

Sprawozdanie projektu z przedmiotu Sztuczna Inteligencja

Opis projektu

Tematem naszego projektu była symulacja drona. Została ona stworzona w języku programowania Python. Porównane zostały różne algorytmy wykorzystane do automatycznego sterowania pochyleniem drona:

- Logika rozmyta,
- Sieć neuronowa,
- Regulator PID.

Zastosowane biblioteki

- **pygame**: biblioteka do utworzenia okienkowej aplikacji w języku Python
- **pymunk**: biblioteka zajmującą się podstawową fizyką drona, dzięki niej mogliśmy zasymulować grawitację oraz inne siły działające na drona.
- **numpy**: biblioteka dodająca obsługę dużych, wielomiarowych tablic i macierzy, a także duży zbiór funkcji matematycznych wysokiego poziomu.
- **fpstimer**: pomocnicza biblioteka do określenia ilości klatek na sekundę aby na dowolnym komputerze symulacja działała w zbliżony sposób.
- **skfuzzy**: biblioteka do obsługi logiki rozmytej
- **tensorflow**: platforma do uczenia maszynowego
- **keras**: biblioteka do obsługi sieci neuronowej

Początkowe założenia

Początkowym założeniem było stworzenie symulacji drona który potrafi sam utrzymywać swoją pozycję. Posiada on dwa silniki, z lewej oraz prawej strony podwozia. Jesteśmy w stanie nadać siłę w każdym silniku o przedziale $<0,1>$.

Końcowy efekt

Efektem końcowym jest dron z modulem AI odpowiadającym za utrzymywanie równowagi (pochylenia). Użytkownik ma możliwość sterowania pochyleniem docelowym, zmieniając w ten sposób pozycję drona. Istnieje możliwość wyboru modułu logiki drona:

- logika rozmyta,
- sieć neuronowa,
- logika oparta na regulatorze PID,
- brak AI.

Dron utrzymuje określone pochylenie. Wprowadza to znaczne ułatwienie sterowania.

Używając strzałki do góry dron nadaje równą moc dla obu silników. Używając bocznych strzałek zmieniamy docelowe pochylenie drona, w ten sposób zmieniając kierunek jego lotu. Dzięki modułowi SI dbającym o utrzymanie stałego pochylenia, sterowanie dronem jest znacznie ułatwione.

Istnieje również możliwość całkowitego wyłączenia SI. W tej wersji symulacji boczne strzałki odpowiadają za moc poszczególnych silników.

Logika rozmyta

Moduł SI oparty na logice rozmytej został stworzony przy użyciu biblioteki skfuzzy. Kontroler jako dane wejściowe przyjmuje kąt pochylenia oraz przyspieszenie kątowe. Obie wejściowe wartości liczbowe są modelowane w następujące funkcje przynależności:

- "negative" - dron pochyla się w lewą stronę (prędkość / nachylenie są ujemne),
- "none" - wartość równa lub bliska zeru,
- "positive" - dron pochyla się w prawą stronę (prędkość / nachylenie są dodatnie).

Dla obu silników niezależnie modelowane są funkcje:

- "low" - moc zerowa lub bliska zeru,
- "average" - średnia moc,
- "positive" - duża moc.

Użyte zostały funkcje trójkątne i trapezowe.

Następnie stworzone zostały reguły określające dla jakiego kąta nachylenia oraz prędkości kątovej, jaka oczekiwana jest moc każdego z silników.

Co stały, określony czas do modułu logiki rozmytej podawane są wartości określające aktualny stan drona. Następnie wartości zwrócone przez kontroler są ustawiane dla obu silników.

Sieć neuronowa

Za pomocą biblioteki Tensorflow została stworzona i wytrenowana sieć neuronowa.

Dane do trenowania zostały wygenerowane w następujący sposób:

1. Wylosowanie N losowych pozycji drona,
2. Podanie wylosowanych pozycji do modułu logiki rozmytej,
3. Obliczenie odpowiedniej reakcji.
4. Zapisanie do pliku csv symetrycznych akcji

W ten sposób został wygenerowany plik csv, zawierający: oczekiwaną siłę lewego i prawego silnika, aktualną pozycję i prędkość kątową drona.

Wygenerowane dane zostają później wczytane do pamięci, znormalizowane do wartości od zera do jeden. Znormalizowane dane są potem dostarczone do treningu AI.

Podczas lotu drona, w stałych odstępach czasu, zostaje odczytana wartość pozycji i prędkości kątowej drona, znormalizowana do wartości 0-1. Następnie podana do modelu który przewiduje siłę lewego i prawego silnika.

Regulator PID

Prosty moduł, oparty na kontrolerze PID.

Wnioski

Po obserwacjach możemy stwierdzić że najlepszym wyborem była logika rozmyta. Okazała się stosunkowo prosta w implementacji i dała zadowalające rezultaty. Mimo to, że dron korzystając z SI opartej na sieci neuronowej okazał się mniej stabilny, nie można powiedzieć że ten algorytm okazał się gorszy, ponieważ sieć neuronowa uczyła się na podstawie logiki rozmytej. W związku z tym nie możemy bezpośrednio porównywać tych dwóch metod.

Ciekawym zjawiskiem okazał się prosty moduł oparty na kontrolerze PID. Był zdecydowanie najprostszym w implementacji algorytmem i co interesujące okazało się że dron jest najstabilniejszy przy użyciu właśnie tego kontrolera.

Kontrola drona bez pomocy sztucznej inteligencji jest wręcz niemożliwa, a dzięki każdej z tych metod jesteśmy w stanie swobodnie się nim poruszać.