

SPECYFIKACJA WSTĘPNA

„Symulacja ruchu robota typu wizyjny follow-line przy różnych algorytmach detekcji linii”

1. Opis działania programu.

- a. Program będzie sterował robotem (a dokładniej jego trójwymiarową symulacją w programie Gazebo) tak by podążał za czarną linią namalowaną na podłożu środowiska (zasymulowanego). Będzie udostępniał funkcję strojenia regulatora PID użytego przy wyznaczaniu prędkości kątowej robota (ręczne wpisanie nastaw) oraz możliwość wyboru jednego z dwóch sposobów wykrywania linii.
- b. Podczas realizacji zadania pojawią się następujące problemy cząstkowe:
 - Komunikacja Matlab z Gazebo – komunikacja między tymi dwoma oprogramowaniami będzie się odbywać poprzez system ROS. Matlab oferuje Robotics Toolbox, który zapewnia interfejs umożliwiający sprawne wysyłanie i odbieranie wiadomości w systemie ROS. Matlab będzie odbierał wiadomości zawierające obraz z kamery robota, a wysyłać będzie wiadomości zawierające prędkość kątową (która ma być zadana na robota).
 - Znaczące zużycie procesora – ponieważ zarówno Matlab jak i symulacja robota w Gazebo będą zużywały duże zasoby komputera, zdecydowano się na uruchomienie tych dwóch środowisk na oddzielnych komputerach. Środowiska będą komunikowały się ze sobą za pośrednictwem sieci (system ROS to umożliwia).
 - Wyliczanie prędkości robota – prędkość kątowa robota (zmieniana tak by robot podążał za linią) będzie wyliczana przy pomocy regulatora PID (zostanie wykorzystany PID dostępny w Control System Toolbox lub implementacja własna, w zależności od efektów regulacji). Wspomniana prędkość kątowa będzie wyjściem regulatora. Wejściem regulatora będzie natomiast uchyb, wyliczony jako różnica współrzędnej X środka obrazu (otrzymanego z robota), a współrzędną X określającą przybliżone położenie linii (wyliczenie jej zostanie omówione w następnym podpunkcie). Regulacja ta powinna zapewnić to, że robot będzie dążył do tego, by współrzędna X położenia linii była bliska współrzędnej X środka obrazu. Należy pamiętać, że robot będzie poruszał się cały czas do przodu, jako że prędkość liniowa robota będzie stała (dobrana eksperymentalnie).
 - Wykrywanie czarnej linii na obrazie z kamery – by rozwiązać ten problem cząstkowy zostanie użyty Image Processing Toolbox. Otrzymany z robota obraz zostanie przycięty do obszaru zainteresowania, a następnie poddany procesowi zmiany na skalę szarości i progowania, by otrzymać obraz binarny. Spodziewanym wynikiem będzie obraz, na którym linia będzie czarnym kształtem, a cała reszta obrazu będzie biała. Kolejne kroki zależą w dużej mierze od wykorzystanej metody określenia linii pojedynczą współrzędną X potrzebną do wyliczenia uchybu dla regulatora PID.
Dla pierwszej metody kolejne kroki będą następujące:
 - wyznaczenie centroidu otrzymanego czarnego kształtu (wykrytej linii)

- pobranie współrzędnej X otrzymanego punktu i wyliczenie uchybu

Dla drugiej metody kolejne kroki będą następujące:

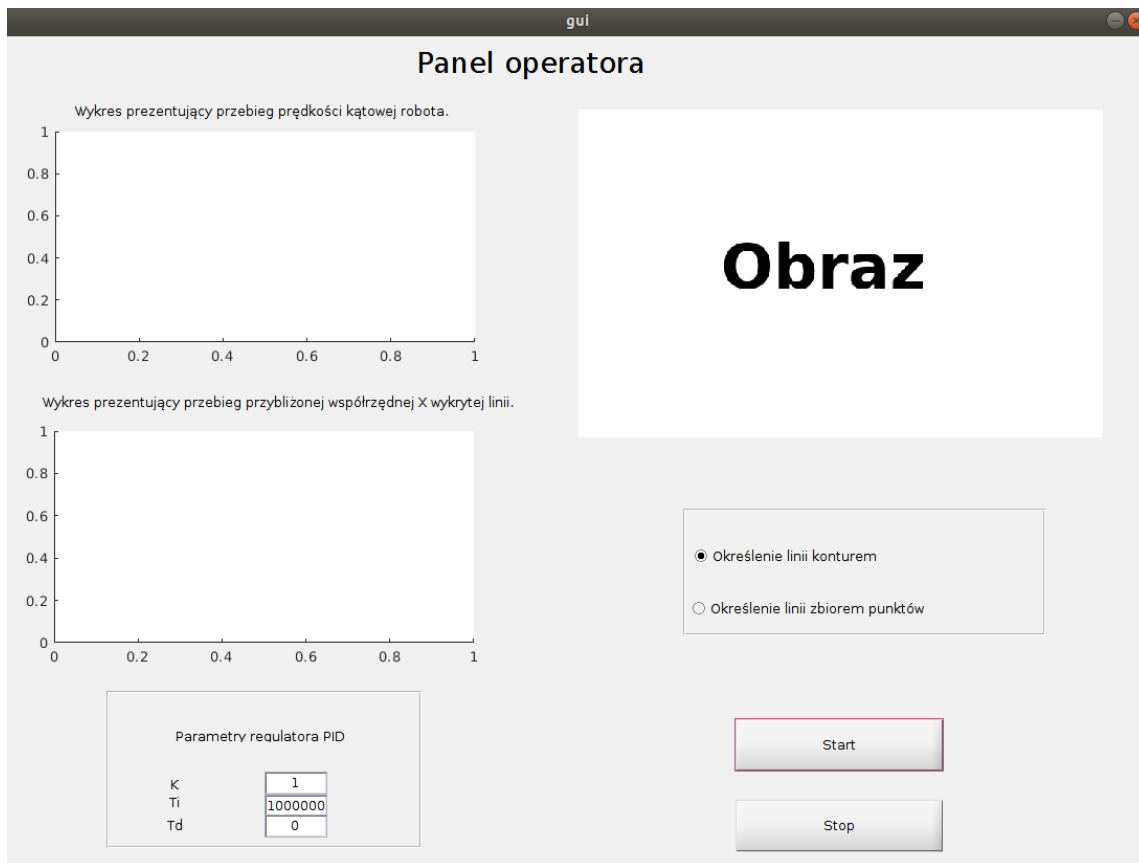
- wyliczenie średniej współrzędnej X dla czarnych pixeli dla danej współrzędnej Y. Przebadanie w ten sposób kilku współrzędnych Y obrazu, znajdujących się w równych odległościach
- wyliczenie średniej ważonej z zebranych średnich współrzędnych X (dla każdej współrzędnej Y), gdzie waga danej średniej współrzędnej X zależy od odpowiadającej jej współrzędnej Y.
- Otrzymana średnia ważona zostanie użyta do wyliczenia uchybu dla regulatora PID

Należy wspomnieć jednak że omawiany program nie będzie służył do porównania dwóch metod wykrywania linii, a jedynie do wizualizacji ich działania.

- c. Dane wejściowe będą pobierane z kamery zamontowanej na robocie (w symulacji) oraz od użytkownika programu (np.: nastawy regulatora PID).
- d. Wynikiem działania programu będzie wyliczenie odpowiedniej prędkości kątowej (liniowa będzie stała) robota, tak by płynnie podążał za linią na podłożu. Program będzie także prezentował 2 wykresy: wykres przebiegu wartości sterowania (prędkości kątowej) zadawanej na robota, oraz wykres prezentujący przebieg wartości współrzędnej X, określającej przybliżone położenie linii (na obrazie). Będzie także na bieżąco pokazywany obraz z kamery robota. Rysowane na nim będą dwie pionowe linie: pierwsza linia będzie nieruchoma i będzie miała współrzędną X środka obrazu, druga linia będzie ruchoma i zawsze będzie miała współrzędną X odpowiadającą przybliżonemu położeniu linii, za którą podąża robot.
- e. Dane wyjściowe nie będą mogły być zapisywane.

2. Opis GUI

- a. Poglądowy rysunek GUI.



b. Aplikacja będzie jedno okienkowa.

c. Wykorzystane elementy Handle Graphics:

- Axes
- Panel
- Static Text
- Edit Text
- Push Button
- Radio Button
- Button Group

3. Opis wykorzystywanych toolboxów Matlaba ze wskazaniem głównych

funkcji/mechanizmów zawartych w tych toolboxach, a wykorzystywanych w programie.

- a. Robotics Toolbox – zostaną wykorzystane funkcje umożliwiające m.in.: nawiązanie komunikacji z systemem ROS (funkcja `rosinit`), tworzenie, wysyłanie oraz odbieranie wiadomości systemu ROS w celu komunikacji z robotem (obiekty `rospublisher`, `rosmesssage`, `rossubscriber`).
- b. Control System Toolbox – zostanie wykorzystana funkcja tworząca dyskretny regulator PID (np.: `pidstd`), w celu regulacji prędkości kątowej robota.
- c. Image Processing Toolbox – zostaną wykorzystane mechanizmy umożliwiające swobodne operowanie na obrazie, w tym: konwersja obrazu na czarno-biały (funkcja `rgb2gray`), progowanie obrazu by uzyskać obraz binarny (funkcja `imbinarize`), wyliczenie centroidu otrzymanego kształtu wykrytej linii (`regionprops`), rysowanie po obrazie.