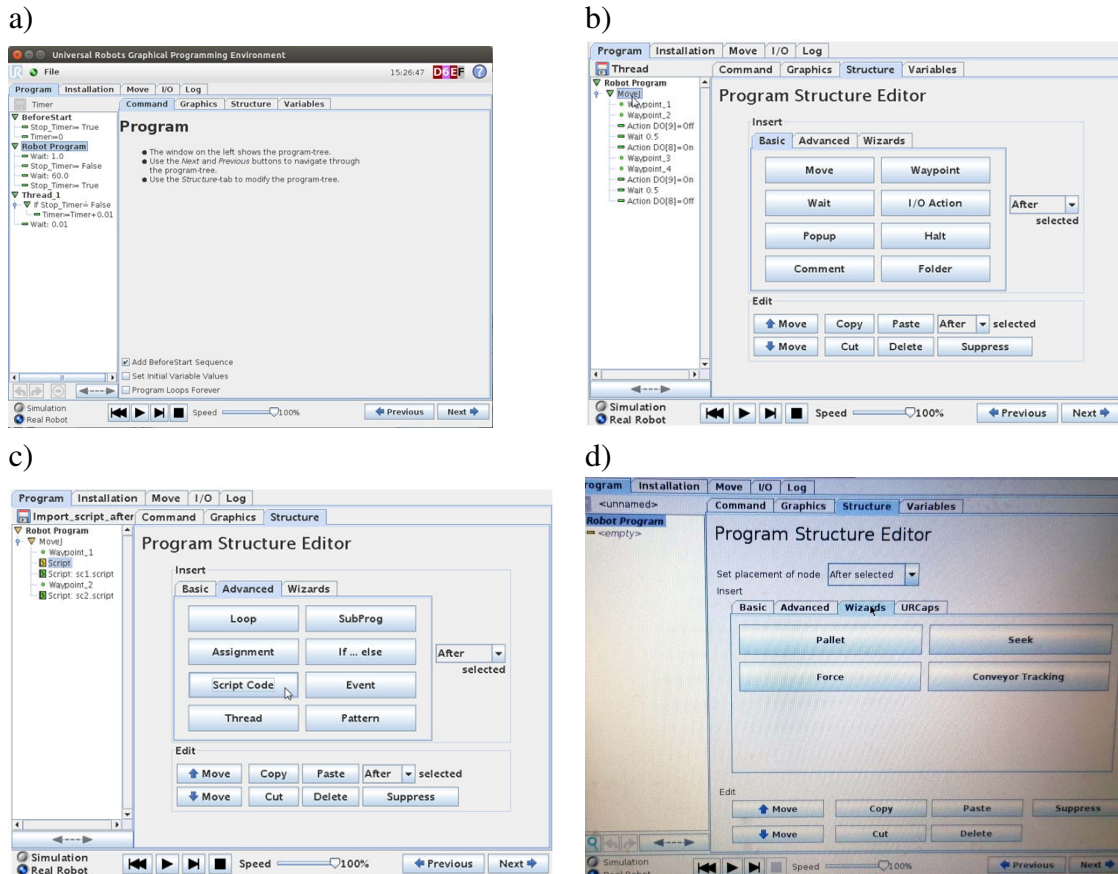
## 3.2 Programowanie

W celu rozpoczęcia programowania należy w menu głównym robota wybrać Program robot i dalej wybrać przycisk Empty program. W lewej części ekranu wyświetlany jest program składający się z drzewa instrukcji ruchu, wejścia/wyjścia, obsługi rejestrów itp. (patrz Rys.4a)). Zadanie realizowane jest poprzez sekwencyjne wykonywanie kolejnych instrukcji. Nawigacja w obrębie drzewa instrukcji odbywa się za pomocą przycisków Previous/Next. Drzewo instrukcji można modyfikować używając zakładki Structure. Edytor struktury programu (patrz Rys.4b) – d) oferuje wstawianie, przenoszenie, kopiowanie i usuwanie poleceń. Aby wstawić nowe polecenie, należy:

1. wybrać polecenie w drzewie instrukcji, poniżej lub powyżej którego będziemy chcieli umieścić nowe polecenie,

2. zdecydować, czy nowe polecenie należy wstawić powyżej czy poniżej wybranego polecenia,
3. wybrać z zakładki Basic/Advanced/Wizards/URCaps nowe polecenie.

W celu zmiany/edycji parametrów polecenia należy wybrać dane polecenie w drzewie instrukcji i przejść do zakładki Command. Polecenia można przesuwać, kopiować i usuwać za pomocą przycisków w ramce edycji. Jeśli posiada ono polecenia podległe (co oznacza trójkąt umieszczony obok polecenia nadrzędnego), to wszystkie one są również przesuwane, klonowane lub usuwane. Drzewo programu zawiera wizualne elementy, które informują o poleceniu



Rysunek 4: Panel operatora – a) drzewo instrukcji; b) zakładka Structure/Basic; c) zakładka Structure/Advanced; d) zakładka Structure/Wizards

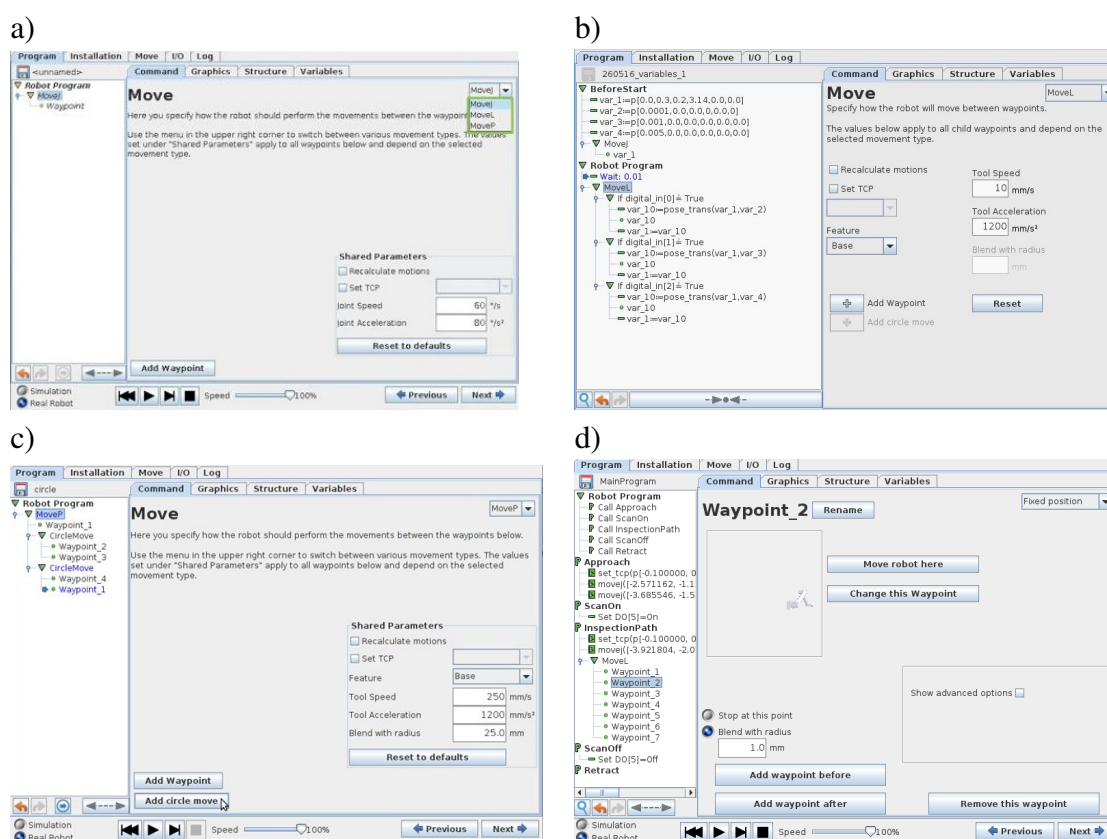
obecnie wykonywanym przez sterownik robota. Z lewej strony ikony polecenia widoczna jest mała ikona wskaźnika, a nazwa wykonywanego polecenia oraz wszystkich poleceń, dla których wykonywane polecenie jest podrzędne, są wyróżniane niebieskim kolorem. Pomaga to użytkownikowi w odszukaniu wykonywanego polecenia w drzewie.

### 3.3 Instrukcje ruchu (*MoveL*, *MoveP*, *MoveJ*)

Najistotniejszą częścią programu jest realizacja ruchu narzędzia. W PolyScope ruchy narzędzia są definiowane za pomocą punktów orientacyjnych (tzw. *Waypoints*), czyli punktów w przestrzeni roboczej robota. I tak, instrukcja ruchu robota mieści się w zakładce *Basic* pod przyciskiem *Move* (Rys. 4b)). Pojedyncza instrukcja ruchu definiuje ruch robota tzn. przemieszczenie środkowego punktu narzędzia (TCP) z obecnej pozycji, w której znajduje się TCP, do zadanej

pozycji wewnątrz przestrzeni roboczej z określoną prędkością przesuwu i o określonym sposobie ruchu. W celu realizacji funkcji przemieszczenia TCP robota należy wybrać jeden z trzech rodzajów ruchu: ruch liniowy narzędzia (MoveL - Linear Motion), ruch liniowy w przestrzeni konfiguracyjnej (MoveJ - Joint Move) oraz ruch liniowy ze stałą prędkością (MoveP - Move Process).

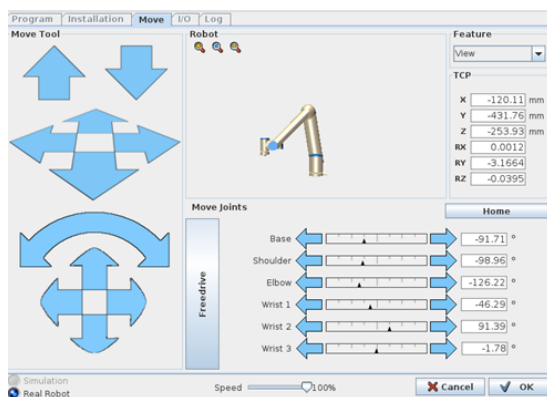
Jak już wcześniej wspomniano, zakładka Structure umożliwia wybór poszczególnych komend/instrukcji, w tym instrukcji ruchu, natomiast szczegóły dotyczące danej instrukcji ruchu ustala się w zakładce Command. I tak, po wybraniu instrukcji ruchu Move, w prawym górnym rogu zakładki Command (Rys. 5a)) znajduje się rozwijane menu, które pozwala uszczegółowić rodzaj ruchu (MoveL, MoveJ, MoveP; Rys. 5a-c). Dalej, w drzewie programu wybieramy WayPoint (w obrębie danej instrukcji ruchu) - punkt docelowy, który będziemy chcieli zapamiętać. W zakładce Command tej instrukcji (Rys. 5d)) wybieramy przycisk Change this Waypoint, a następnie przeprowadzamy robota do zadanej pozycji korzystając z jednej z następujących metod:



Rysunek 5: a) Instrukcja MoveJ; b) instrukcja MoveL; c) instrukcja MoveP; d) instrukcja Waypoint

1. Wybieramy zakładkę Move (Rys. 6) i używając strzałek nawigujemy robotem w jego przestrzeni zadaniowej bądź przegubowej. Tutaj można również bezpośrednio zadawać (względne lub bezwzględne) wartości kątów w przegubach bądź położenia i orientacje narzędzia TCP.
2. Przytrzymując wciśnięty przycisk Freedrive znajdujący się z tyłu (patrz Rys. 3) lub na ekranie (patrz Rys. 6) panelu operatora jednocześnie manewrujemy robotem (przegubami/narzędziem) tak, by umieścić TCP w zadanej pozycji.





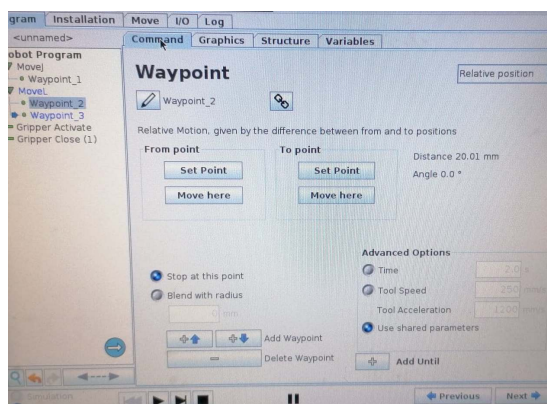
Rysunek 6: Zakładka Move

Przycisk OK zatwierdza wybraną pozycję. Dodatkowymi parametrami instrukcji *Move* są przyspieszenie i prędkość z jaką będzie odbywał się ruch pomiędzy punktami orientacyjnymi.

Jeśli chcemy w obrębie jednej instrukcji ruchu zrealizować przejście pomiędzy kilkoma punktami pośrednimi, należy w zakładce *Command* wybranego polecenia *Waypoint* wybierać kolejno przycisk *Add waypoint after* \ *Add waypoint before*, który umożliwia dodawanie kolejnych punktów trasy.

Aby zrealizować ruch wzdłuż łuku, wybieramy instrukcję ruchu *Move* i w zakładce *Command* tej instrukcji wybieramy w rozwijanym menu instrukcję *MoveP*. Dalej, wybieramy tę instrukcję w drzewie programu i w jej zakładce *Command* wybieramy *Add circle move* (Rys. 5c)). Ruch po łuku odbywa się z zadaniem dwóch punktów trasy. Pierwszy z nich definiuje środek łuku, natomiast drugi - koniec łuku. Robot w tym przypadku zaczyna ruch z pozycji bieżącej i przechodzi poprzez oba zadane punkty.

Istnieje również możliwość zadania względnego punktu orientacyjnego, czyli pozycji zadawanej względem poprzedniej pozycji robota (na przykład „dwa centymetry w lewo”). Aby zdefiniować taki punkt należy wybrać opcję *Relative Position* w prawym górnym rogu rozwijanego menu zakładki *Command* polecenia *Waypoint*; (patrz Rys. 7). Zadawany dystans i orientacja względem pozycji poprzedniej są w tym przypadku wyliczane na podstawie różnicy między pozycjami *From point* i *To point*. Należy tu pamiętać, że powtarzanie ruchu zadanego poprzez pozycję względną może przesunąć ramię robota poza jego przestrzeń roboczą.



Rysunek 7: Zadawanie względnego punktu orientacyjnego

Bieżące położenie ramienia robota jest przedstawione w grafice 3D w zakładce *Graphics*. Rysunek 3D ramienia robota pokazuje aktualną pozycję manipulatora, a jego cień pokazuje jego

ewentualną pozycję docelową. Naciśnięcie ikony szkła powiększającego pozwala na przybliżenie lub oddalenie, a przeciągnięcie palcem po ekranie na zmianę widoku robota. Jeśli określona pozycja docelowa punktu TCP robota jest blisko płaszczyzny bezpieczeństwa lub wyzwalającej, albo orientacja narzędzia robota jest w pobliżu limitu granicznego orientacji, przedstawiana jest reprezentacja 3D limitu granicznego w pobliżu. Płaszczyzny bezpieczeństwa są przedstawiane w kolorach żółtym i czarnym z małą strzałką normalną do płaszczyzny, która wskazuje tę stronę, po której dozwolone jest pozycjonowanie punktu TCP. Płaszczyzny wyzwalające są przedstawiane w kolorach niebieskim i zielonym z małą strzałką, która wskazuje tę stronę płaszczyzny, po której aktywne są limity trybu normalnego. Granice orientacji narzędzia są wizualizowane za pomocą sferycznego stożka oraz wektora określającego bieżącą orientację narzędzia. Wnętrze stożka odpowiada polu dopuszczalnemu dla orientacji narzędzia (wektora). Jeśli docelowy punkt TCP narusza lub bardzo zbliża się do limitu granicznego, wizualizacja tego limitu zmienia kolor na czerwony.

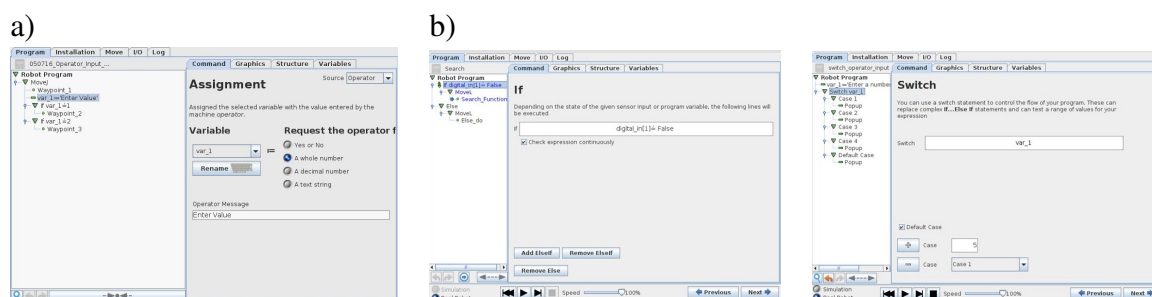
### 3.4 Instrukcje z zakładki Advanced

Pętla (Loop) umożliwia wielokrotne wykonywanie zawartych w niej poleceń. W zależności od wybranej opcji zawarte w pętli polecenia programu są zapętłone w nieskończoność, wykonywane określoną liczbę razy albo dopóki podany warunek jest prawdziwy. W drugim przypadku tworzona jest dedykowana zmienna pętli, której można używać w wyrażeniach wewnątrz pętli. Zmienna przybiera kolejne wartości od 0 do  $N - 1$ . W przypadku wykonywania pętli z użyciem wyrażenia jako warunku zakończenia, oprogramowanie PolyScope udostępnia opcję ciągłego szacowania tego wyrażenia.

Podprogram (SubProg) może przechowywać fragmenty programu, które chcemy wykorzystać w kilku miejscach. Wywołanie podprogramu powoduje wykonanie wierszy podprogramu, a następnie powrót do następnego wiersza programu.

Assignment (patrz Rys. 8a)) nadaje wartości zmiennym. Instrukcja przypisuje zmiennej znajdującej się z lewej strony, obliczoną wartość wyrażenia z prawej strony.

Konstrukcja `if... else` (Rys. 8b)) może zmienić zachowanie robota w zależności od wejść czujników lub wartości zmiennych. Edytor wyrażeń służy do opisanego warunku, przy którym robot powinien przejść do poleceń podrzędnych instrukcji warunkowej. Każda instrukcja warunkowa `If` może mieć kilka instrukcji `ElseIf` i jedną instrukcję `Else`. Można je dodawać za pomocą przycisków na ekranie. Pole `Check expression continuously` w sposób ciągły umożliwia ocenę warunków instrukcji `If` oraz `ElseIf`, podczas gdy wykonywane są zawarte w nich wiersze.



Rysunek 8: a) Instrukcja Assignment; b) instrukcja `if...else`; instrukcja Switch Case

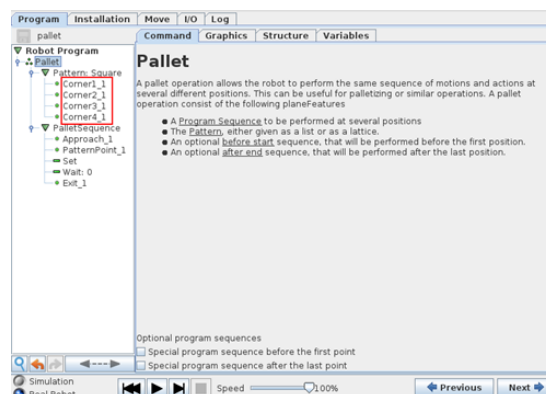
Polecenie `Script Code` umożliwia dostęp do wbudowanego języka skryptowego czasu rzeczywistego wykonywanego przez sterownik robota. Instrukcję jego używania można znaleźć

w podręczniku skryptów na stronie (<https://www.universal-robots.com/support/>). W przypadku wybrania w lewym górnym rogu opcji File można tworzyć i edytować pliki z programami w skryptach.

Konstrukcja Switch Case (Rys. 8c)) może zmienić zachowanie robota w zależności od wejść czujników lub wartości zmiennych. Edytor wyrażeń służy do opisanie warunku podstawowego i zdefiniowania przypadków, przy których robot powinien przejść do poleceń podrzędnych danego przełącznika.

Polecenia Pattern umożliwia zdefiniowanie w łatwy sposób grupy punktów w przestrzeni roboczej. W ramach tego polecenia możemy zdefiniować jeden z czterech typów wzorców: Line, Square, Box i List. Pierwsze trzy mogą zostać użyte do pozycjonowania narzędzia w punktach ułożonych w regularny wzorec. Regularne wzorce są definiowane za pomocą punktów charakterystycznych definiujących początek i koniec krawędzi wzorca oraz liczbę punktów pośrednich wzdłuż danej krawędzi. Następnie na podstawie dostarczonych danych, sterownik sam wylicza potrzebne przesunięcia, aby wykonać daną grupę ruchów. Do obsłużenia dowolnego rozmieszczenia pozycji w przestrzeni roboczej można wybrać opcję List, gdzie programista wprowadza listę wszystkich pozycji. Zmienna licznika jest używana podczas przechodzenia między pozycjami wzoru. Nazwa zmiennej jest wyświetlana na ekranie polecenia Pattern. Zmienna przyjmuje kolejno wartości od 0 do  $X * Y * Z - 1$ , czyli liczby punktów we wzorze. Tą zmienną można manipulować za pomocą przypisania i można jej używać w wyrażeniach.

**Uwaga:** W celu wykonania zadania paletyzacji warto posłużyć się kreatorem Pallet z zakładki Wizards (Rys. 9).



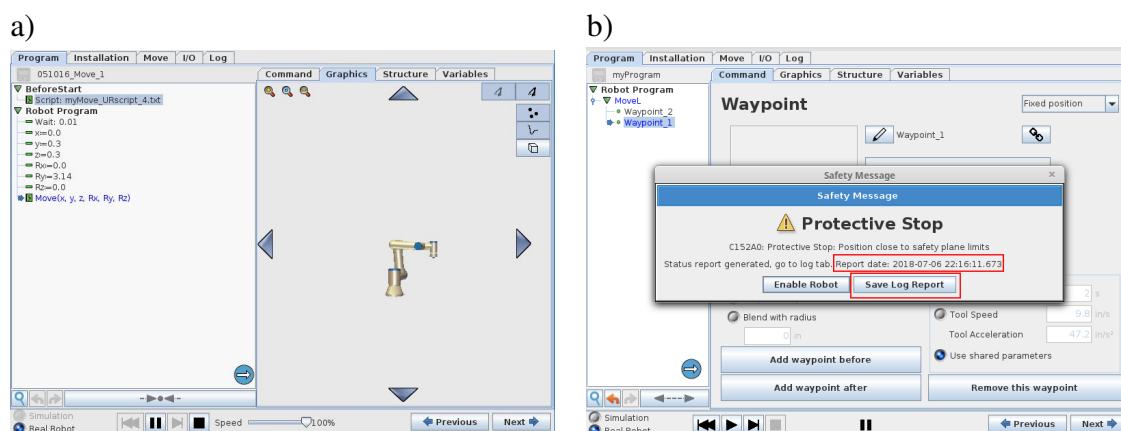
Rysunek 9: Kreator zadania paletyzacji Pallet

### 3.5 Uruchamianie programu

Aby uruchomić program należy stanąć w bezpiecznej odległości od robota i w gotowości do naciśnięcia przycisku zatrzymania awaryjnego nacisnąć przycisk Play (Rys. 10a). Wykonywanie programu nastąpi natychmiast, pod warunkiem, że pozycja bieżąca robota jest pozycją startową, czyli zaznaczoną przez nas pozycją w programie, od której będziemy chcieli rozpocząć odtwarzanie programu. Jeśli tak nie jest, program poprosi o ustawienie pozycji początkowej. Wówczas należy wcisnąć przycisk auto i przytrzymać go aż do uzyskania pozycji startowej programu, dalej wcisnąć OK i wreszcie jeszcze raz wcisnąć Play w celu wykonania programu. Pozostałe przyciski pozwalają zatrzymać program, wykonać go krok po kroku lub ponownie uruchomić. Suwak prędkości umożliwia regulację prędkości programu w dowolnej chwili, w czasie rzeczywistym. Z lewej strony znajdują się przyciski Simulation i Real Robot, które przełączają uruchomienie programu między pracą w trybie symulacji i pracą na prawdzi-



wym robocie. Podczas pracy w trybie symulacji ramię robota się nie porusza, dzięki czemu nie może ulec uszkodzeniu ani uszkodzić pobliskiego wyposażenia w efekcie kolizji. **Uwaga: ZAWSZE przed uruchomieniem programu na robocie należy uruchomić program w trybie symulacji i upewnić się, że wykonywany program jest poprawny.**



Rysunek 10: Uruchamianie programu

Jeśli robot napotka na przeszkodę lub będzie musiał użyć zbyt dużej siły w celu realizacji ruchu, pojawi się komunikat `Protective Stop` (patrz Rys. 10b). To będzie oznaczało, że aby umożliwić dalsze nawigowanie robotem, trzeba będzie usunąć wszelkie przeszkody znajdujące się w zasięgu ramienia i wcisnąć przycisk `Enable Robot`.

W celu natychmiastowego zatrzymania robota, należy użyć przycisku awaryjnego zatrzymania na panelu programowania. Chcąc ponownie uruchomić program, należy wyeliminować przyczynę awaryjnego zatrzymania (na przykład, poprawić program) i obrócić przycisk awaryjnego zatrzymania zgodnie z ruchem wskazówek zegara, w celu odblokowania.

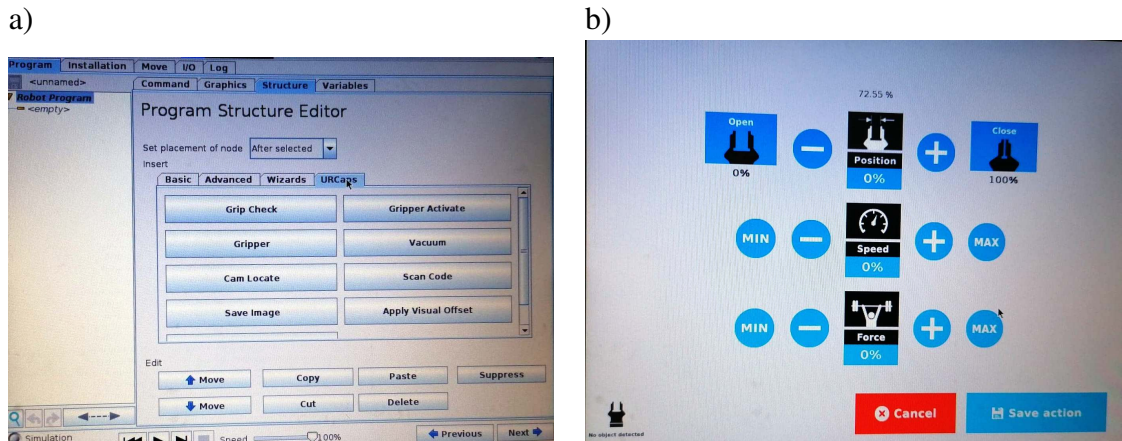
### 3.6 Obsługa zewnętrznych sygnałów oraz chwytaka

Poniższy podrozdział pokazuje jak zaprogramować oczekiwanie na sygnały z czujników/urządzeń zewnętrznych oraz ich obsługę. W tym celu wykorzystujemy polecenia `Wait` oraz `Set` z zakładki `Structure/Basic`. Polecenie `Wait` umożliwia zaprogramowanie oczekiwania na konkretne wydarzenie, natomiast polecenie `Set` pozwala zaprogramować cyfrowe/analogowe wejścia/wyjścia. W zakładce `Command` dotyczącej polecenia `Wait` należy określić na jaki sygnał oczekujemy (cyfrowy/analogowy), z jakiego urządzenia/czujnika powinien się pojawić sygnał (rozwijane menu obok typu sygnału), na jaką wartość sygnału oczekujemy (`Low/High`) oraz ile ma wynosić czas oczekiwania.

Chcąc użyć chwytak należy wybrać zakładkę `Structure/URCaps` (Rys. 11). W celu aktywowania chwytaka wybieramy polecenie `Gripper Active` i dalej przycisk `Gripper`. W zakładce `Command` wybieramy polecenie `Edit action` i ustalamy szczegóły dotyczące działania chwytaka, w szczególności decydujemy tu z jaką prędkością i siłą chwytak ma być zamknięty/otwarty.

### 3.7 Definiowanie układów współrzędnych: USER FRAME

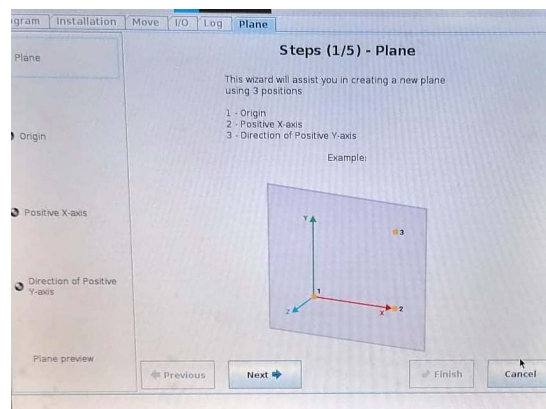
Dobłą praktyką jest określenie na początku programu w jakim układzie współrzędnych porusza się robot. Dotyczy to zarówno definicji układu współrzędnych stowarzyszonego z narzędziem (`TOOL`) jak i układu współrzędnych użytkownika związanego np. ze stołem roboczym



Rysunek 11: Programowanie chwytaka: a) zakładka Structure/URCaps; b) ustawienia chwytaka

(USER FRAME). Jawne określenie używanych układów współrzędnych (ang. features) na początku programu zwiększa bezpieczeństwo oraz niezawodność działania robota. Uzyskujemy pewność, że punkty zapamiętane w pewnym układzie współrzędnych będą wykonywane w tym układzie.

W celu zdefiniowania układu współrzędnych użytkownika (USER FRAME) wybieramy zakładkę Installation, dalej Features, wreszcie przycisk Plane u dołu po prawej stronie ekranu (to oznacza, że nasz układ współrzędnych zdefiniujemy za pomocą płaszczyzny o nazwie Plane\_1;  $i$ -te układy współrzędnych otrzymują nazwy Plane\_1). Jesteśmy przeprowadzani przez całą procedurę definiowania układu współrzędnych użytkownika (patrz Rys. 12). W szczególności, płaszczyznę definiujemy za pomocą trzech punktów. Punkt pierwszy definiuje środek układu współrzędnych, punkt drugi definiuje kierunek osi  $X$ , natomiast punkt trzeci zamyka płaszczyznę  $XY$ . A zatem w kolejnych krokach będziemy zadawać kolejne punkty płaszczyzny. Punkty zadajemy poprzez kolejno: wybór przycisku Change Position, umieszczenie robota w odpowiedniej pozycji i zatwierdzenie pozycji przyciskiem OK. Płaszczyznę definiującą układ współrzędnych użytkownika możemy w każdej chwili zmodyfikować. Należy wówczas w zakładce Installation wybrać interesującą nas płaszczyznę oraz wcisnąć przycisk Modify Plane. Dzięki temu jeszcze raz zostaniemy przeprowadzeni przez procedurę zadawania kolejnych punktów płaszczyzny.



Rysunek 12: Definiowanie układu współrzędnych użytkownika

W odniesieniu do tak zdefiniowanych układów współrzędnych możemy używać wszystkich

poleceń ruchu. A zatem w celu zdefiniowania kolejnego punktu trasy realizowanej w ramach zdefiniowanego układu współrzędnych należy w zakładce `Command` dotyczącej kolejno zadawanych punktów orientacyjnych (`Waypoint`) wybrać polecenie `Set Waypoint` i dalej w rozwijanym menu `Feature` wybrać nazwę uprzednio zdefiniowanego układu współrzędnych (np. `Plane_1`).

W celu przeniesienia naszego układu współrzędnych w nowe miejsce należy przeddefiniować punkty definiujące wyznaczoną przez nas uprzednio płaszczyznę (omawiany wcześniej przycisk `Modify Plane`).

## 4 Zadania do wykonania

1. Zapoznaj się z trybem pracy ręcznej. W szczególności potrenuj ręczne zadawanie położenia i orientacji końca efektora w przestrzeni zadaniowej oraz konfiguracji w przestrzeni przegubowej robota (zakładka `Move`).
2. W trybie pracy ręcznej zaobserwuj jak zachowuje się robot w pobliżu granic zdefiniowanej wcześniej strefy bezpieczeństwa.
3. Zastanów się kiedy/w jakich sytuacjach stosujemy różne typy ruchu (`MoveL`, `MoveJ`, `MoveP`).
4. Zdefiniuj układ współrzędnych użytkownika (rozdział 3.7).
5. Zaprogramuj robota tak, by wykonywał ruch:
  - po okręgu (polecenie ruchu `MoveP`);
  - wzdłuż kolejnych krawędzi kwadratu (najpierw za pomocą zadawania kolejnych punktów orientacyjnych (tzw. `Fixed Position`), a później z wykorzystaniem zadawania względnych punktów orientacyjnych (rozdział 3.3)).
6. Zrealizuj zadanie wkręcania śruby do podstawki umieszczonej na blacie (zadanie z wykorzystaniem chwytaka).
7. Zrealizuj przykładowe zadanie paletyzacji (polecenie `Structure/Wizards/Pallet`, w drzewie programu wybierz `Pattern` oraz zakładkę `Command`, z której zostanie przeprowadzony przez kolejne kroki procedury definiowania zadania paletyzacji).

## 5 Przydatne materiały

1. Universal Robots UR3 User Manual,
2. Materiały szkoleniowe Universal Robots  
[www.universal-robots.com/pl/akademia/](http://www.universal-robots.com/pl/akademia/)