

## Temat: Badanie odometrii

W ramach tego zadania w folderze /scripts zostały stworzone dwa skrypty kinematic.py i kinematic2.py oraz skrypt pomocniczy corrections.py do celu obróbki danych prezentowanych w sprawozdaniu. Skrypt kinematic.py implementuje węzeł sterujący robotem bazujący na domniemanej (idealnie) zrealizowanej poprzedniej akcji ruchu po kwadracie. Skrypt kinematic2.py implementuje sterowanie robota na bazie odometrii (pozyskanych informacji o jego położeniu i orientacji na podstawie pomiarów z enkoderów w kołach).

Do badania zostały przyjęte prędkości podane przez prowadzącego:

liniowa = 0.11

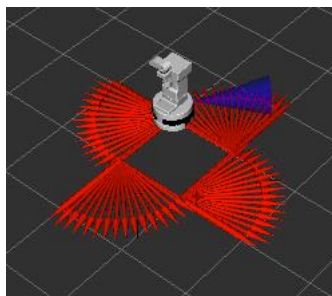
kątowa = 0.1

Poniżej zostały zaprezentowane wygenerowana przez robota trajektorie i przebiegi kolejno położenia  $x$  w funkcji czasu, orientacji (kąt  $\theta$ ) w funkcji czasu, prędkości liniowej i kątowej w funkcji czasu oraz błędu chwilowego w funkcji czasu.

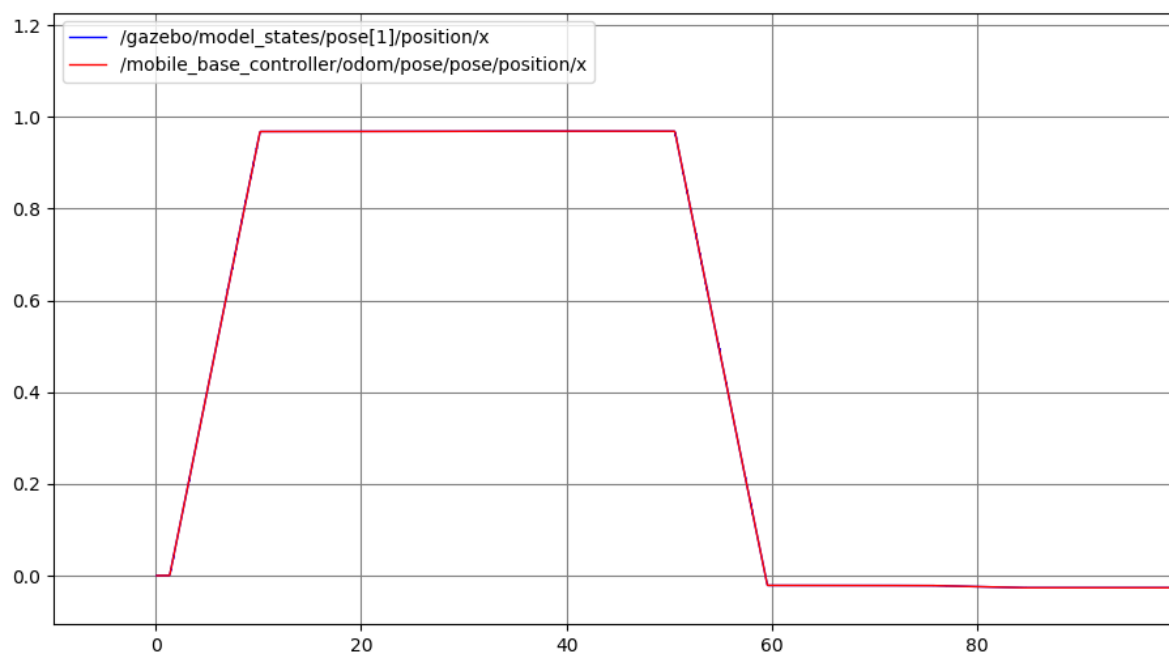
**Badanie implementacji bazującej na idealnie zrealizowanej poprzedniej akcji**

Badanie to przeprowadzono dla trzech wariantów prędkości:

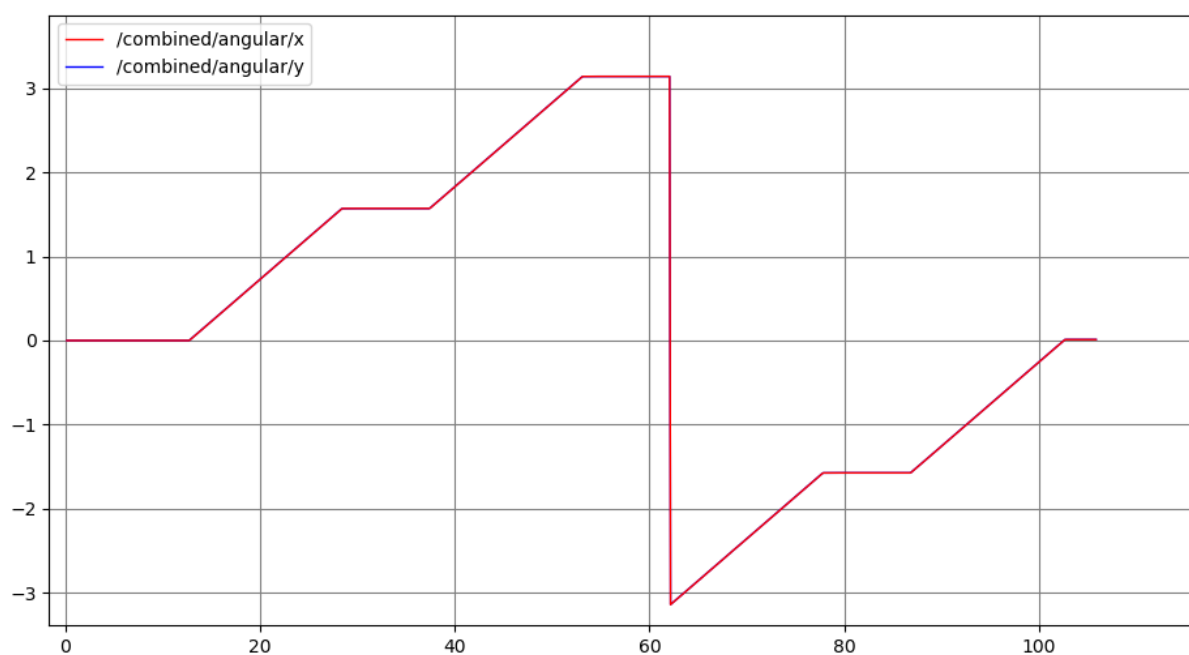
- |   |                                    |
|---|------------------------------------|
| 1. liniowa = 0.11                       | kątowa = 0.1                       |
| 2. liniowa = $0.11 \cdot 2.4$           | kątowa = $0.1 \cdot 2.4$           |
| 3. liniowa = $0.11 \cdot 2.4 \cdot 2.4$ | kątowa = $0.1 \cdot 2.4 \cdot 2.4$ |

**Wariant 1.**

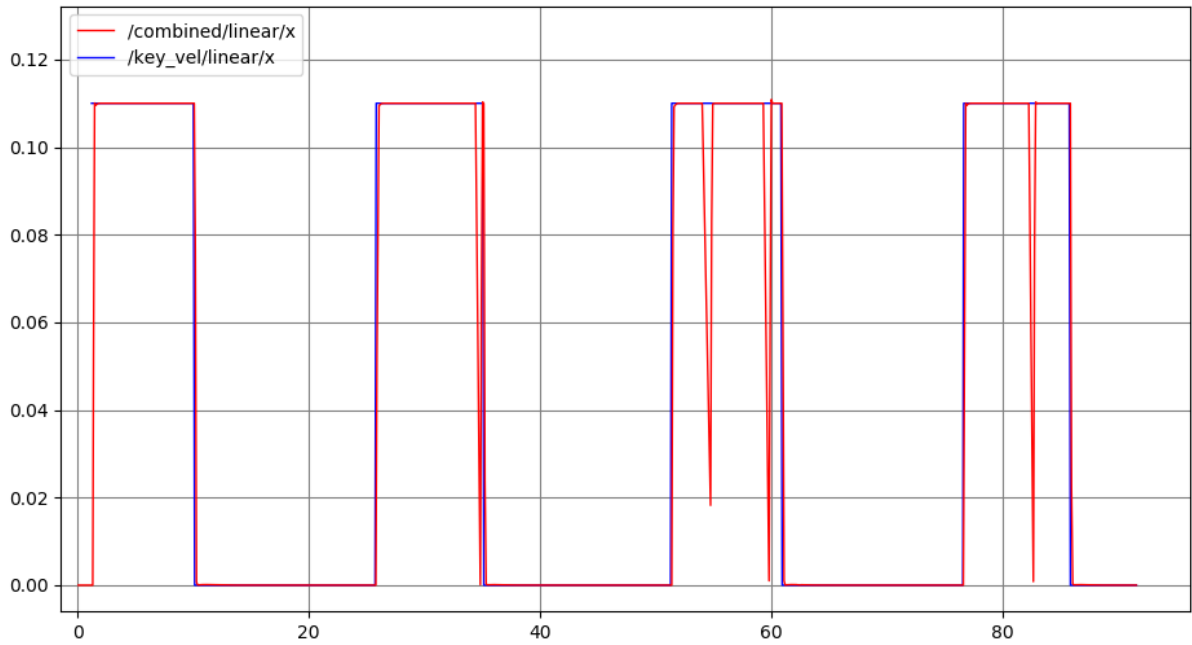
Rys. 1 Wygenerowana trajektoria przez robota



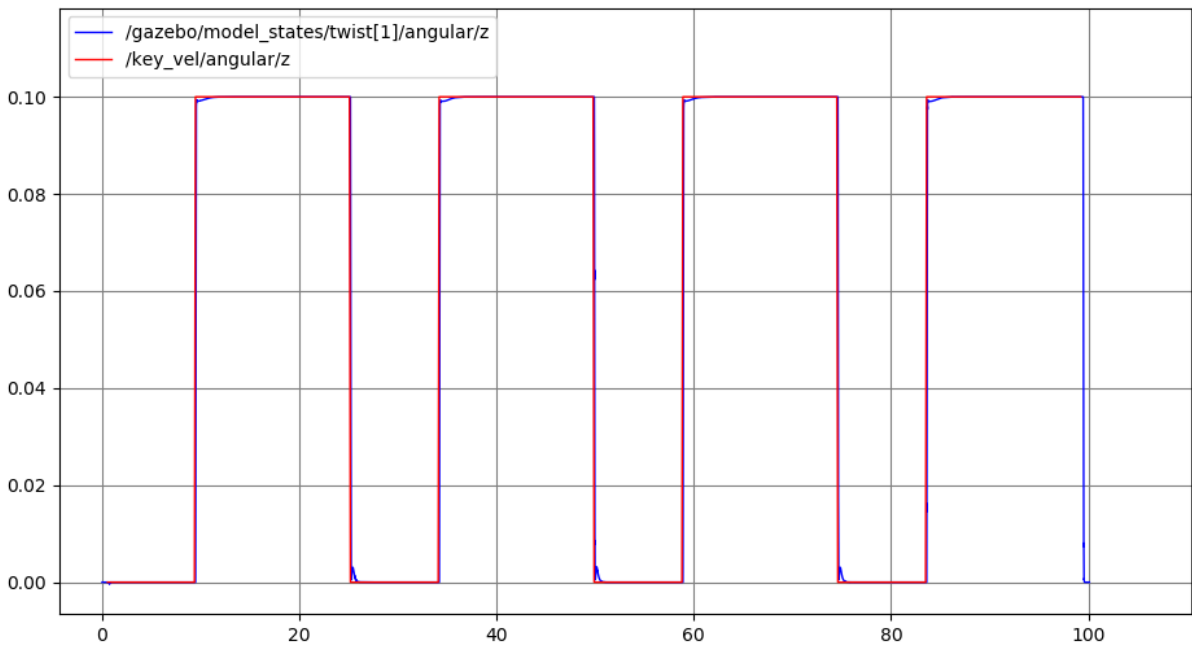
Rys. 2 Zależność położenia odczytanego z odometrii z położeniem referencyjnym w funkcji czasu



Rys 3. Zależność położenia kąтового odczytanego z odometrii z położeniem referencyjnym w funkcji czasu (y – pomiar referencyjny)

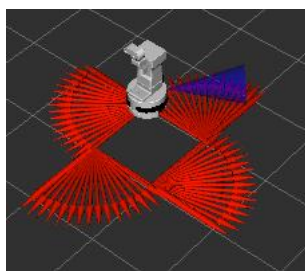


Rys. 4 Przebieg zadawanej prędkości liniowej z referencyjną w funkcji czasu

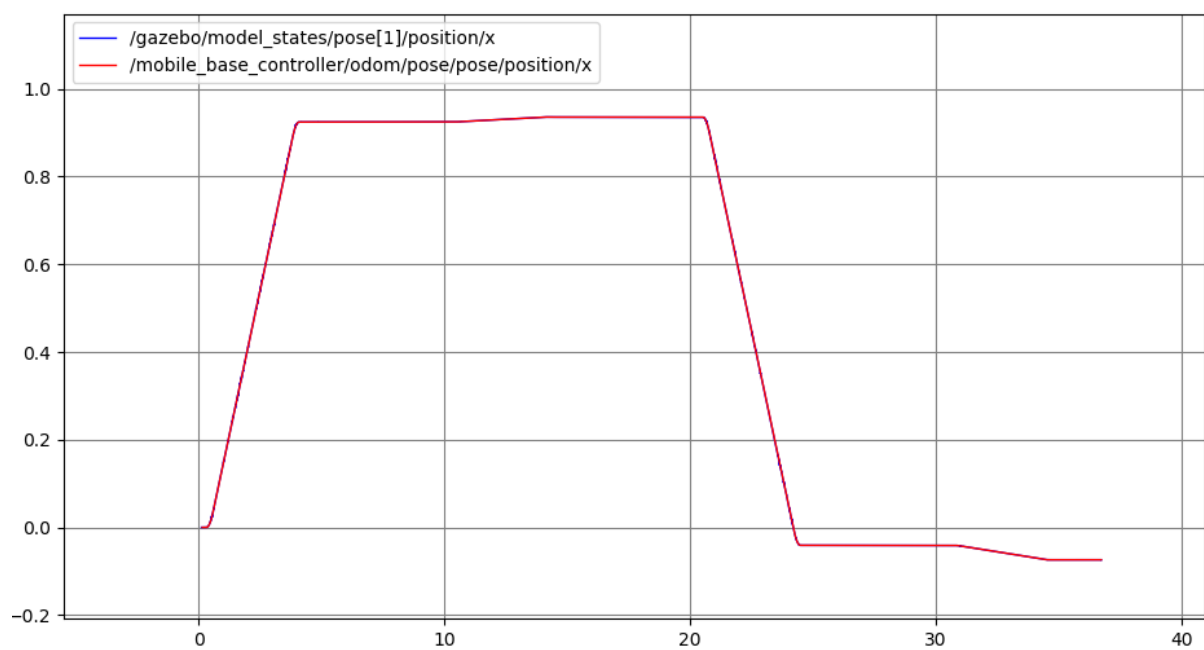


Rys. 5 Przebieg zadawanej prędkości kątowej z referencyjną w funkcji czasu

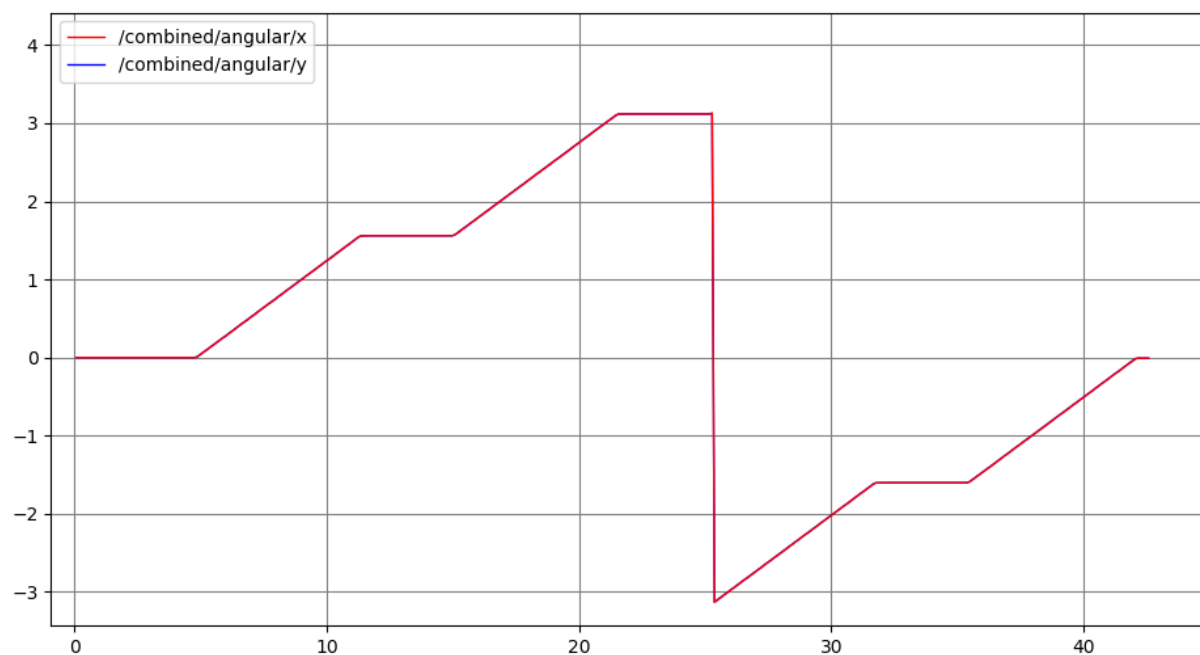
## Wariant 2.



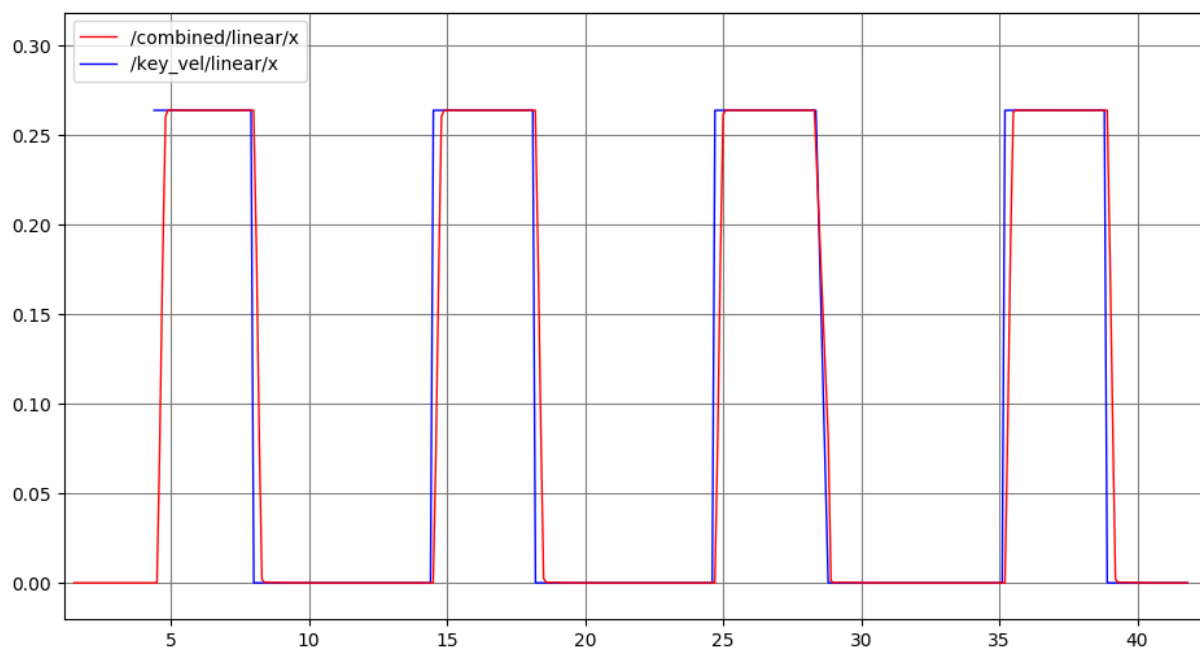
Rys. 6 Wygenerowana trajektoria przez robota



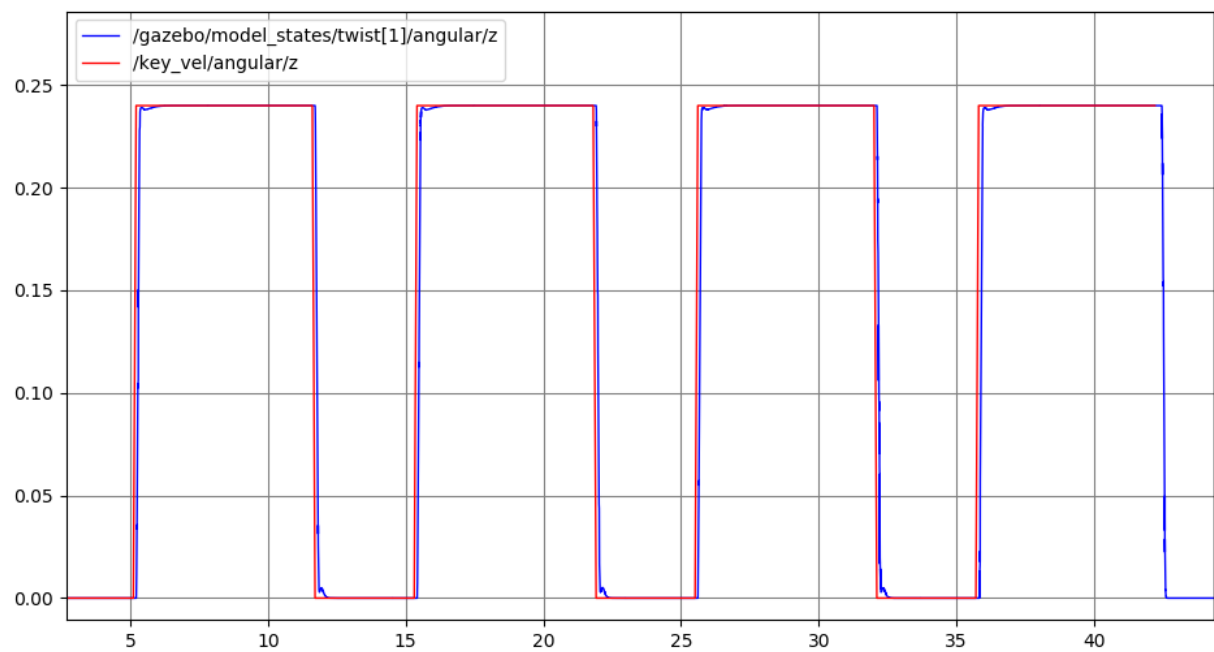
Rys. 7 Zależność położenia odczytanego z odometrii z położeniem referencyjnym w funkcji czasu



Rys. 8 Zależność położenia kąтового odczytanego z odometrii z położeniem referencyjnym w funkcji czasu (y – pomiar referencyjny)

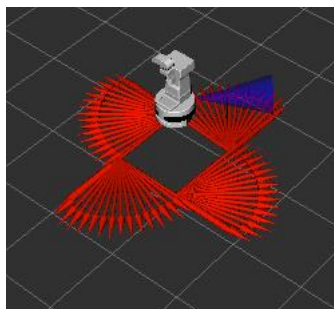


Rys. 9 Przebieg zadawanej prędkości liniowej z referencyjną w funkcji czasu

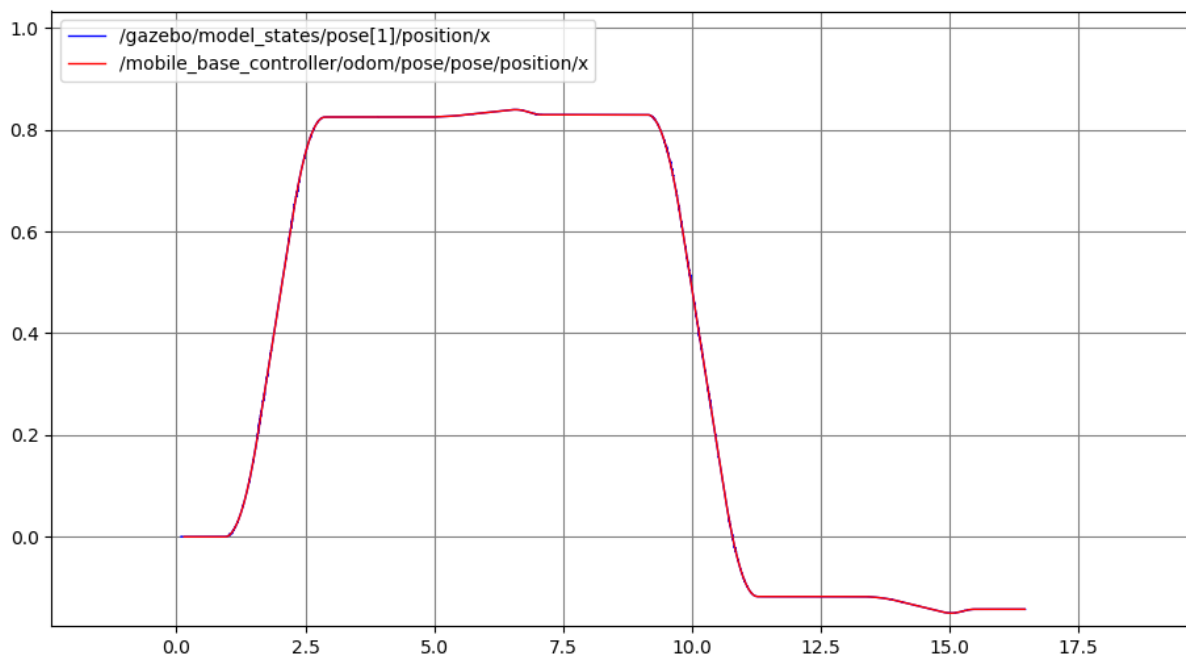


Rys. 10 Przebieg zadawanej prędkości kątowej z referencyjną w funkcji czasu

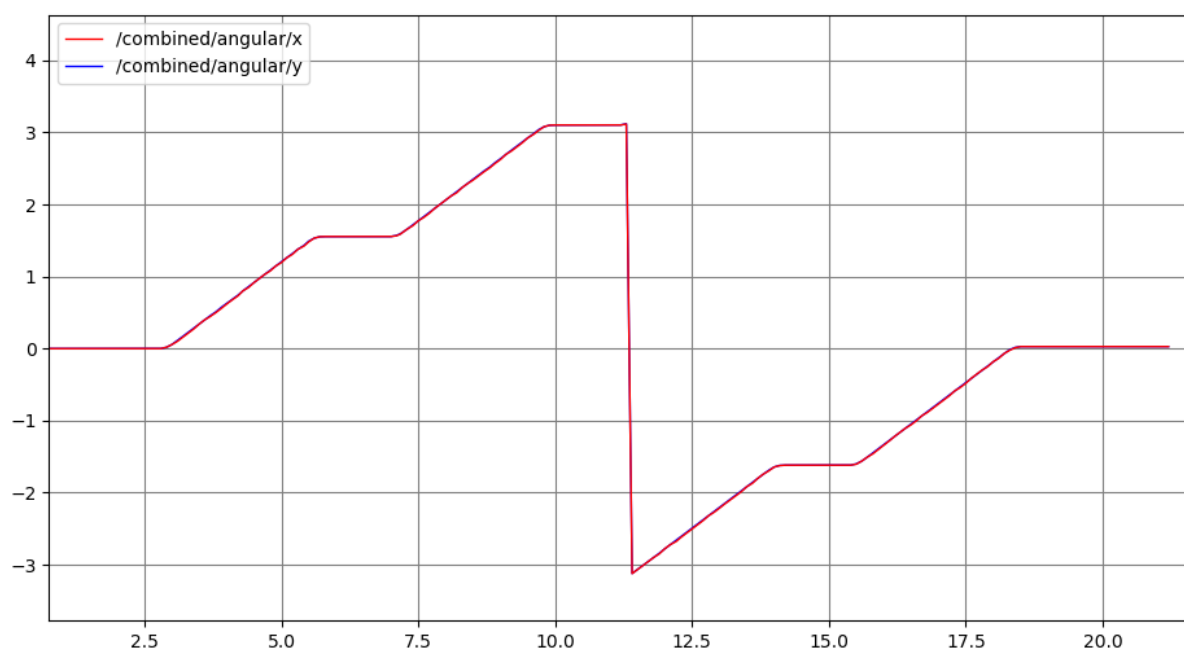
### Wariant 3.



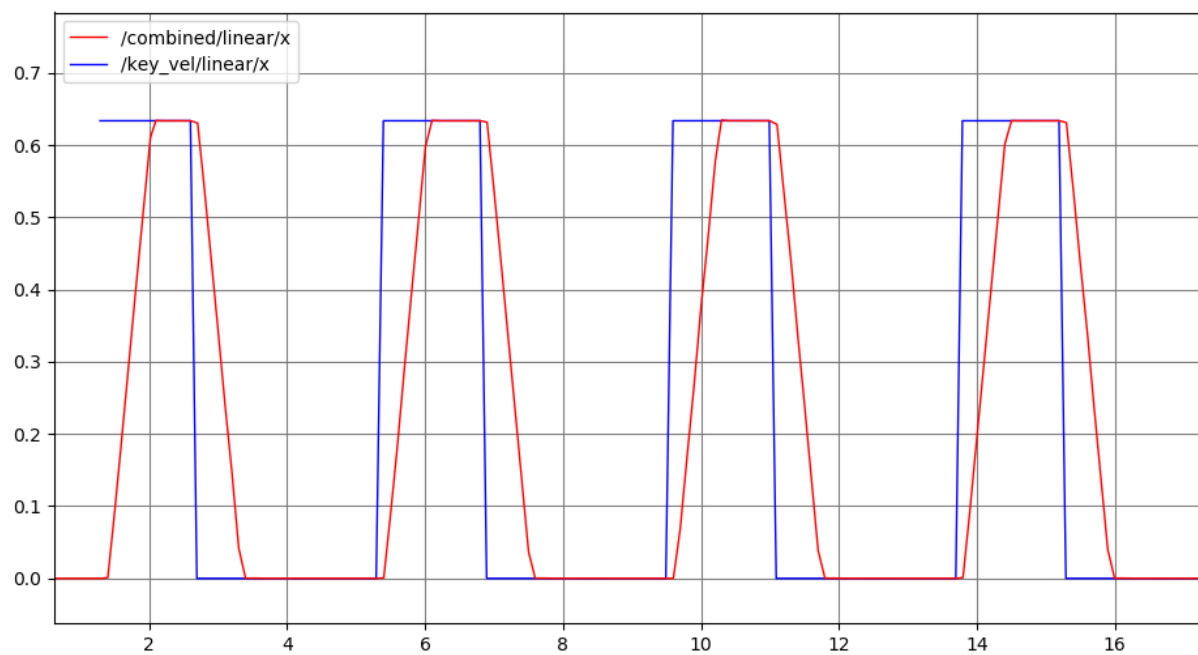
Rys. 11 Wygenerowana trajektoria przez robota



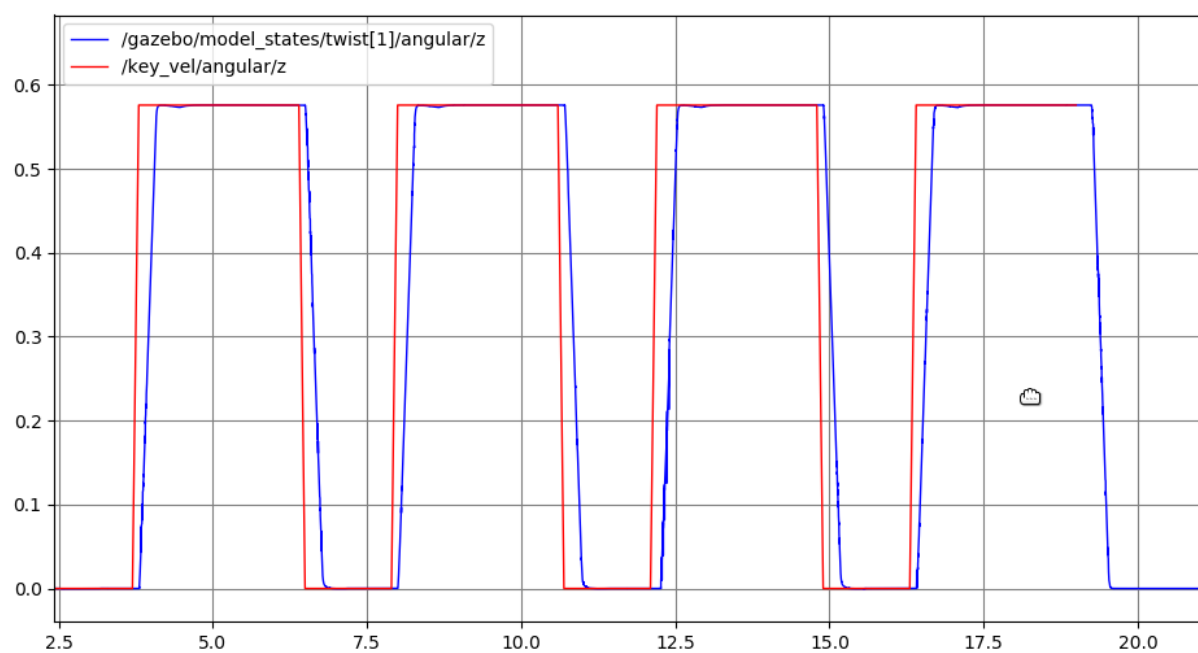
Rys. 12 Zależność położenia odczytanego z odometrii z położeniem referencyjnym w funkcji czasu



Rys. 13 Zależność położenia kąтового odczytanego z odometrii z położeniem referencyjnym w funkcji czasu (y – pomiar referencyjny)



Rys. 14 Przebieg zadawanej prędkości liniowej z referencyjną w funkcji czasu



Rys. 15 Przebieg zadawanej prędkości kątowej z referencyjną w funkcji czasu

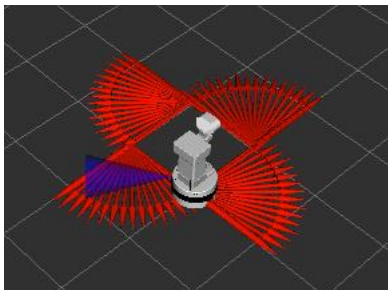


## Badanie implementacji bazującej na odometrii

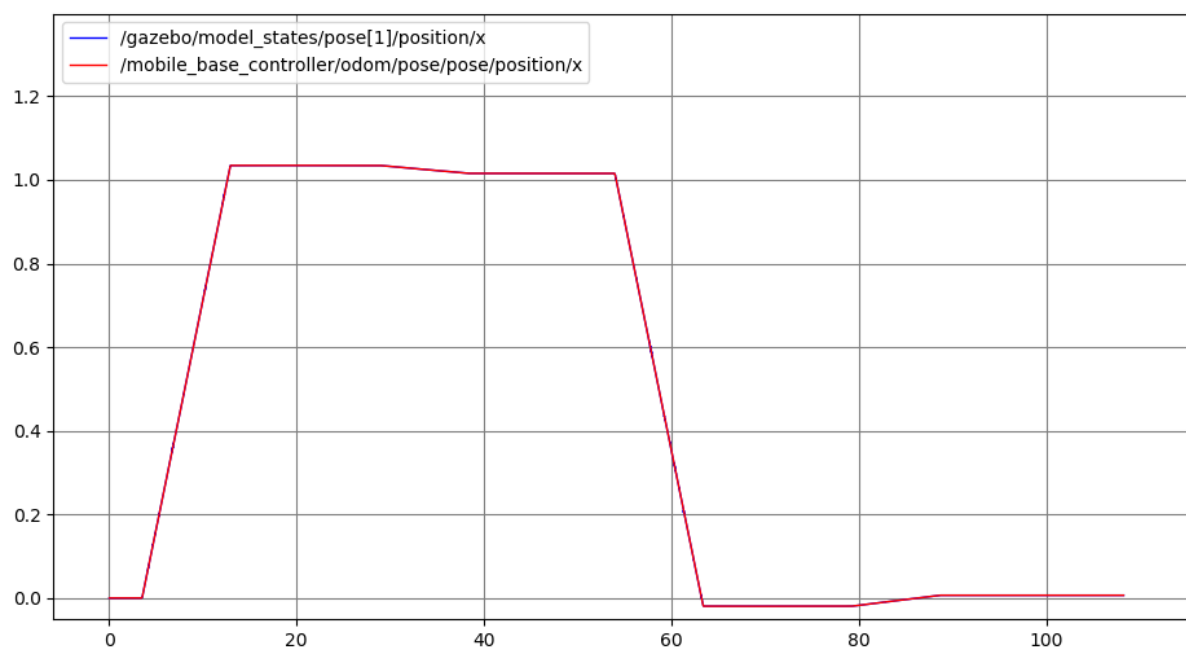
Badanie to przeprowadzono dla trzech wariantów prędkości:

- |                                 |                            |
|---------------------------------|----------------------------|
| 1. liniowa = 0.11               | kątowa = 0.1               |
| 2. liniowa = $0.11 * 2.4$       | kątowa = $0.1 * 2.4$       |
| 3. liniowa = $0.11 * 2.4 * 2.4$ | kątowa = $0.1 * 2.4 * 2.4$ |

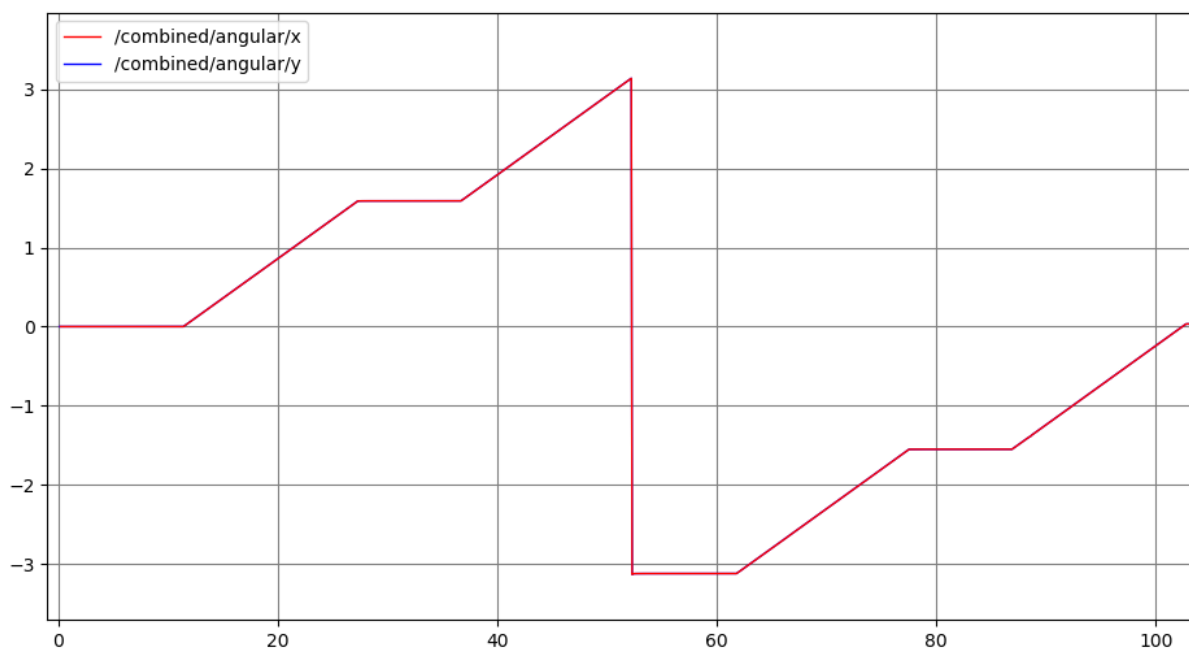
### Wariant 1.



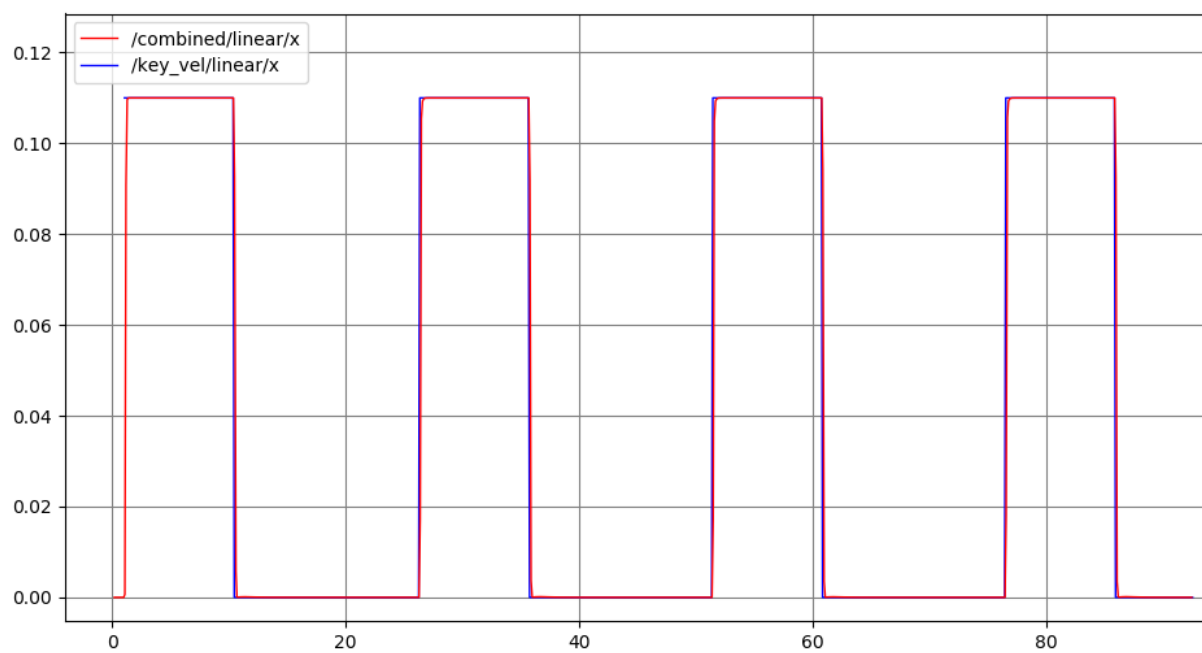
Rys. 16 Wygenerowana trajektoria przez robota



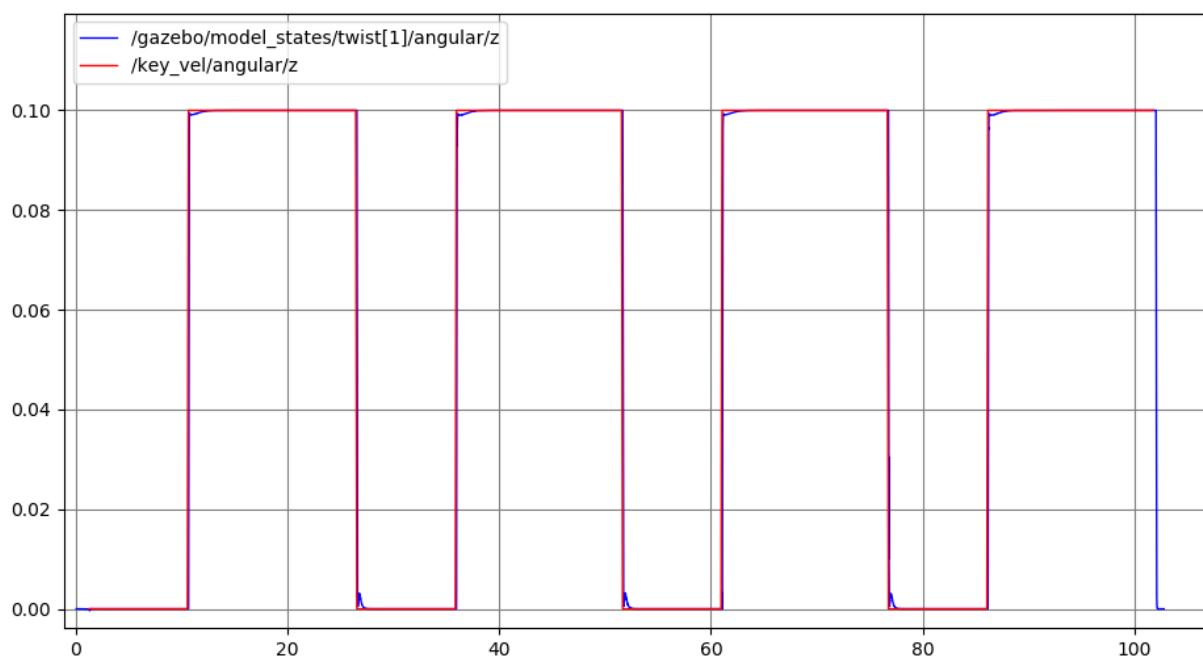
Rys. 17 Zależność położenia odczytanego z odometrii z położeniem referencyjnym w funkcji czasu



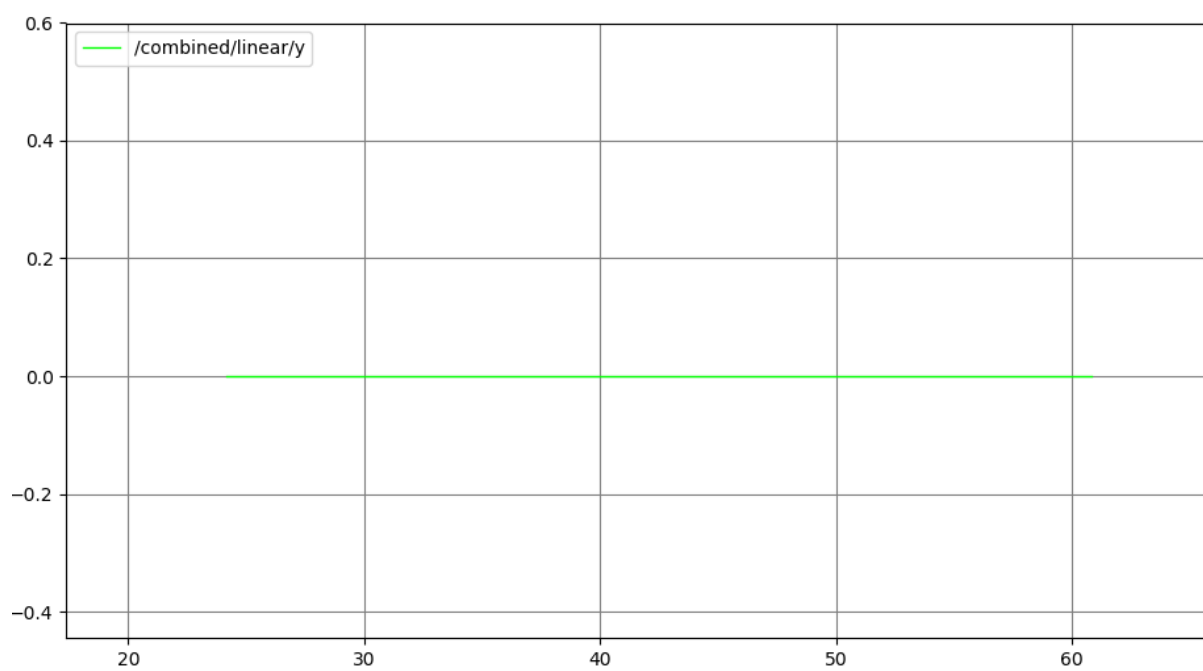
Rys. 18 Zależność położenia kąowego odczytanego z odometrii z położeniem referencyjnym w funkcji czasu (y – pomiar referencyjny)



Rys. 19 Przebieg zadawanej prędkości liniowej z referencyjną w funkcji czasu

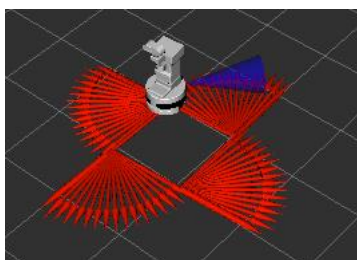


Rys. 20 Przebieg zadawanej prędkości kątowej z referencyjną w funkcji czasu

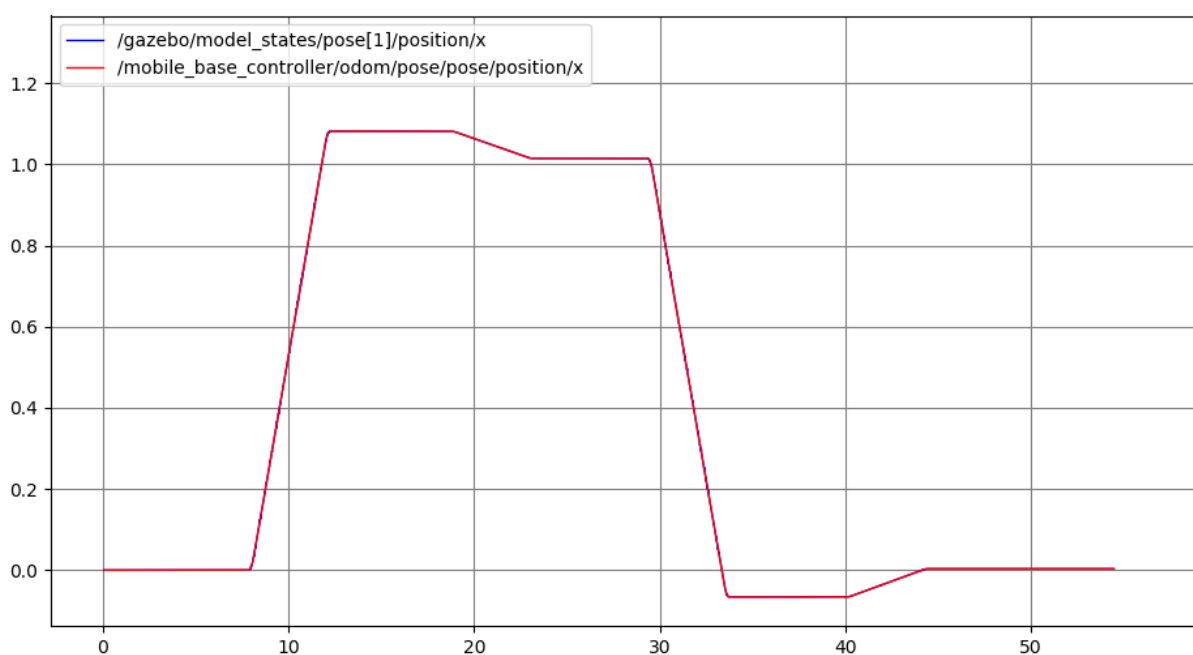


Rys. 21 Błąd chwilowy w funkcji czasu. Błąd sumaryczny wyniósł 0.0.

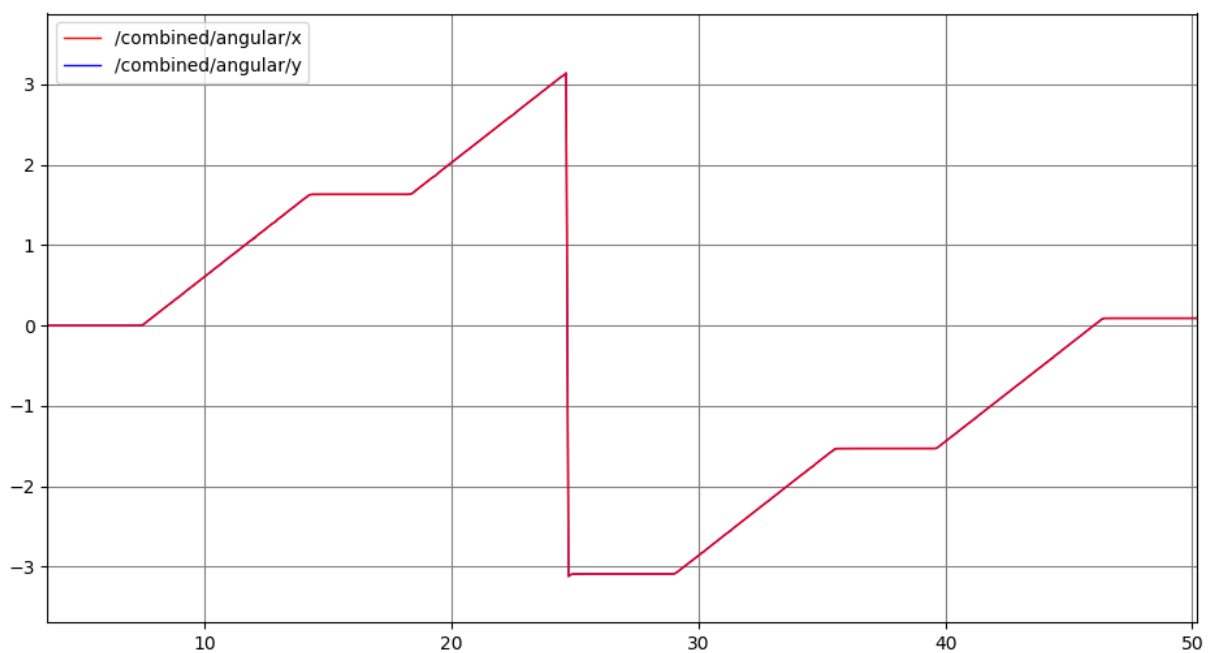
## Wariant 2.



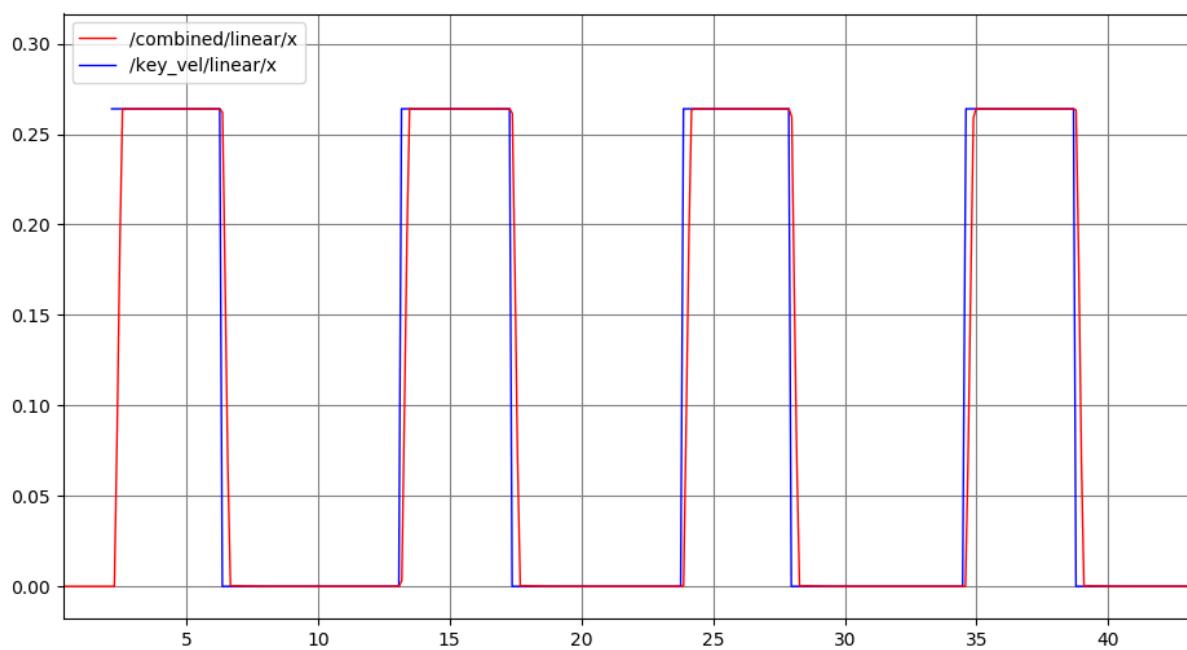
Rys. 22 Wygenerowana trajektoria przez robota



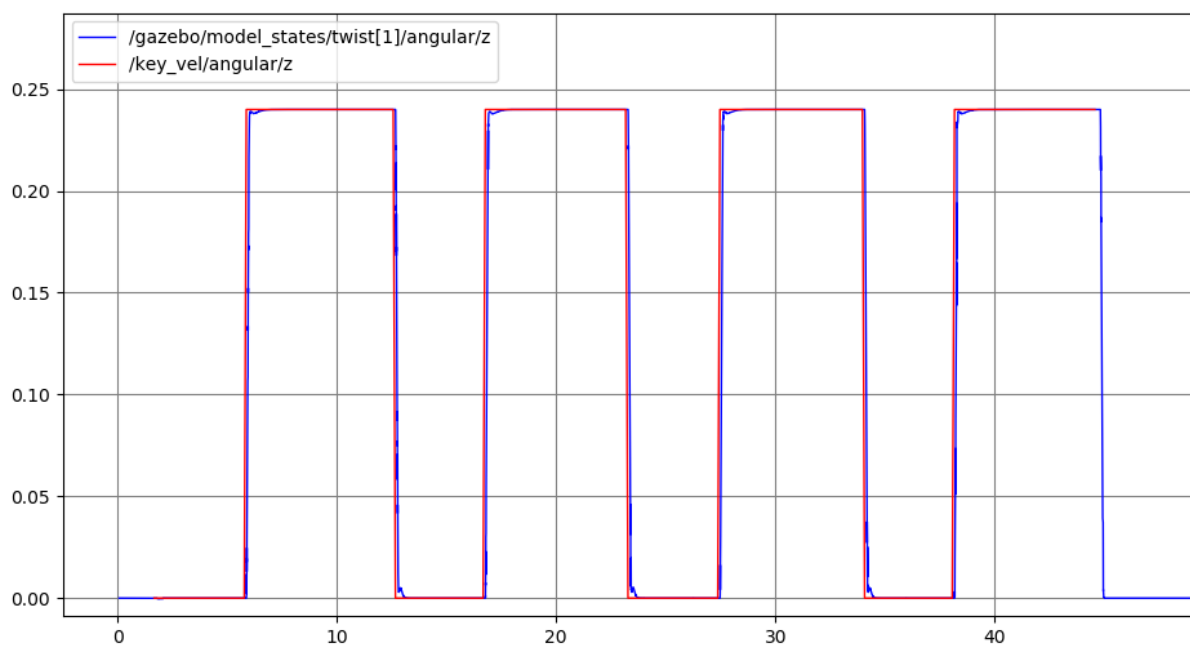
Rys. 23 Zależność położenia odczytanego z odometrii z położeniem referencyjnym w funkcji czasu



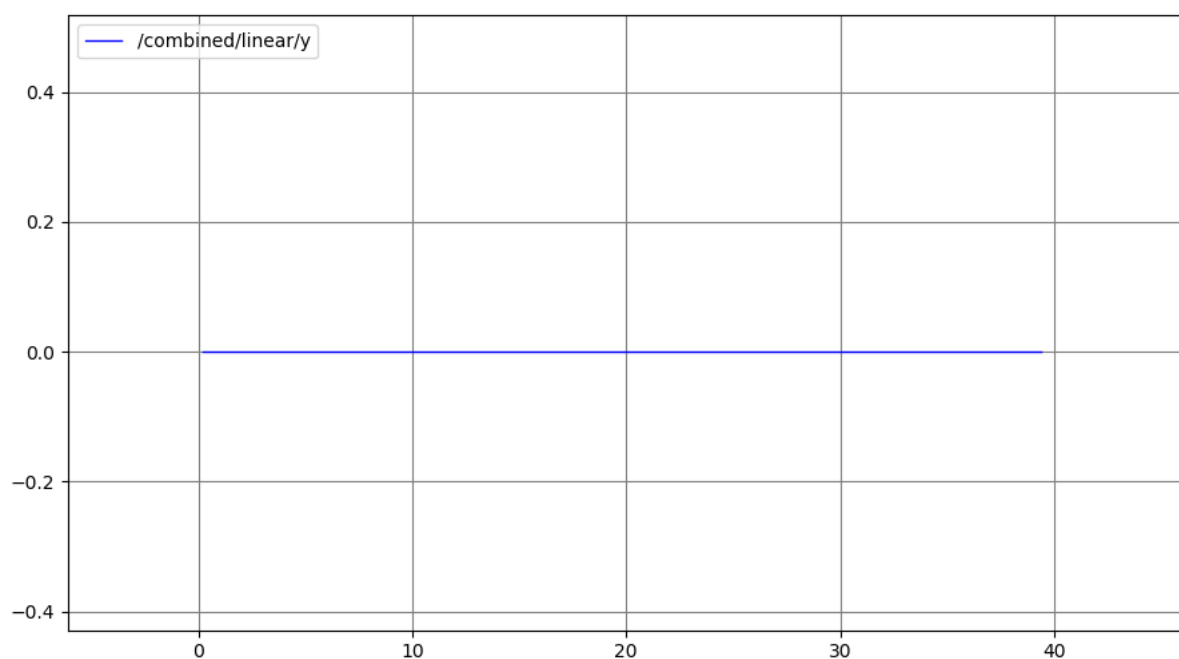
Rys. 24 Zależność położenia kąowego odczytanego z odometrii z położeniem referencyjnym w funkcji czasu (y – pomiar referencyjny)



Rys. 25 Przebieg zadawanej prędkości liniowej z referencyjną w funkcji czasu

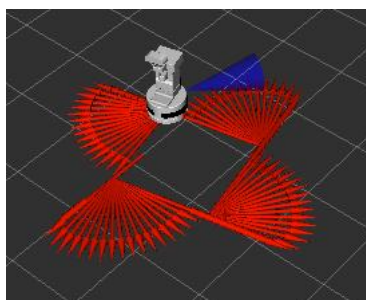


Rys. 26 Przebieg zadawanej prędkości kątowej z referencyjną w funkcji czasu

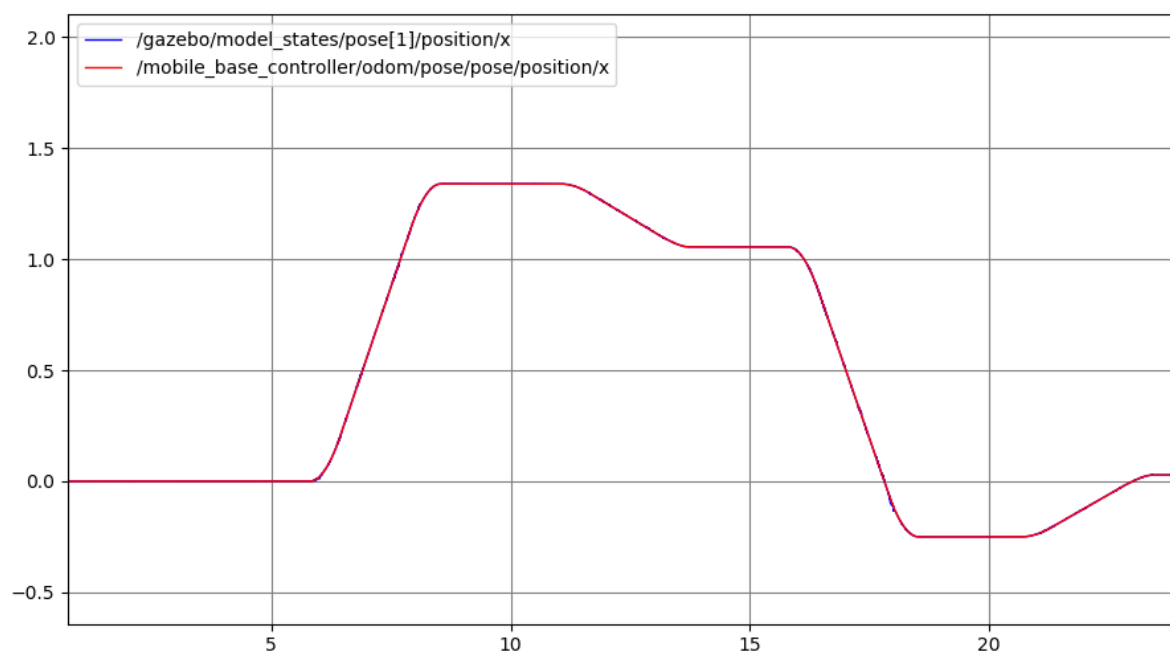


Rys. 27 Błąd chwilowy w funkcji czasu. Błąd sumaryczny wyniósł 0.0.

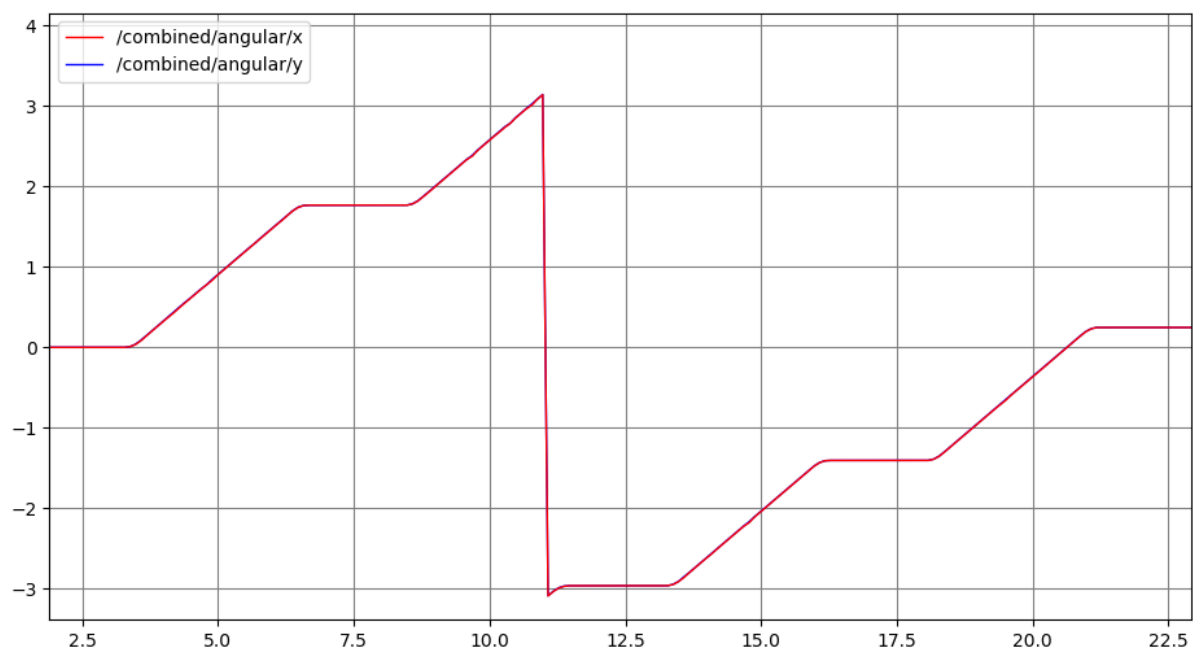
### Wariant 3.



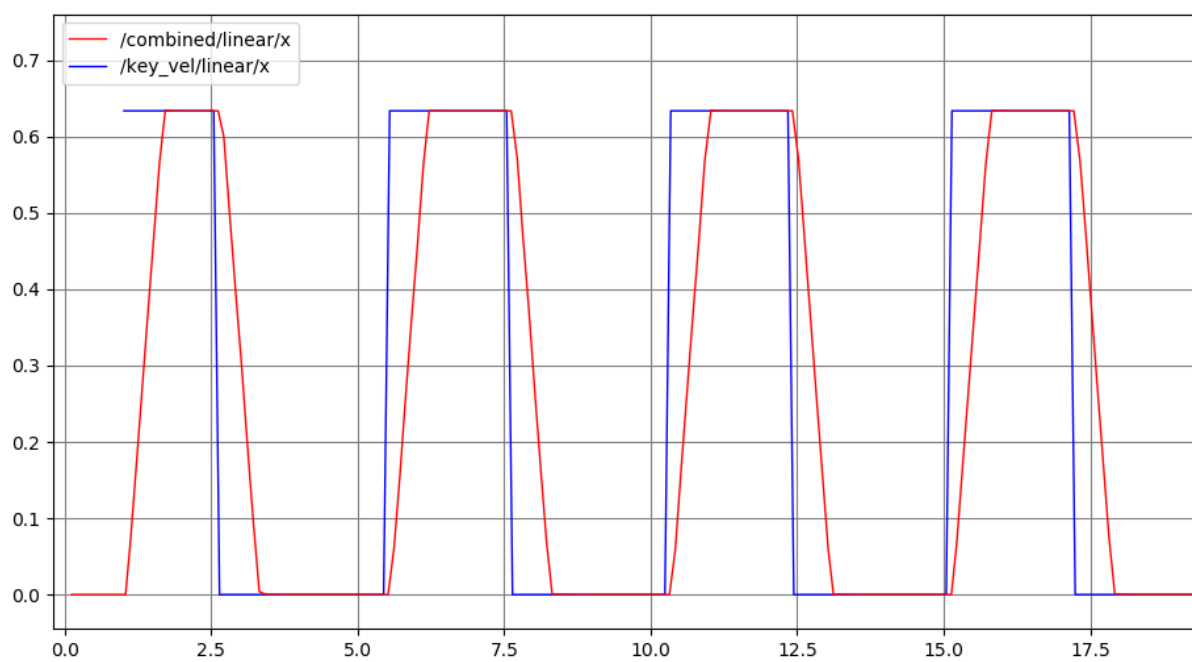
Rys. 28 Wygenerowana trajektoria przez robota



Rys. 29 Zależność położenia odczytanego z odometrii z położeniem referencyjnym w funkcji czasu

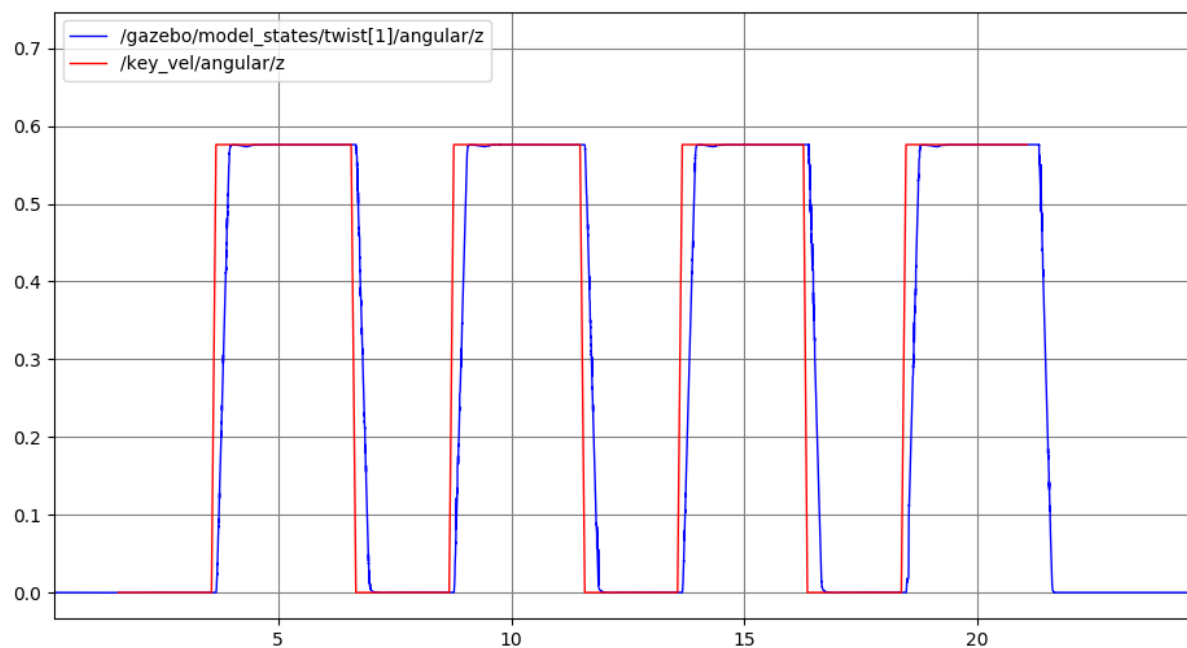


Rys. 30 Zależność położenia kąтового odczytanego z odometrii z położeniem referencyjnym w funkcji czasu (y – pomiar referencyjny)

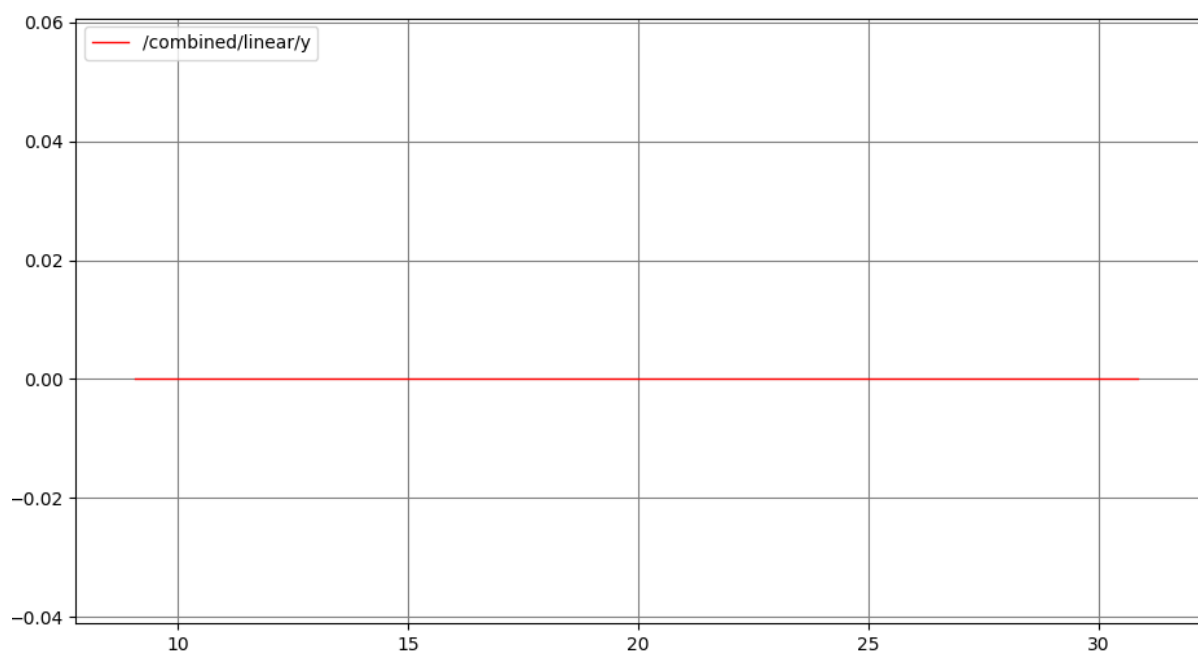


Rys. 31 Przebieg zadawanej prędkości liniowej z referencyjną w funkcji czasu



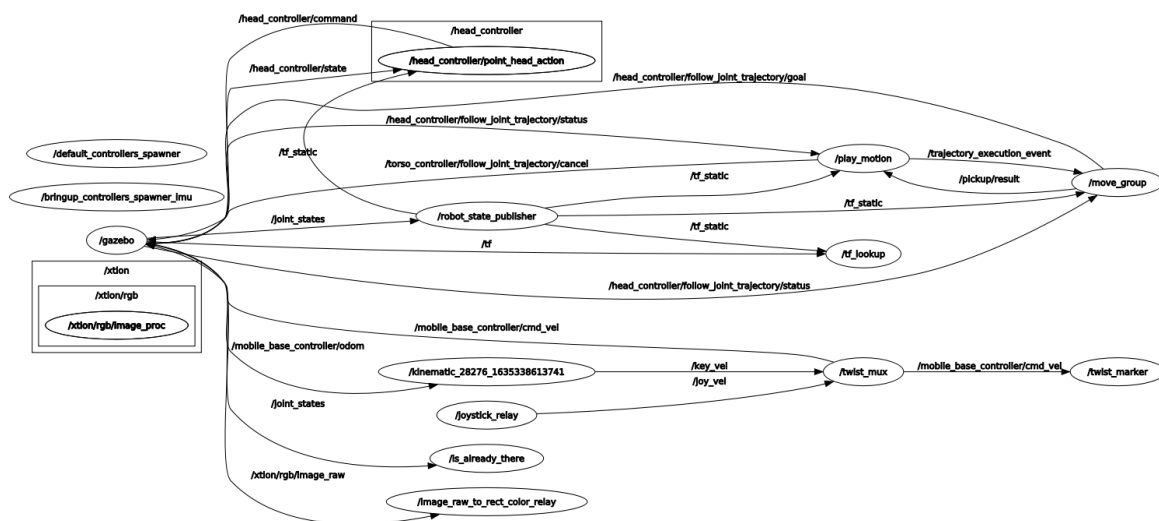


Rys. 32 Przebieg zadawanej prędkości kątowej z referencyjną w funkcji czasu



Rys. 33 Błąd chwilowy w funkcji czasu. Błąd sumaryczny wyniósł 0.0.

### Struktura połączenia - między-węzłowe



Rys. 34 Tryb symulacji



Rys. 35 Symulacja na podstawie danych zapisanych w rosbag

## Wnioski i spostrzeżenia

Przy analizie i tworzeniu wykresów za pozycje referencyjne przyjęto wiadomości publikowane na kanale `/gazebo/model_states/....`. Publikowane przez algorytm sterowania prędkości były na kanale `/key_vel/...`. Natomiast dane o odometrii publikowane przez robota były pobierane z kanału `/mobile_base_controller/odom/....`

W sprawozdaniu nie zostały dołączone analizy przebiegów dla symulacji algorytmów na rzeczywistym robocie. Próba zaprezentowania tej analiz została podjęta, jednak nie udało się w po prawny sposób przekonwertować układu odom na map (implementacja podjętej próby znajduje się w pliku `corrections.py`). Natomiast dogłębnie zostały przebadane symulacje algorytmów sterowania. Jak widać na przebiegach zarówno położenia  $x$ , jak i położenia kątownego ( $\theta$ ) położenie referencyjne praktycznie pokrywa się z położeniem uzyskiwanym z kanału odom. Potwierdza to także, bardzo mały błąd chwilowy oraz zerowy błąd sumaryczny (możliwe, że tak małe wartości zostały pominięte – na pewno błąd ten nie wynosi dokładnie zero). Można zauważyć z wykresów zadawanej i wykonywanej prędkości liniowej i kątowej, że robot działa z opóźnieniem, które jest widoczne przy coraz większych wartościach prędkości. Jest to zapewne związane z faktem, że robot nie od razu przyjmuje komendy, tzn. nie działa natychmiastowo i symulowaną fizyką robota – robot potrzebuje się rozpędzić i potem zahamować i nie dzieje się to momentalnie, a w pewnym czasie.

Podsumowując odometria dała zaskakująco dobre wyniki – szczególnie dlatego, że błąd pozycji w symulacji wyszedł równy 0. Można to argumentować tym, że prawdopodobnie symulacja nie symuluje wszystkich czynników i niedokładności pomiarów. Pewne jest to, że na rzeczywistym robocie takie wyniki zerowego błędu nie byłyby osiągalne.

Jeżeli coś należałoby poprawić w działaniu naszego programu to przede wszystkim trzeba by dodać stopniowe zwalnianie robota przed wierzchołkiem kwadratu. Stosowanie interpolacji liniowej na tym odcinku przy większych prędkościach, nie jest dobrym rozwiązaniem, ponieważ robot przemieszczając się z pełną prędkością nie może zatrzymać się momentalnie, więc jeżeli by zbliżał się do celu coraz wolniej to „przestrzelenie” (wyjechanie za punkt docelowy) było by zdecydowanie mniejsze o ile w ogóle zauważalne.

Autorzy: Lena Zubik, Mateusz Palczuk