# SST MISK
# Projekt semestralny

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

Jerzy Baranowski
Artur Czopor
Ignacy Ruksza
Krzysztof Zarzycki

June 10, 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 latający 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 podążyć 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:

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

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

Funkcja celu składa się zasadniczo z dwóch członów. Pierwszy z nich odpowiada za ich odległość od śledzonego człowieka. Znana jest pozycja każdego z L dronów, a także położenie celu, w kierunku którego mają lecieć. Zastosowana została więc minimalizacja kwadratu różnicy zadanej odległości między celem, a każdym z dronów. Drugi człon równania odpowiada za minimalizację odległości pomiędzy dronami. Parametr d to macierz, która określa położenie każdego z dronów w stosunku do jego sąsiada o strukturze:

$$d = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1L} \\ d_{21} & d_{22} & \cdots & d_{dL} \\ \vdots & \vdots & \ddots & \vdots \\ d_{L1} & d_{L2} & \cdots & d_{LL} \end{bmatrix}$$

gdzie dij – jest odległością między i-tym, a j-tym dronem. W szczególności gdy i=j, to dij wynosi 0.

Przykładowo, dla czterech dronów i zadanej formacji o kształcie kwadratu o boku a:

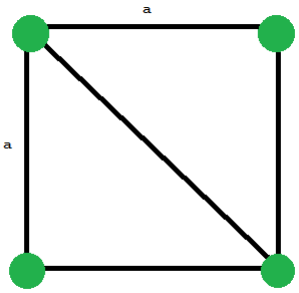


Figure 2: Formacja dla czterech dronów.

Macierz ma postać:

$$d = \begin{bmatrix} 0 & a & a\sqrt{2} & a \\ a & 0 & a & a\sqrt{2} \\ a\sqrt{2} & a & 0 & a \\ a & a\sqrt{2} & a & 0 \end{bmatrix}$$

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

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.

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

Pierwsza część równania opisuję warunek na zachowanie minimalnej odległości od celu. Druga natomiast opisuję 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 trajektorię dronów zaprezentowana jest na poniższym diagramie akcji.

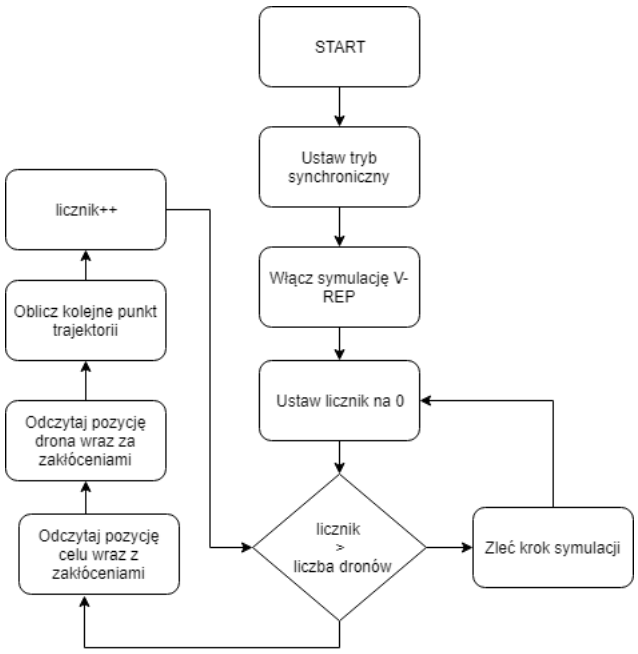


Figure 3: Uproszczony diagram akcji.

Zadanie potymalizacji jest rozwiązywane przez funkcję fmincon(..).

# 5 Realizacja

Docelowo został utworzony interfejs użytkownika(fig.4) umożliwiający zmianę parametrów symulacji m.in. maksymalne przyśpieszenie i prędkość ruchu dronów, odległość od celu zachowywaną przez drony po utworzeniu formacji oraz odległość pomiędzy sąsiednimi dronami a także wybór dostępnych formacji.

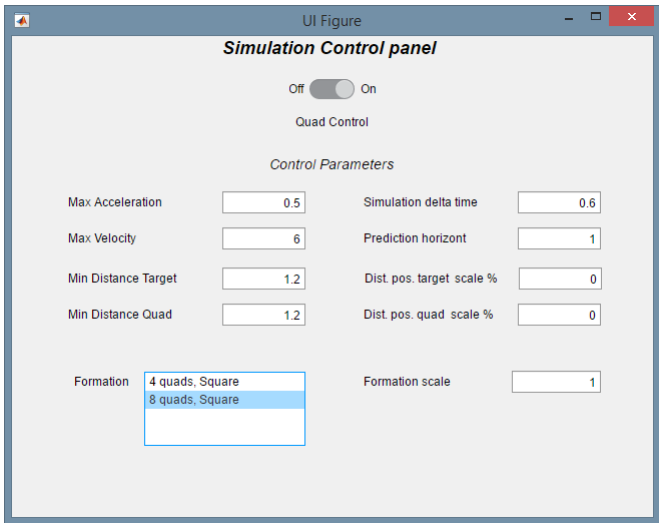


Figure 4: Interfejs użytkownika.

Działanie zaimplementowanej optymalizacji dla czterech oraz ośmiu dronów, tworzących formację "kwadrat" nad celem, zostało przedstawione na poniższych rysunkach. Rysunki 5 i 7 przedstawiają początkowe położenie dronów oraz celu na planszy w programie V-rep. Natomiast Rysunki 6 i 8 przedstawiają utworzoną formacje nad przemieszczającym się człowiekiem.

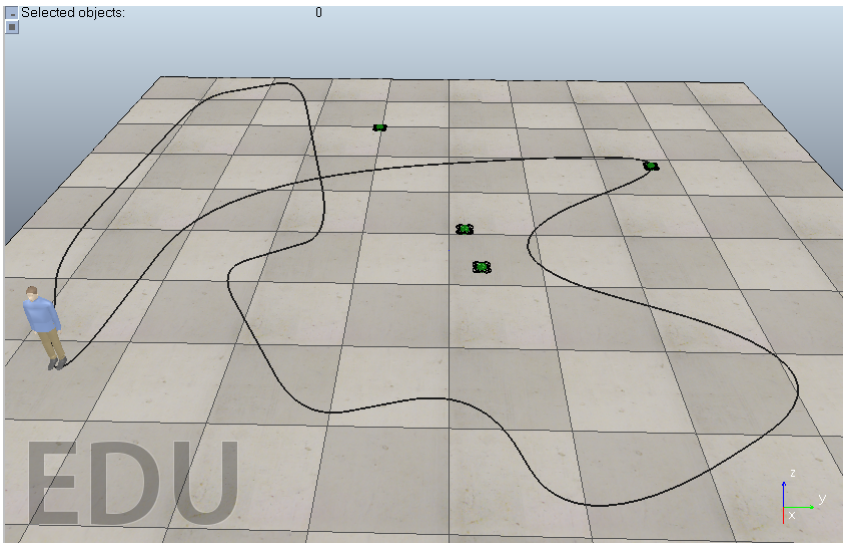


Figure 5: Początkowe położenie dla symulacji czterech dronów.

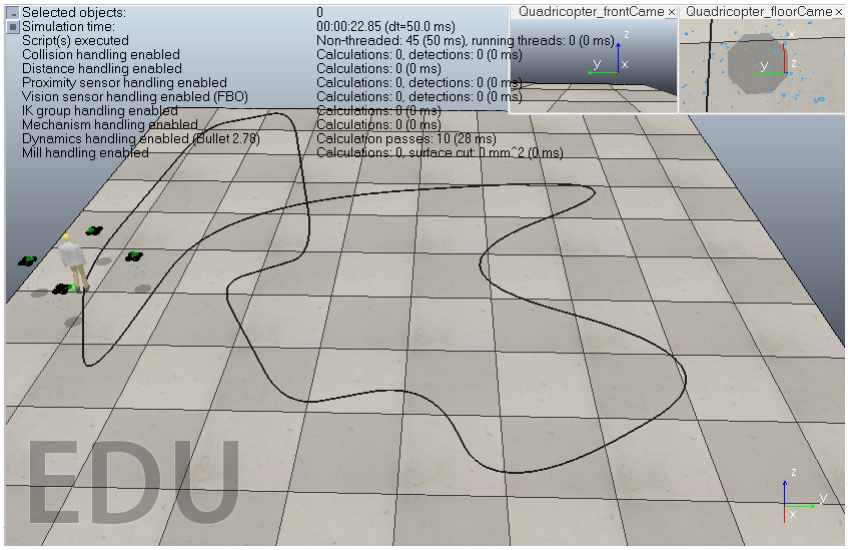


Figure 6: Utworzona formacja dla czterech dronów.

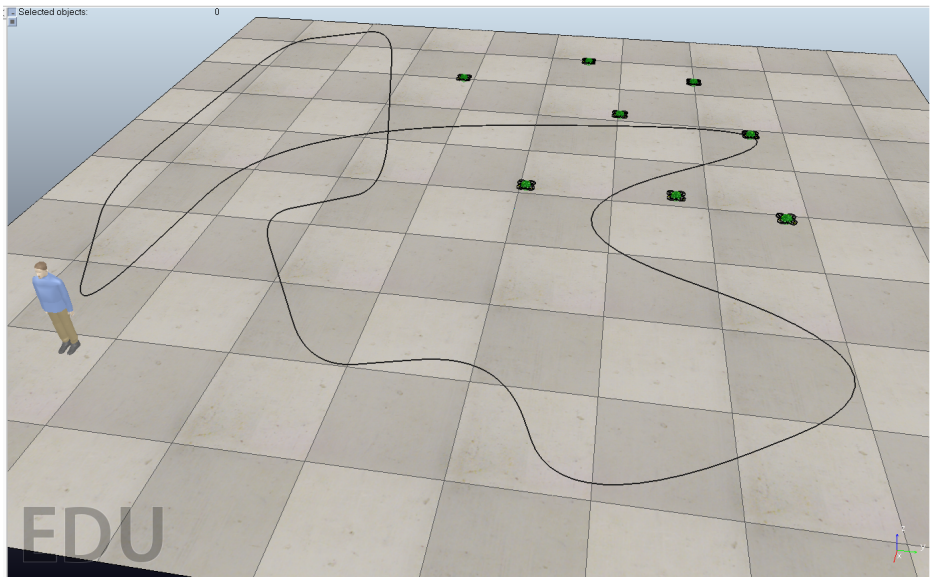


Figure 7: Początkowe położenie dla symulacji ośmiu dronów.

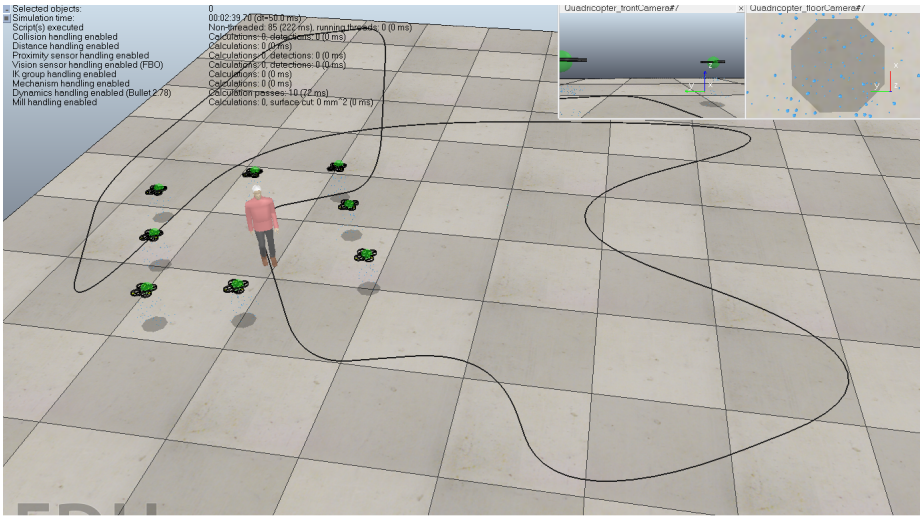


Figure 8: Utworzona formacja dla ośmiu dronów.

## 6 Analiza rozwiązania

W celu oceny poprawności oraz jakości zaimplementowanego zadania optymalizacji, wykreślono przebiegi w czasie, przedstawiające porównanie działania programu bez optymalizacji oraz po zastosowaniu optymalizacji. Pierwszy przebieg przedstawia sumaryczną drogę pokonaną przez każdego drona z pozycji początkowej do momentu utworzenia formacji nad celem. Można zauważyć, że po zastosowaniu optymalizacji znacznie skrócił się czas jaki jest potrzebny do utworzenia formacji z położenia początkowego. Drugą bardzo istotną korzyścią płynącą z zastosowania zadania optymalizacji, jest odległość przebyta przez drony. Po jej zastosowaniu dystans jaki jest potrzebny do utworzenia formacji nad celem zmalał o 30%.
Drugi przebieg przedstawia sumaryczne koszty energetyczne dla każdego statku, wykorzystywane do utworzenia formacji z położenia początkowego. W tym przypadku również można zauważyć znaczną poprawę po zastosowaniu zadania optymalizacji. Koszty energetyczne są znacznie mniejsze, również zmalały o 30% w stosunku do ruchu dronów bez zadania optymalizacji.

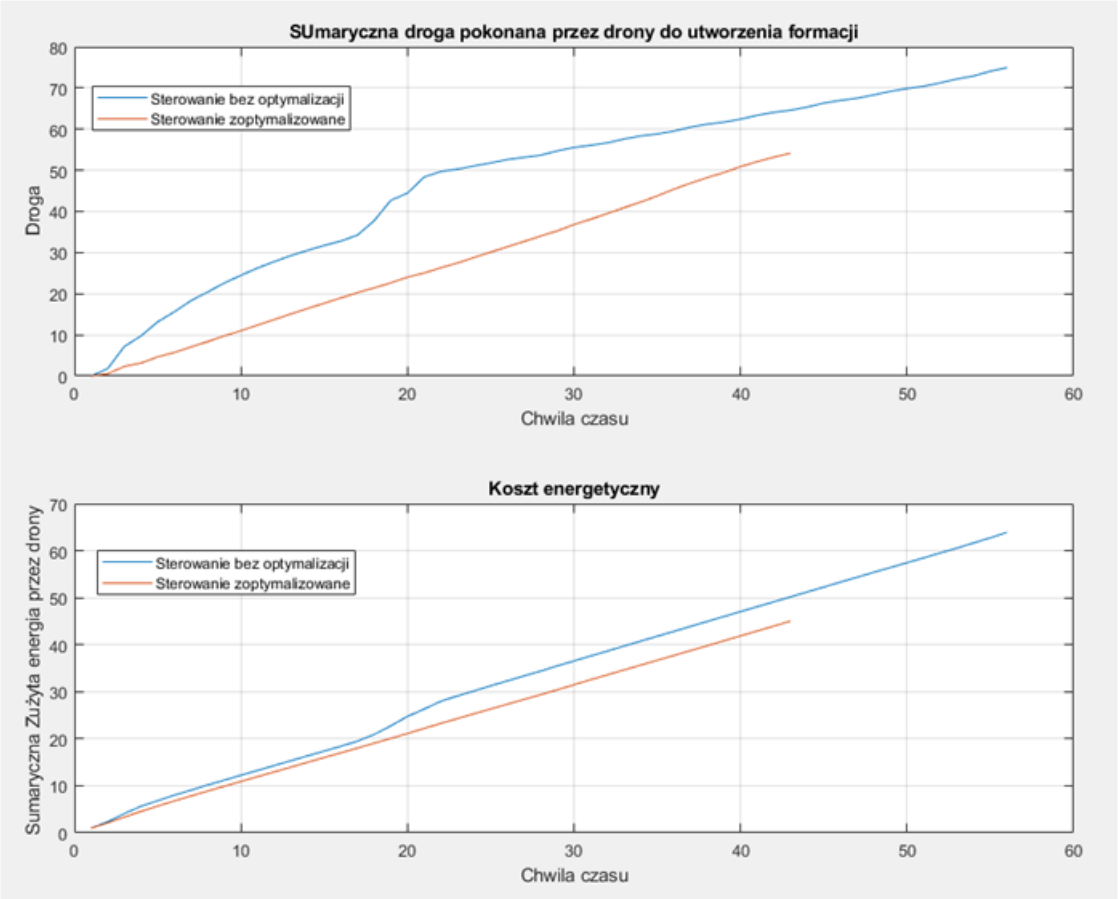


Figure 9: Rezultaty przed i po zastosowaniu optymalizacji.

# 7 API

Funkcja ConstraintFunction definiująca więzy narzucone na układ:
$function[c, ceq] = ConstraintFunction(a, X, D_{min}, d_{min}, a_0, v_0, x_0, t, V_{max}, a_{max})$
Parametry:

- $a$ - minimalna odległość do sąsiada
- $X$ - zmienna celu
- $D_{min}$ - minimalna odległość do celu
- $d_{min}$ - macierz minimalnych odległość od sąsiada
- $a_0$ - początkowa odległość od sąsiada
- $v_0$ - początkowa prędkość
- $x_0$ - pozycja początkowa
- $t$ - czas
- $V_{max}$ - prędkość maksymalna
- $a_{max}$ - największa, dopuszczalna odległość od sąsiada

Funkcja zwracająca pozycje celu.
$functionPosition = GetMainTrgtPos(VrepAPI, ClientID)$
Parametry:

- $VrepAPI$ - struktura zawierająca wszystkie funkcje dostępne w V-rep
- $ClientID$ - identyfikator użytkownika

Funkcja zwracająca zakłóconą pozycje celu:
$function[dpos, pos] = GetDisturbedMainTrgtPosition(VrepAPI, ClientID, scale)$
Parametry:

- $VrepAPI$ - struktura zawierająca wszystkie funkcje dostępne w V-rep
- $ClientID$ - identyfikator użytkownika
- $scale$ - współczynnik zakłócenia

Funkcja ustawiająca drony w pozycjach początkowych:
$functionSetQuadTrgtPos(VrepAPI, ClientID, QuadNumber, Position)$
Parametry:

- $VrepAPI$ - struktura zawierająca wszystkie funkcje dostępne w V-rep

- $ClientID$ -identyfikator użytkownika
- $QuadNumber$ - numer drona dla którego aktualnie jest ustalana pozycja
- $Position$ - pozycja początkowa w układzie 2D

Funkcja zwracająca początkową pozycję dla każdego drona z zakłóceniem oraz bez zakłócenia:
$function[dpos, pos] = GetDisturbedQuadPosition(VrepAPI, ClientID, size, scale)$
Parametry:

- $VrepAPI$ - struktura zawierająca wszystkie funkcje dostępne w V-rep
- $ClientID$ - identyfikator użytkownika
- $size$ - liczba dronów
- $scale$ - współczynnik zakłócenia

Funkcja optymalizująca kolejne położenia dronów:
$function[a_x, a_y] = OptimizeNextMove(X, D, d, a_0, v_0, x_0, t, D_{min}, d_{min}, V_{max}, a_{max})$
Parametry:

- $X$ - zmienna celu
- $D$ - odległość do celu
- $d$ - odległość od sąsiada
- $a_0$ - początkowa odległość od sąsiada
- $v_0$ - początkowa prędkość
- $x_0$ - pozycja początkowa
- $t$ - czas
- $D_{min}$ - minimalna odległość do celu
- $d_{min}$ - minimalna odległość od sąsiada
- $V_{max}$ - prędkość maksymalna
- $a_{max}$ - największa, dopuszczalna odległość od sąsiada

Funkcja zwracająca optymalną pozycje każdego z dronów oraz gradient sumaryczny zawierający gradient odległości od celu, gradient błędu utworzonej formacji oraz gradient prędkości poruszania się utworzonej formacji:
$function[out, grad] = TargetFunction(a, X, D, d, v_0, x_0, t)$
Parametry:

- $a$ - minimalna odległość do sąsiada
- $X$ - zmienna celu
- $D$ - zadana odległość od celu
- $d$ - zadana macierz odległości do sąsiada
- $v_0$ - początkowa prędkość
- $x_0$ - pozycja początkowa
- $t$ - czas