Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

Wydział Elektryczny

Sterowanie Złożonymi Układami Mechanicznymi Sprawozdanie z projektu		
Projekt 1	Temat: Sterowanie manipulatorem na satelicie	
Kierunek studiów: AIR	Studia: S1	Rok: 4
Data oddania projektu: 9.12.2020r.	Prowadzący: mgr. inż. Adam Łukomski	Wykonał: Piotr Brandebura nr albumu 38618 Kamil Przybył nr albumu 38671

1. Cel projektu

Celem projektu jest opracowanie sterownia manipulatorem na satelicie łapiącego przelatujące przedmioty. Wykorzystując wiedzę zdobytą na zajęciach obliczyć kinematykę manipulatora oraz napisać sterowanie jakobianowe ale wybranego urządzenia.

2. Wykorzystane oprogramowanie

Do zrealizowania projektu wykorzystaliśmy oprogramowanie VMware Workstation, Ubuntu 18.04, Matlab oraz Gazebo. Maszyna wirtualna VMware pozwoliła nam na uruchomienie systemu operacyjnego Ubuntu w wersji 18.04. Środowisko ROS oraz symulator Gazebo pozwolił nam na przygotowanie wybranego manipulatora a środowisko Matlab posłużyło do obliczenia kinematyki i jakobianu manipulatora jak i jego sterowania.

3. Wybór manipulatora i specyfikacja

Na serwisie Github dostępnych jest wiele rodzajów manipulatorów do uruchomienia na platformie Gazebo. Niestety nie każdy projekt urządzenia dla tego środowiska jest w stanie wystawiać potrzebne dane na zewnątrz, które potrzebujemy wykorzystać w Matlabie. Istniej możliwość stworzenia własnego robota, lecz postanowiliśmy użyć gotowy przykład. Wybór padł na projekt OpenMANIPULATOR-X. Jest to robot szeregowy posiadający 4 stopnie swobody oraz chwytak szczękowy, łączenie 5 stopni.

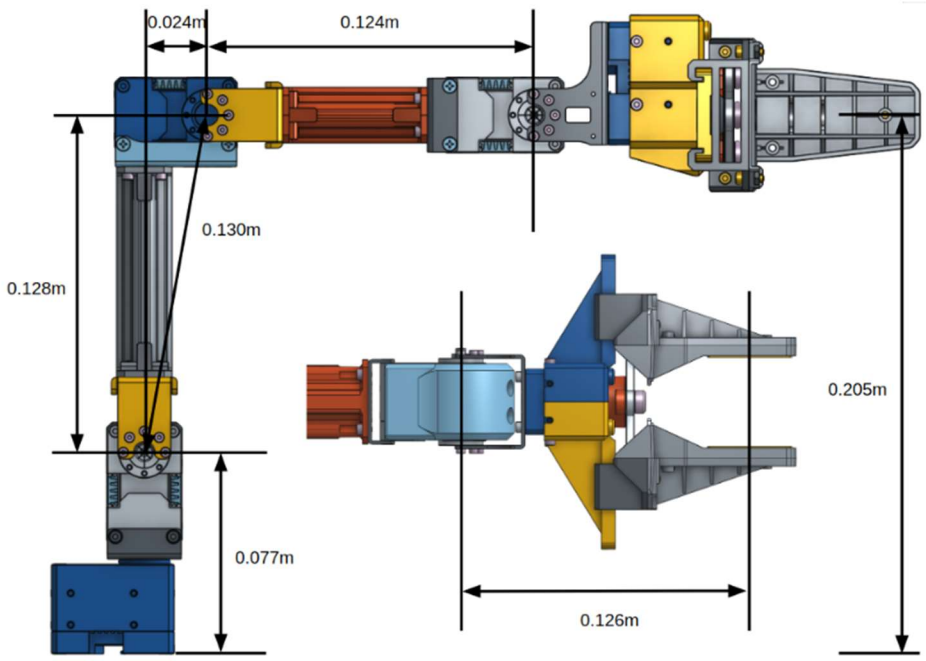
Specyfikacja manipulatora.

Wybrany manipulator to model fizycznego robota dostępnego na rynku. Jego specyfikacje prezentuje tabela 1:

Tab.1

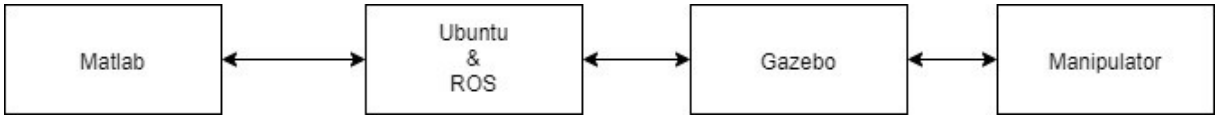
Opis	Jednostka	OpenManipulator-X
Napęd	-	DYNAMIXEL XM430-W350-T
Napięcie zasilania	V	12
DOF	-	5 (4DOF + 1DOF chwytak)
Ładowność	g	500
Powtarzalność	mm	<0.2
Prędkość (speed jonit)	RPM	46
Waga	kg	0.70
Obszar roboczy	mm	380
Szerokość pracy chwytaka	mm	20~75
Oprogramowanie	-	ROS, DYNAMIXEL SDK, Arduino
Główny sterownik	-	PC, OpenCR

Wymiary:



4. Schemat blokowy systemu.

W punkcie 2. zostało wymienione wykorzystanego oprogramowanie. Poniżej przestawione jest schemat wymiany danych między poszczególnymi programami.



OpenMANIPULATOR-X otwarty w programie Gazebo przekazuje informacje do Matlaba na temat swojej pozycji i kątów. Na podstawie tego wyznaczana jest kinematyka i jakobian.

5. Kinematyka manipulatora

Kinematyką to nauka zajmująca się badaniem ruchu pomijając siły wywołujące ten ruch. Analizowane są zmiany położenia prędkości oraz przyspieszeń dla każdego członu czyli każdego punktu w danym członie, szczególnie chwytaka.

Proste zadanie kinematyki manipulatora to obliczanie pozycji i orientacji członu roboczego robota. Mając informacje o współrzędnych konfiguracyjnych można obliczyć jego pozycje punktu związanego z robotem względem globalnego układu współrzędnych. W skrócie jest to opis położenia manipulatora w przestrzeni współrzędnych kartezjańskich.

Aby obliczyć kinematykę robota działamy w poniższych krokach:

1. Wyznaczamy wektor p1, który określa położenie bazy układu (pierwszego przegubu) w układzie XYZ. Wektor p1, dla ogólnego przypadku, przyjmuje postać:

$$P_1 = \begin{bmatrix} P_X \\ p_y \\ p_z \end{bmatrix}$$

gdzie:
Px – składowa wektora położenia wzdłuż osi x
Py – składowa wektora położenia wzdłuż osi y
Pz – składowa wektora położenia wzdłuż osi z
Najczęściej przyjmujemy, że ten punkt leży na początku układu współrzędnych

2. Wyznaczamy wektor r10, który określa położenie drugiego przegubu względem bazy w stanie początkowym. Robimy to poprzez dodanie do wektora położenia bazowego (p1) wektora długości pierwszego członu wzdłuż wszystkich osi (l1). Dla członu pierwszego można to zapisać następująco:

$$r_{10} = p_1 + l_1 = \begin{bmatrix} p_{1x} + l_{1x} \\ p1y + l_{1y} \\ p_{1z} + l_{1z} \end{bmatrix}$$

3. Określenie wektora prędkości obrotowych przegubu pierwszego ω1 wzdłuż wszystkich osi. W postaci ogólnej wektor ω (dla danego przegubu) wygląda następująco:

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

gdzie:
ω_x – prędkość obrotowa przegubu wokół osi x
ω_y – prędkość obrotowa przegubu wokół osi y
ω_z – prędkość obrotowa przegubu wokół osi z

4. Wyznaczamy wektor prędkości liniowej v1, który jest iloczynem wektorowym prędkości obrotowej pierwszego przegubu i położenia przegubu pierwszego. W postaci ogólnej prędkość liniowa jest równa:

$$v_n = -\omega_n \times p_n$$

gdzie:
ω_n – wektor prędkości obrotowej n-tego przegubu
p_n – wektor położenia n-tego przegubu

5. Wyznaczenie macierzy s_1, która w postaci ogólnej wygląda tak:

$$s_n = \begin{bmatrix} w\hat{}_n & v_n \\ 0 & 0 \end{bmatrix}$$

Macierz ta składa się z wektora prędkości liniowej v_n oraz macierzy prędkości obrotowej ω z daszkiem. Już wiemy jak wyznaczyć prędkość liniową, ale nie znamy postaci ω z daszkiem. Ta macierz, w postaci ogólnej, wygląda następująco:

$$\hat{w_n} = \begin{bmatrix} 0 & -\omega_{nz} & \omega_{ny} \\ \omega_{nz} & 0 & -\omega_{nx} \\ -\omega_{ny} & \omega_{nx} & 0 \end{bmatrix}$$

gdzie:
ω_{nx} – prędkość obrotowa n-tego przegubu wokół osi x
ω_{ny} – prędkość obrotowa n-tego przegubu wokół osi y
ω_{nz} – prędkość obrotowa n-tego przegubu wokół osi z

6.Wyznaczenie równania opisującego położenie końcówki pierwszego członu, które przyjmuje postać:

$$r_1(\theta) = e^{s_1 \theta_1} \cdot r_{10}$$

Obliczając pochodną równia r1 możemy wyznać prędkość tego punktu:

$$\dot{r_1} = \hat{s_1} . \theta_1 . e^{\hat{s_1} \hat{\theta_1}} \cdot r_{10}$$

6. Jakobian manipulatora

Definicja Macierzy Jakobiego

$$\mathbf{J}^{a}(\mathbf{x}) = \frac{\partial \mathbf{k}}{\partial \mathbf{x}}(\mathbf{x}),$$

Po zróżniczkowaniu względem czasu równania opisujące jego kinematykę manipulatora, zauważamy, że jakobian analityczny opisuje transformację, prędkości zmian współrzędnych przegubowych w prędkości zmian współrzędnych zadaniowych.

$$\dot{p} = J\dot{q}$$

Po przekształceniu:

$$\dot{q} = J^{-1}\dot{p}$$

Powyższe wyprowadzenie pozwala na zaprojektowanie sterowania manipulatorem opartego o jakobian manipulatora.

7. Otwieranie OpenMANIPULATOR-X w Gazebo

Aby uruchomić wybrany manipulator należy postępować zgodnie z poniższą instrukcją.

1. Aktualizujemy oprogramowanie

```
$ sudo apt update
```

2. Instalujemy platformę GIT

```
$ sudo apt install git
```

3. Instalujemy biblioterkę ROS

```
$ sudo apt-get install ros-kinetic-ros-controllers ros-kinetic-gazebo* ros-kinetic-moveit* ros-kinetic-industrial-core
```

4.Tworzymy katalog catkin_ws

```
$ mkdir catkin_ws
```

5. Przechodzimy do catkin_ws i zakładamy w nim katalog src

```
$ mkdir src
```

6. W katalogu src dodajemy pliki robota OpenMANIPULATOR X z Github

```
$ git clone https://github.com/ROBOTIS-GIT/DynamixelSDK.git
$ git clone https://github.com/ROBOTIS-GIT/dynamixel-workbench.git
$ git clone https://github.com/ROBOTIS-GIT/dynamixel-workbench-msgs.git
$ git clone https://github.com/ROBOTIS-GIT/open_manipulator.git
$ git clone https://github.com/ROBOTIS-GIT/open_manipulator_msgs.git
$ git clone https://github.com/ROBOTIS-GIT/open_manipulator_simulations.git
$ git clone https://github.com/ROBOTIS-GIT/robotis_manipulator.git
```

7. Wychodzimy z katalogu src i kompilujemy wgrane pliki

```
$ cd ..
$ catkin_make
```

8.Tworzymy odwołanie

```
$ source devel/setup.bash
```

9. Uruchomiamy manipulator w symulatorze Gazebo:

```
$ roslaunch open_manipulator_gazebo open_manipulator_gazebo.launch
```

Przy jeżeli nie uruchamia się program gazebo należy uruchomić platformę ros

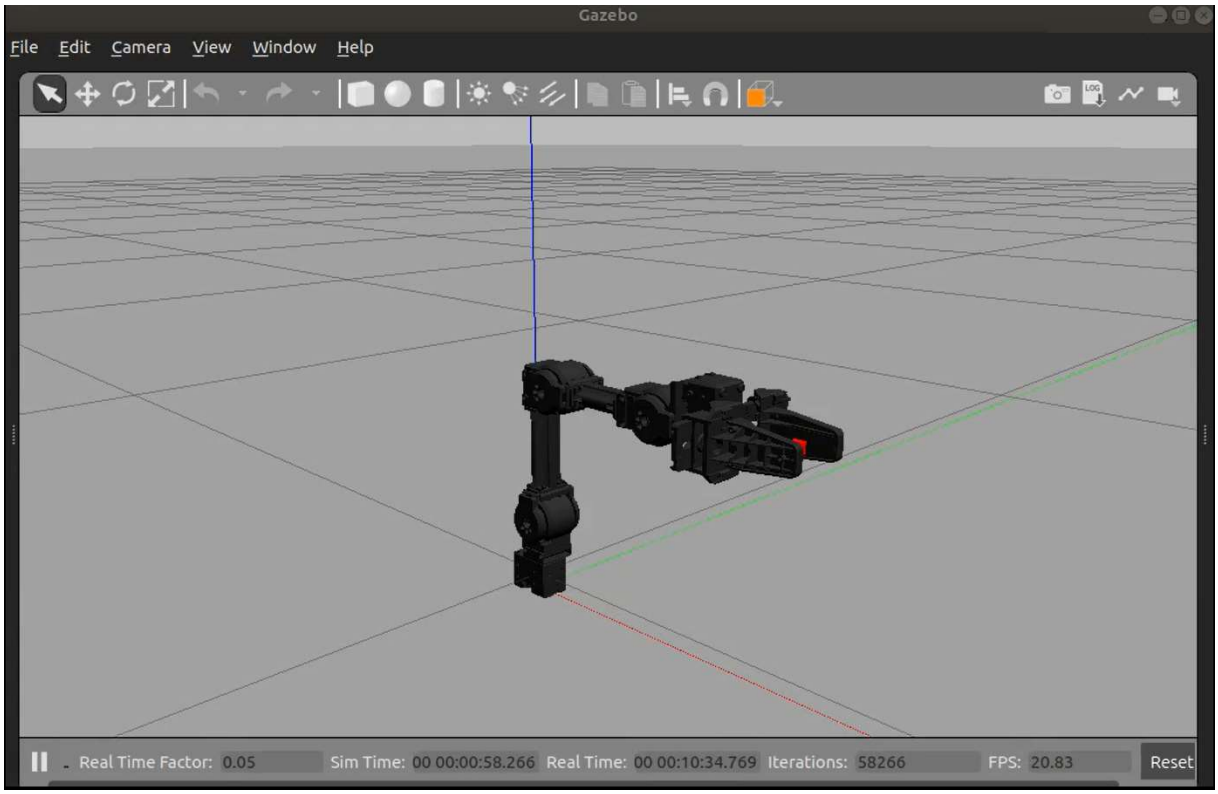
```
$ rosrun
```

Komenda ta pozwala na uruchomienie pliku wykonywalnego w dowolnym pakiecie z dowolnego miejsca bez konieczności podawania pełnej ścieżki lub cd/roscd. Po wpisaniu tej komendy pojawia się komunikat:

```
Usage: rosrun [--prefix cmd] [--debug] PACKAGE EXECUTABLE [ARGS]
rosrun will locate PACKAGE and try to find
an executable named EXECUTABLE in the PACKAGE tree.
If it finds it, it will run it with ARGS.
```

Jeżeli nie, należy zainstalować ROS:

```
$ sudo apt install ros-melodic-desktop-full
```



8. Połączenie Matlab z symulacją Gazebo

Po otwarciu programu Matlab w Comand window wpisujemy komendę rosinit. Polecenie to uruchamia globalny węzeł ROS z Matlab i próbuje połączyć się urządzeniem główny ROS uruchomiony na hoście lokalnym i porcie 11311.

Komenda rostopic list pozwala na zwrócenie do Matlaba informacji oraz możliwego sterowania dla robota. OpenManipulator-X zwraca następujące pozycje:

```
/clock
/gazebo/link_states
/gazebo/model_states
/gazebo/set_link_state
/gazebo/set_model_state
/open_manipulator/gripper/kinematics_pose
/open_manipulator/gripper_position/command
/open_manipulator/gripper_sub_position/command
```

```
/open_manipulator/joint1_position/command
/open_manipulator/joint2_position/command
/open_manipulator/joint3_position/command
/open_manipulator/joint4_position/command
/open_manipulator/joint_states
/open_manipulator/option
/open_manipulator/states
/rosout
/rosout_agg
```

9. Zakończenie

Projekt nie został wykonany w pełni. Udało się obliczyć kinematykę wybranego manipulatora oraz jego jakobian, lecz sterowanie obywa się za pośrednictwem wysyłania komend z matalba do gazebo. Napotkane problemy z wyciąganiem zmiennych z macierzy jakobina nie pozwoliła na wykonanie zdania w pełni, ale mimo to robot daje się sterować.

Link do filmu z działania systemu: https://www.youtube.com/watch?v=u-Gnp_Lg5N4&feature=youtu.be&fbclid=IwAR13DTZzwPle7VAlUSVMPYPvri83CqD36cPzevv7zXRsynB4b0udlZ_PLPU