

SST MISK

Projekt semestralny

Monitorowanie celu za pomocą autonomicznej chmury dronów
opracowanie algorytmu kooperacji i symulacja

Jerzy Baranowski
Artur Czopor
Ignacy Ruksza
Krzysztof Zarzycki

May 14, 2018

1 Opis projektu

Celem projektu jest stworzenie i symulacja działania algorytmu sterowania rojem dronów tak by w sposób optymalny formowały ustaloną formację nad wyznaczonym celem.

2 Użyte oprogramowanie

Symulacja wyżej opisanego zadania została wykonana w programie symulacyjnym V-Rep połączonym dedykowanym API ze środowiskiem Matlab. W celu realizacji zadania został zaimplementowany w języku M skrypt optymalizujący trasę przelotu drona w zależności od pozycji celu i roju. Obiekty latające odwzorowane są w środowisku V-Rep poprzez model symulacyjny "Quadricopter", który posiada możliwość sterowania poprzez wyznaczenia pozycji docelowej. Model ten został dostarczony przez producenta oprogramowania i ingerencja w jego mechanikę nie jest celem projektu. W celu obejścia ograniczeń modelu trajektoria wyliczana jest wyznaczana poprzez generację punktów docelowych dla kolejnych chwil dla każdego z dronów.



Figure 1: Przykładowa scena z symulatora V-Rep.

3 Opis algorytmu

Zadaniem, które mają wykonać drony, jest podążanie za poruszającym się człowiekiem w odpowiedniej formacji (kwadrat, wielokąt). Założone zostało, że drony startują z losowych miejsc na planszy i znają położenie zbiega. Muszą one podążać w jego kierunku optymalną ścieżką, otoczyć go w formacji (która na bieżąco tworzona jest w czasie ich lotu) a następnie go śledzić. Przy tym nie powinno dojść do sytuacji, w której dwa drony zderzą się ze sobą lub ze śledzonym człowiekiem. Zadanie to zostało opisane za pomocą nieliniowego problemu optymalizacji.

3.1 Problem optymalizacji

W celu wyznaczenia optymalnej pozycji dronów rozwiązywany jest nieliniowy problem optymalizacji wielu zmiennych z liniowymi i nieliniowymi oraz z równościowymi i nierównościowymi

ograniczeniami. Formalny zapis problemu:

$$\min_x f(x) = \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases} \quad (1)$$

Uwzględnione zostały tylko więzy opisane funkcją $c(x)$. Dane problemu:

- X - zmienna celu
- D - zadana odległość od celu
- d - zadana macierz odległości do sąsiada
- A - minimalna odległość do celu
- a - minimalna odległość do sąsiada
- L - liczba dronów

3.2 Funkcja celu

Pierwsza część równania opisuje zadanie zachowanie odpowiedniej odległości od celu. Druga natomiast opisuje geometrię żądanej formacji dronów, poprzez zdefiniowanie zalecanych odległości pomiędzy parami dronów.

$$f_{target}(x) = \underbrace{\sum_{i=0}^L \{D_i^2 - [(X_x - x_{xi})^2 + (X_y - x_{yi})^2]\}^2}_{\text{Odległość od celu}} + \underbrace{\sum_{i=0}^L \sum_{j=0}^L w_i \{d_{ij}^2 - (X_y - x_{yi})^2\}^2}_{\text{Odległości pomiędzy dronami}} \quad (2)$$

Opis zmiennych:

- X - zmienna celu
- x - zmienne pozycji
- D - zadana odległość od celu
- d - zadana macierz odległości do sąsiada
- L - liczba dronów
- w_i - wagi

Powyższa funkcja jest wczytywana do procedury optymalizacyjnej rozwiązującej zadanie optymalizacji kwadratowej.

3.3 Funkcja więzów

Funkcja $c(x)$ definiuje więzy narzucone na układ.

$$f_{constraints}(x)_k = c(x) = \begin{cases} A^2 - [(X_x + x_x)^2 + (X_y + x_y)^2] & k \in 0, \dots, L-1 \\ a^2 - [(x_{xj} + x_{xi})^2 + (x_{yj} + x_{yi})^2] & k \in L, \dots, \binom{L}{2} \end{cases}$$

Pierwsza część równania opisuje warunek na zachowanie minimalnej odległości od celu. Druga natomiast opisuje minimalne dopuszczalne odległości pomiędzy parami dronów. Dla L statków powietrznych definiujemy $\binom{L}{2}$ warunków. Opis zmiennych:

- X - zmienna celu
- x - zmienne pozycji
- A - minimalna odległość do celu
- a - minimalna odległość do sąsiada
- x_0 - pozycja początkowa
- dx - maksymalne przesunięcie
- L - liczba dronów

Powyższa funkcja jest wczytywana do procedury optymalizacyjnej rozwiązującej zadanie optymalizacji kwadratowej.

4 Implementacja

Pętla główna programu obliczającego trajektorie dronów zaprezentowana jest na poniższym diagramie akcji.

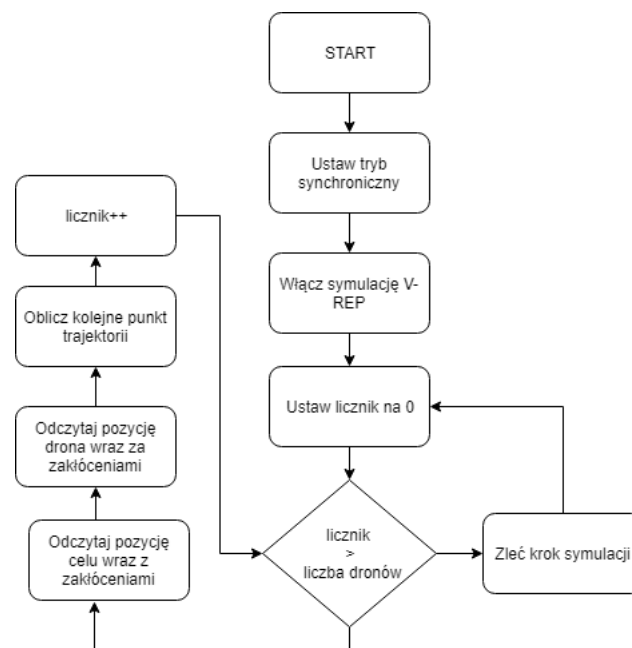


Figure 2: Uproszczony diagram akcji.

Zadanie optymalizacji jest rozwiązywane przez funkcję `fmincon(..)`.