Projekt - Metody optymalizacji

Wyprowadzenie wzoru fukncji kosztu

$T(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$ - przybliżenie zależności ciągu silnika od położenia na trasie - wektor $a = [a_0, ...a_n]^T$ bedziemy optymalizować.

$$f = C_T + C_F$$

f - funkcja kosztu
C_T - koszt związany z czasem wykorzystania pojazdu
C_F - koszt związany ze zużyciem paliwa

$$C_T = t_cP_T$$

P_T - wartość jednostki cxasu wykorzystania pojazdu
t_c - całkowity czas trwania podróży

$$C_F = P_F \int_0^{x_c} F_c(T(x))\, dx$$

P_T - koszt jednostki paliwa
x_c - długość trasy
F_c - funkcja zużycia paliwa zależna siły ciągu $T(x)$ wyrażona w $\frac{k}{Nm}$
Dla uproszczenia przyjmujemy zależność (Zazwyczaj jest to zależność zbliżona do liniowej)

$$F_c(T(x)) = F_CT(x)$$

Równanie opisujące zależność mocy silnika od warunków zewnętrznych:

$$T(x) = D_a(v) + D_o(x)$$

$D_a(v)$ - zależność oporu aerodynamicznego od prędkości
$D_o(x)$ - zależność oporów ruchu związanych z uwarunkowaniem trasy
Przyjmujemy zazwyczaj $D_a(v) = Sv^2$ dla stałej pewnej S. Wtedy prędkość na pozycji x wynosi:

$$v = \sqrt{\frac{T(x)-D_o(x)}{S}}$$

Prędkość chwilowa jest pochodną pozycji po czasie:

$$v = \frac{dx}{dt}$$

Po przekształceniach otrzymujemy równanie różniczkowe:

$$\frac{dt}{dx} = \frac{\sqrt{S}}{\sqrt{T(x)-D_o(x)}}$$

I rozwiązujemy równanie:

$$t(0) + C = 0$$

Wtedy:

$$t_c = t(x_c) + C$$

Ostatecznie optymalizujemy:

$$f([a_0, ..., a_n]) = (t(x_c) + C)P_t + P_F F_C \int_0^{x_c} T(x)\,dx$$

Ograniczenia minimalizacji

a) siłą wypadkowa nie może być być ujemna:

$$T(x) - D_o(x) > 0$$

b) siła ciągu silnika nie może przekraczać pewnej maksymalnej wartości:

$$T(x) <= T_{max}$$

c) inne dodatkowe, których nie będę implementować, np.: prędkośc maksymalna albo minimalny ciąg cilnika.

Implementacja

Do implementacji funkcji kosztu używam kilku modułów z biblioteki *scipy*:

a) *quad* - całkowanie numeryczne do wyznaczenia C_F

b) *odeint* - moduł rozwiązujący numercznie równania różniczkowe potrzebny do policzenia całki nieoznaczonej, której rozwiązaniem jest $t(x) + C$

c) Do implementacji ograniczeń używam funkcji *minimize_scalar* z modułu *optimize_scalar*, żeby znaleźć odpowiednio minimum i maksimum funkcji $T(x)$ dla danego a.

d) Do optymalizacji funkcji kosztu korzystam z metodu *SLSQP* z modułu *minimize*. Okazuje się jednak, że nie jest to dobra metoda, gdyż wykonuje działania dla całej przestrzeni, co powoduje liczenie pierwiastka z liczby ujemnej.

Wnioski

Temat jest bardzo rozbudowany i bardziej dogłębne analizowanie go przekracza moją dotychczasową wiedzę zdobytą na studiach. W przyszłośći najprawdopodobniej wrócę do tego zagadnienia.