

Komunikacja i sterowanie dronem za pośrednictwem MATLABa

Jakub Szczygiał



Zakres podstawowy

- Implementacja modułu dla środowiska MATLAB pozwalającego na komunikację z dronem za pośrednictwem ekosystemu ROS i biblioteki bebop_autonomy.
- Odczyt kompletu danych z sensoryki pokładowej, pobieranie danych z kamery pokładowej w czasie rzeczywistym, odczyt stanu z wewnętrznej maszyny stanu.
- Wydawanie komend z poziomu modułu.

Platforma - Parrot Bebop 2

- komunikacja przez Wi-Fi
- bateria 2700 mAh
- czas pracy 25 min
- kamera 14 Mpix
- wideo 1080p, do 60 fps
- akcelerometr, barometr, czujnik ultradźwiękowy, magnetometr, wysokościomierz, żyroskop



Biblioteka bebot_autonomy

Komendy:

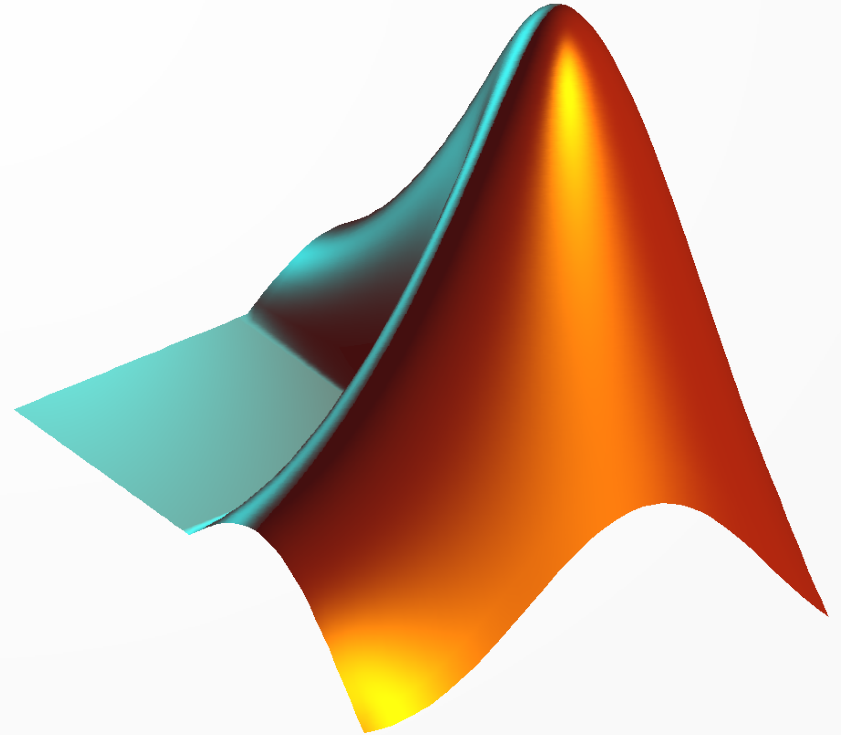
- Start/Lądowanie/Awaria (takeoff/land/reset, std_msgs/Empty)
- Sterowanie (cmd_vel, geometry_msgs/Twist)
- Ruch wirtualną kamerą (camera_control, geometry_msgs/Twist)

Odczytywanie danych:

- Obraz z kamery (image_raw, sensor_msgs/Image)
- Odometria (odom, nav_msgs/Odometry)
- Stan drona (states/*)

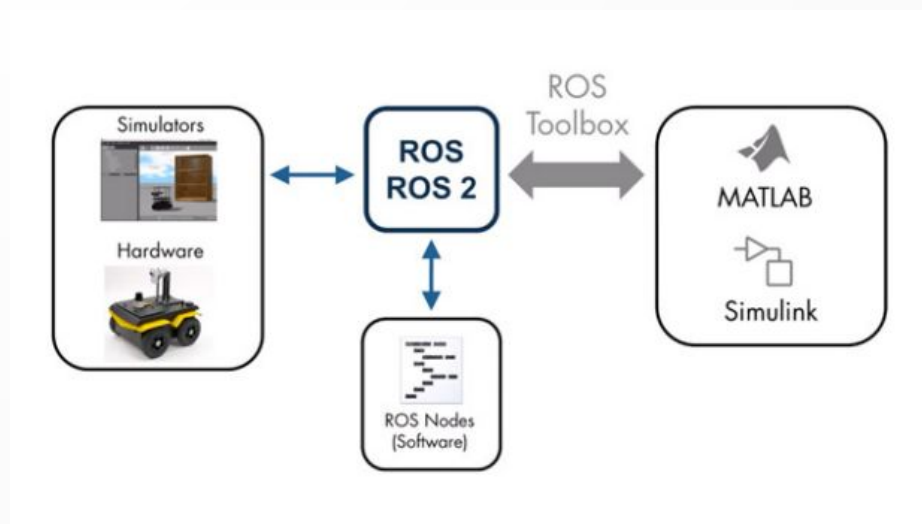
MATLAB

- Stworzony do operacji na macierzach i wektorach
- Możliwość pisania skryptów i funkcji
- Operacje na obrazach
- Tworzenie wykresów dwu- i trójwymiarowych



ROS Toolbox

- Pozwala na połączenie MATLABA i Simulinka do ROSa
- Działa zarówno z działającym systemem, jak i wiadomościami z rosbaga
- Może wysyłać i odczytywać wiadomości



Plan działania

- Uruchomić maszynę wirtualną z biblioteką bebop_autonomy
- Przetestować współpracę MATLABa i ROSa
- Napisać funkcje w MATLABie do podstawowego sterowania dronem i odczytywania danych korzystając z możliwości bebop_autonomy
- Zaimplementować bardziej zaawansowane metody sterowania

Pomysły

- Wykonanie i wczytanie zdjęcia do MATLABa do dalszego przetwarzania
- Wczytywanie komend ruchowych z macierzy
- Zadanie prostej trajektorii podanej w macierzy i wykonanie jej na podstawie odometrii
- Wczytanie wykonanej trajektorii z odometrii i wyświetlenie jej na wykresie 3D

Dziękuję za uwagę

