POLITECHNIKA WARSZAWSKA

DYSCYPLINA NAUKOWA INŻYNIERIA MECHANICZA

DZIEDZINA NAUK INŻYNIERYJNO TECHNICZNE

Rozprawa doktorska

mgr inż. Dariusz Miedziński

Optymalizacja procesu sterowania pociskiem rakietowym wyposażonym w gazodynamiczny moduł sterujący

Promotor

dr hab. inż. Robert Głębocki, prof. PW

WARSZAWA 2025

Podziękowania

Pragnę serdecznie podziękować mojemu promotorowi, dr hab. inż. Robertowi Głębockiemu, prof. PW, za cenne wskazówki, nadzór oraz dzielenie się wiedzą, które były nieocenionym wsparciem podczas realizacji niniejszej rozprawy.

Szczególne podziękowania kieruję do dr hab. inż. Radosława Pytlaka, prof. PW, za pomoc w zrozumieniu złożonych zagadnień optymalizacji numerycznej oraz wsparcie, które umożliwiło doprowadzenie pracy do końca.

Dziękuję również dr hab. inż. Janowi Kindrackiemu, prof. PW, oraz jego zespołowi za możliwość uczestniczenia w projekcie rakiety ORION, a także za niezapomniane doświadczenia związane z wyjazdami poligonowymi.

Wyrazy wdzięczności kieruję także do dr hab. inż. Piotra Lichoty, prof. PW, dr inż. Mariusza Jacewicza oraz mgr inż. Mateusza Sochackiego za zawsze okazywane wsparcie, gotowość do pomocy i możliwość licznych konsultacji.

Streszczenie

Temat pracy: Optymalizacja procesu sterowania pociskiem rakietowym wyposażonym w gazodynamiczny moduł sterujący.

Celem pracy było opracowanie algorytmu sterowania pociskiem rakietowym typu ziemia-ziemia wyposażonym w gazodynamiczny moduł sterujący, składający się z zestawu silników korekcyjnych na stały materiał pędny, który zapewniłby trafienie w cel naziemny o znanych współrzędnych z wymaganą dokładnością i wykorzystał w tym celu minimalną ilość energii. Algorytm ten miał opierać się o bazę danych rozwiązań uzyskanych w procesie optymalizacji, a sterowanie miało odbywać się w pętli otwartej z korekcją w ostatniej fazie lotu. W ramach prac opracowano matematyczny model rakiety, który został zweryfikowany oraz zwalidowany wykorzystując dane i wyniki testów lotnych opracowanej na PW rakiety RPT ORION. W celu rozwiązania postawionego zagadnienia opracowano specjalną procedurę optymalizacji. Układ sterowania uproszczono do postaci dwóch ciągłych parametrów sterujących. Optymalizację przeprowadzano z użyciem oprogramowania IPOPT. Sprawdzono działanie dwóch metod: bezpośredniej i pośredniej. W pierwszej metodzie zagadnienie to traktowano jako problem optymalizacji nieliniowej, a w drugiej gradient funkcji kosztu wyznaczono korzystając z teorii sterowania optymalnego pod postacią Zasady Maksimum Pontryagina. Niezbędne gradienty i jakobiany obliczano stosując metodę automatycznego różniczkowania wykorzystując pakiet ADOL-C. Uzyskane w procesie optymalizacji przebiegi czasowe parametrów sterujących zamieniano w procedurze dyskretyzacji na czasy i kierunki odpaleń silników korekcyjnych. Przeprowadzając te obliczenia dla wielu warunków początkowych opracowano bazę danych rozwiązań składającą się z sekwencji odpaleń silników korekcyjnych oraz odpowiadającej im trajektorii referencyjnej. Opracowano docelowy algorytm sterujący oraz przedstawiono wyniki jego działania. Potwierdzono, że pozwala on na znaczącą redukcję dyspersji miejsc upadku względem klasycznych metod sterowania.

Słowa kluczowe: symulacja, sterowanie optymalne, rakiety

Abstract

Thesis title: Optimization of control strategy for a gasodynamically controlled projectile.

The aim of the thesis was to develop a control algorithm for a surface-to-surface missile equipped with a gasodynamic control module consisting of a set of solid-propellant correction thrusters, that would ensure hitting a ground-based target with the required accuracy and using minimum amount of energy. The control approach was based on a database of solutions obtained through optimization and was implemented as open-loop control with final-phase trajectory correction. As part of the work, a mathematical model of the missile was developed, verified, and validated using data and results from flight tests of the RPT ORION missile developed at Warsaw University of Technology. To solve the posed problem, a dedicated optimization procedure was created. The control system was simplified to two continuous control parameters. The optimization was conducted using the IPOPT software. Two methods were examined: the direct method and the indirect method. In the first method, the problem was treated as a non-linear optimization task, while in the second, the cost function's gradient was derived using optimal control theory in the form of Pontryagin's Maximum Principle. Necessary gradients and Jacobians were computed using automatic differentiation with the ADOL-C library. The time profiles of the control parameters obtained through optimization were discretized into ignition times and directions for the correction thrusters. By performing these calculations for various initial conditions, a solution database was created consisting of thruster's firing sequences and corresponding reference trajectories. The final control algorithm was developed and its performance results were presented. It was confirmed that the algorithm significantly reduces the dispersion of impact points compared to classical control methods.

Key words: simulation, optimal control, missiles

Spis treści

Wykaz skrótów i oznaczeń

CAD	Computer Aided Design
CEP	Circular Error Probable
HIL	Hardware-In-the-Loop
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISA	International Standard Atmosphere
LOS	Line Of Sight
MIL	Model-In-the-Loop
NLP	Non-Linear Programming
PIL	Processor-In-the-Loop
PNG	Proportional Navigation Guidance
PW	Politechnika Warszawska
RPT	Rakietowa Platforma Testowa
SIL	Software-In-the-Loop
WGS-84	World Geodetic System 1984
ZMP	Zasada Maksimum Pontryagina

$\mathbf{1}$	Macierz jednostkowa
a	Promień Ziemi na równiku, [m]
a	Prędkość dźwięku, [m/s]
a_i, b_i, c_i	Współczynniki do wyznaczenia ciśnienia saturacji pary wodnej dla lodu
a_{th}	Progowa wartość przyspieszenia bocznego dla wysłania sygnału odpalenia silnika korekcyjnego, [m/s^2]
a_w, b_w, c_w	Współczynniki do wyznaczenia ciśnienia saturacji pary wodnej dla wody
$\mathbf{a}_{cg}$	Wektor przyspieszenia środka masy, [m/s^2]
$\mathbf{a}_{cmd}$	Zadany wektor przyspieszenia, [m/s^2]
$\mathbf{a}_{IMU}$	Wektor przyspieszenia punktu położenia czujnika inercjalnego, [m/s^2]
$\mathbf{a}_{meas}$	Wektor przyspieszenia zwracany przez model akcelerometru, [m/s^2]
A_e	Powierzchnia wylotowa dyszy, [m^2]
b_i	Błąd przesunięcia zera na osi $i = x, y, z$, [m/s^2] lub [rad/s]
c	Promień Ziemi na biegunie, [m]
c_{ij}	Współczynnik błędu sprzężeń skrośnych między osiami $i = x, y, z$ oraz $j = x, y, z$, [-]
C_l	Współczynnik momentu przechylającego, [-]
C_{l_0}	Współczynnik momentu przechylającego wymuszający obrót, [-]
C_{l_P}	Pochodna współczynnika momentu przechylającego po prędkości kątowej przechylania, [1/rad]

$C_{l_{\gamma\alpha}}$	Pochodna współczynnika momentu przechylającego wyższego rzędu, [1/rad^2]
C_m	Współczynnik momentu pochylającego, [-]
C_{m_α}	Pochodna współczynnika momentu pochylającego po kącie natarcia, [1/rad]
C_{m_Q}	Pochodna współczynnika momentu pochylającego po prędkości kątowej pochylania, [1/rad]
C_n	Współczynnik momentu odchylającego, [-]
$C_{n_{\gamma\alpha}}$	Pochodna współczynnika momentu odchylającego wyższego rzędu, [1/rad^2]
C_{N_Q}	Pochodna współczynnika siły normalnej po prędkości kątowej pochylania, [1/rad]
C_{N_α}	Pochodna współczynnika siły normalnej po kącie natarcia, [1/rad]
C_X	Współczynnik siły osiowej, [-]
$C_{X_{off}}$	Współczynnik siły osiowej przy niepracującym silniku głównym, [-]
$C_{X_{on}}$	Współczynnik siły osiowej przy pracującym silniku głównym, [-]
$C_{X_{\alpha^2}}$	Pochodna współczynnika siły osiowej po kwadracie całkowitego kąta natarcia, [1/rad^2]
C_Y	Współczynnik siły bocznej, [-]
$C_{Y_{\gamma\alpha}}$	Pochodna siły bocznej wyższego rzędu, [1/rad^2]
C_Z	Współczynnik siły normalnej, [-]
$\mathbf{C}_B^R$	Macierz transformacji z układu rakiety (B) do układu wyrzutni (R)
$\mathbf{C}_{LOS}^N$	Macierz transformacji układu związanego z linią wizowania (LOS) do układu nawigacyjnego (N)
$\mathbf{C}_N^B$	Macierz transformacji z układu nawigacyjnego (N) do układu rakiety (B)
$\mathbf{C}_{SK_i}^B$	Macierz transformacji z układu i-tego silnika korekcyjnego (SK_i) do układu rakiety (B)
$\mathbf{C}_T^B$	Macierz transformacji z układu silnika głównego (T) do układu rakiety (B)
d_{ref}	Średnica rakiety, [m]
$\mathbf{f}(\dots)$	Ciągła funkcja prawych stron
$\hat{\mathbf{f}}(\dots)$	Ciągła funkcja prawych stron dla znormalizowanego czasu
$\mathbf{f}_0$	Wektor obciążeń czynnych
$\mathbf{f}_C$	Wektor obciążeń reakcji od wyrzutni
$\hat{\mathbf{F}}(\dots)$	Dyskretna funkcja prawych stron
F_T	Siła ciągu silnika głównego, [N]
$F_{T_{sk_i}}$	Siła ciągu i-tego silnika korekcyjnego, [N]
$\mathbf{F}_A$	Wektor sił aerodynamicznych, [N]
$\mathbf{F}_B$	Wektor sił działających na rakietę, [N]
$\mathbf{F}_G$	Wektor sił grawitacji, [N]

$\mathbf{F}_{SK}$	Wektor sił od układu sterowania, [N]
$\mathbf{F}_T$	Wektor sił ciągu silnika głównego, [N]
g_0	Przyspieszenie grawitacyjne zgodne z ISA, [m/s^2]
$g_0(\phi)$	Przyspieszenie grawitacyjne w funkcji szerokości geograficznej, [m/s^2]
g_{acc}	Przyspieszenie grawitacyjne, [m/s^2]
$\mathbf{g}$	Wektor przyspieszenia grawitacyjnego, [m/s^2]
$\mathbf{G}$	Macierz uwzględniająca wpływ przyspieszeń na wartość wskazań giroskopu, [rad·s/m]
h	Wysokość, [m]
h_{geo}	Wysokość geopotencjalna, [m]
I_c	Impuls całkowity silnika głównego, [Ns]
$I_{c_{sk_i}}$	Impuls całkowity i-tego silnika korekcyjnego, [Ns]
$I_{ii_{sk}}$	Moment bezwładności główny silnika korekcyjnego względem osi $i = x, y, z$, [kg·m^2]
I_{ij}	Moment bezwładności względem osi i, gdy ciało obracane jest względem osi j, $(i, j = x, y, z)$, [kg·m^2]
$\mathbf{I}$	Macierz momentów bezwładności, [kg·m^2]
$\mathbf{I}_0$	Macierz momentów bezwładności rakiety w chwili startu, [kg·m^2]
$\mathbf{I}_k$	Macierz momentów bezwładności rakiety po końcu pracy silnika głównego, [kg·m^2]
$\mathbf{I}_{sk_i}$	Macierz momentów bezwładności i-tego silnika korekcyjnego, [kg·m^2]
$\mathbf{I}_R$	Macierz momentów bezwładności rakiety, [kg·m^2]
J, J_1, J_2, J_d	Funkcja kosztu
$\mathbf{J}$	Jakobian funkcji więzów nieholonomicznych
k	Numer kroku czasowego
L	Koszt ciągły
L	Dolna granica danych statystycznych
L_B	Składowa podłużna momentu działającego na rakietę, [Nm]
$\mathcal{L}$	Lagrangian
m	Masa rakiety, [kg]
m_0	Masa rakiety w chwili startu, [kg]
m_k	Masa rakiety po końcu pracy silnika głównego, [kg]
m_{sk_i}	Masa materiału pędnego i-tego silnika korekcyjnego, [kg]
m_p	Masa materiału pędnego silnika głównego, [kg]
M	Liczba parametrów sterujących w jednej płaszczyźnie
M_B	Składowa boczna momentu działającego na rakietę, [Nm]
Ma	Liczba Macha, [-]
$\mathbf{M}$	Macierz masowa
$\mathbf{M}_A$	Wektor momentów aerodynamicznych, [Nm]
$\mathbf{M}_B$	Wektor momentów działających na rakietę, [Nm]

$\mathbf{M}_{SK}$	Wektor momentów od układu sterowania, [Nm]
$\mathbf{M}_T$	Wektor momentów od siły ciągu silnika głównego, [Nm]
N	Stała nawigacji proporcjonalnej
N	Liczba parametrów optymalizacyjnych
N	Liczba kroków czasowych
N_B	Składowa pionowa momentu działającego na rakietę, [Nm]
N_{sk}	Liczba silników korekcyjnych
$\mathbf{n}_{cmd}$	Zadane przyspieszenie normalne do linii wizowania, [m/s^2]
p	Ciśnienie, [Pa]
p_0	Ciśnienie w miejscu startu rakiety, [Pa]
p_e	Ciśnienie odniesienia dla ciągu silnika głównego, [Pa]
p_s	Ciśnienie pary wodnej w stanie saturacji, [Pa]
p_v	Ciśnienie cząstkowe pary wodnej, [Pa]
P	Prędkość kątowa przechylania, [deg/s]
P_{eng}	Flaga oznaczająca działanie silnika głównego
q_i	Składowa $i = 0, 1, 2, 3$ kwaternionu. [-]
q_{dyn}	Ciśnienie dynamiczne, [Pa]
$\mathbf{q}$	Kwaternion, [-]
Q	Prędkość kątowa pochylania, [deg/s]
Q_1, Q_3	Pierwszy i trzeci kwartyl
$r_e(\phi)$	Odległość punktu na szerokości geograficznej ϕ od środka Ziemi, [m]
R	Prędkość kątowa odchylania, [deg/s]
R	Stała gazowa mieszaniny powietrza i pary wodnej, [J/molK]
R_0	Stała gazowa powietrza, [J/molK]
RH	Względna wilgotność powietrza, [-]
$\mathbf{r}_{aero}$	Wektor położenia punktu referencyjnego momentów aerodynamicznych, [m]
$\mathbf{r}_b$	Wektor położenia tylnego ślizgacza względem układu (B), [m]
$\mathbf{r}_{cg}$	Wektor położenia środka ciężkości, [m]
$\mathbf{r}_{cg_0}$	Wektor położenia środka ciężkości rakiety w chwili startu, [m]
$\mathbf{r}_{cg_k}$	Wektor położenia środka ciężkości rakiety w momencie końca pracy silnika głównego, [m]
$\mathbf{r}_{cg_{sk_i}}$	Wektor położenia środka ciężkości i-tego silnika korekcyjnego, [m]
$\mathbf{r}_{cg_R}$	Wektor położenia środka ciężkości rakiety, [m]
$\mathbf{r}_{cg_{w_i}}$	Wektor położenia środka ciężkości i-tego silnika korekcyjnego względem środka ciężkości rakiety, [m]
$\mathbf{r}_{IMU}$	Wektor położenia czujnika inercjalnego, [m]
$\mathbf{R}$	Wektor położenia rakiety w układzie nawigacyjnym, [m]
$\mathbf{R}_{TM}$	Wektor względnego położenia celu w układzie nawigacyjnym, [m]

$\mathcal{R}$	Funkcja więzów nieholonomicznych
s	Zmienna zespolona w dziedzinie Laplace'a
s_i	Błąd skali na osi $i = x, y, z$, [-]
S_{ref}	Powierzchnia referencyjna, [m^2]
t	Czas, [s]
$t_0, t_1, \ldots, t_f$	Granice przedziałów całkowania przy dyskretyzacji, [s]
t_{0_i}	Czas odpalenia i-tego silnika korekcyjnego, [s]
t_g	Progowy czas zadziałania układu sterowania, [s]
t_{last}	Czas od uruchomiania ostatniego silnika korekcyjnego, [s]
t_{sk_i}	Czas odpalenia i-tego silnika korekcyjnego po dyskretyzacji, [s]
$\mathbf{t}_{sk}$	Wektor czasów odpalenia silników korekcyjnych, [s]
T	Temperatura, [K]
T_0	Temperatura w miejscu startu rakiety, [K]
T_0	Początkowy czas symulacji, [s]
T_{eng}	Stała czasowa modelu pierwszego rzędu wygładzająca współczynnik oporu, [s]
T_f	Czas końcowy symulacji, [s]
T_s	Czas próbkowania, [s]
u_A	Prędkość podłużna rakiety względem powietrza, [m/s]
u_V, u_H	Parametr sterujący w płaszczyźnie pionowej (V) i poziomej (H), [-]
$\mathbf{u}$	Wektor sterowań
U	Prędkość podłużna, [m/s]
U	Górna granica danych statystycznych
U_1, U_2	Parametry sterujące, [-]
v_A	Prędkość boczna rakiety względem powietrza, [m/s]
v_w	Prędkość wiatru, [m/s]
V	Prędkość boczna, [m/s]
$\mathbf{v}$	Wektor prędkości liniowych, [m/s]
$\mathbf{v}_A$	Wektor prędkości rakiety względem powietrza, [m/s]
$\mathbf{v}_{N_{CG}}$	Wektor prędkości środka masy rakiety w układzie nawigacyjnym, [m/s]
$\mathbf{v}_{N_{IMU}}$	Wektor prędkości punktu zamocowania czujników inercjalnych w układzie nawigacyjnym, [m/s]
$\mathbf{v}_R$	Wektor prędkości liniowych miejsca styku tylnego ślizgacza z wyrzutnią, [m/s]
$\mathbf{v}_w$	Wektor prędkości wiatru w układzie nawigacyjnym, [m/s]
$V_{c_{ij}}$	Prędkość zbliżania w płaszczyźnie ij układu nawigacyjnego, [m/s]
$\mathbf{V}_c$	Wektor prędkości zbliżania, [m/s]
w_A	Prędkość pionowa rakiety względem powietrza, [m/s]
W	Prędkość pionowa
x_{cg}	Składowa podłużna środka ciężkości, [m]

x_{cg_0}	Składowa podłużna środka ciężkości w chwili startu rakiety, [m]
x_{cg_k}	Składowa podłużna środka ciężkości po końcu pracy silnika głównego, [m]
$\mathbf{x}$	Wektor zmiennych stanu
$\mathbf{x}_0$	Warunki początkowe wektora zmiennych stanu
X	Składowa położenia rakiety w kierunku północnym, [m]
X_B	Składowa podłużna siły działającej na rakietę, [N]
X_{TM}	Składowa podłużna względnego położenia celu, [m]
$\mathbf{X}_{cmd}$	Wektor współrzędnych celu, [m]
y_{cg}	Składowa boczna środka ciężkości, [m]
y_{cg_0}	Składowa boczna środka ciężkości w chwili startu rakiety, [m]
y_{cg_k}	Składowa boczna środka ciężkości po końcu pracy silnika głównego, [m]
Y	Składowa położenia rakiety w kierunku wschodnim, [m]
Y_B	Składowa boczna siły działającej na rakietę, [N]
Y_{TM}	Składowa boczna względnego położenia celu, [m]
z_{cg}	Składowa pionowa środka ciężkości, [m]
z_{cg_0}	Składowa pionowa środka ciężkości w chwili startu rakiety, [m]
z_{cg_k}	Składowa pionowa środka ciężkości po końcu pracy silnika głównego, [m]
Z	Składowa położenia rakiety w kierunku "w dół", [m]
Z_B	Składowa pionowa siły działającej na rakietę, [N]
Z_{TM}	Składowa pionowa względnego położenia celu, [m]
α	Kąt natarcia, [deg]
α_T	Całkowity kąt natarcia, [deg]
β	Kąt ślizgu, [deg]
γ	Faza błędu kierunku odpalenia silnika korekcyjnego, [deg]
γ_t	Wartość progowa kąta fazy błędu γ, [deg]
Θ	Kąt pochylenia, [deg]
Θ_g	Wartość progowa kąta pochylenia dla zadziałania algorytmu sterowania, [deg]
Θ_T	Kąt pochylenia siły ciągu, [deg]
λ_{ij}	Kąt linii wizowania z płaszczyzną ij układu nawigacyjnego, [deg]
$\boldsymbol{\lambda}$	Wektor mnożników Lagrange'a
$\boldsymbol{\lambda}$	Wektor kątów linii wizowania z płaszczyznami układu nawigacyjnego, [rad]
ξ_a	Współczynnik tłumienia akcelerometru, [-]
ξ_g	Współczynnik tłumienia giroskopu, [-]
ρ	Gęstość powietrza, [kg/m^3]
σ_a	Szum modelu akcelerometrów, [m/s^2]
σ_g	Szum modelu giroskopów, [rad/s]
τ	Minimalny czas między uruchomieniami kolejnych silników korekcyjnych, [s]

τ	Bezwymiarowy czas, [-]
τ_d	Opóźnienie zadziałania spłonki silnika korekcyjnego, [s]
τ_{sk}	Czas, po którym silnik korekcyjny wykorzysta połowę swojej energii, [s]
ϕ	Szerokość geograficzna, [deg]
ϕ_{sk_i}	Kąt przechylenia, na którym znajduje się i-ty silnik korekcyjny, [deg]
ϕ_{sk_i}	Kierunek odpalenia i-tego silnika korekcyjnego po dyskretyzacji, [deg]
ϕ_T	Aerodynamiczny kąt przechylenia, [deg]
ϕ_u	Kierunek działania ciągu silnika korekcyjnego, [deg]
$\boldsymbol{\phi}_{sk}$	Wektor kierunków odpalenia silników korekcyjnych, [deg]
Φ	Kąt przechylenia, [deg]
Φ	Koszt końcowy
Φ_T	Kąt przechylenia siły ciągu, [deg]
ψ_{sk_i}	Kątowe położenie i-tego silnika korekcyjnego na obwodzie rakiety, [deg]
ψ_w	Kierunek meteorologiczny wiatru, [deg]
Ψ	Kąt odchylenia, [deg]
ω_{n_a}	Częstość drgań własnych akcelerometru, [rad/s]
ω_{n_g}	Częstość drgań własnych giroskopu, [rad/s]
$\boldsymbol{\omega}$	Wektor prędkości kątowych, [deg/s]
$\boldsymbol{\omega}_{meas}$	Wektor prędkości kątowych zwracanych przez model giroskopu, [rad/s]
$\boldsymbol{\Omega}$	Macierz prędkości kątowych służąca do propagacji kwaternionu, [deg/s]

1 Wstęp

We współczesnych działaniach wojennych widać bardzo duży udział wszelkiego rodzaju rakiet, których celem jest zwalczanie stacjonarnych i ruchomych obiektów przeciwnika na krótkich, średnich, czy bardzo dużych dystansach. Rakiety można generalnie podzielić na sterowane i niesterowane [1]. Do tej pierwszej kategorii najczęściej zalicza się rakiety przeciwlotnicze (ziemia-powietrze) oraz walki powietrznej (powietrze-powietrze). Rakiety ziemia-ziemia oraz powietrze-ziemia mogą generalnie należeć do obu kategorii. Głównym problemem rakiet niesterowanych typu ziemia-ziemia jest celność. Losowość warunków atmosferycznych w czasie lotu, a także niedokładności wykonania każdego pocisku, czy prowadnic startowych, powodują, że nawet przy dokładnym celowaniu i ustawieniu wyrzutni w odpowiednim kierunku odległość uderzenia rakiety od celu może wynosić, w zależności od jej zasięgu, nawet kilkaset metrów. Z tego względu rakiety takie wysyłane są w salwach wynoszących do ponad stu egzemplarzy. Taka ilość znacznie zwiększa koszt operacji wojskowej, a także, ze względu na duży rozrzut miejsc upadku, może powodować bardzo dużo zniszczeń ubocznych. Użycie rakiet sterowanych powoduje, że możliwe staje się precyzyjne rażenie celu, lub co najmniej wielokrotne zmniejszenie rozrzutu, co znacząco zmniejsza ilość niezbędnych do wystrzelenia rakiet pozwalających na spełnienie celu misji. Odciąża to także logistykę związaną z transportem rakiet do miejsca działań, a także powoduje zmniejszenie strat ubocznych w postaci zniszczonej infrastruktury cywilnej czy strat wśród ludności. Wyposażenie pocisku w układ nawigacji i sterowania zwiększa jednak koszty produkcji pojedynczego egzemplarza. Ważne jest więc, aby móc zaprojektować taki system w sposób niedrogi i mało skomplikowany. W niniejszej pracy główny nacisk położony został na rakiety kierowane typu ziemia-ziemia wyposażone w gazodynamiczne moduły sterujące.

Systemy sterowania rakiet podzielić można na aerodynamiczne i gazodynamiczne [2, 3]. Zmianę trajektorii rakiety przy sterowaniu aerodynamicznym uzyskuje się poprzez wychylenia różnego rodzaju powierzchni nośnych, które mogą znajdować się w przedniej, tylnej lub środkowej części kadłuba. Zaletą takich systemów jest możliwość zmiany zarówno kierunku jak i wartości generowanej siły, co pozwala na dokładne odwzorowanie zadanych przez układ naprowadzania przyspieszeń i precyzyjne rażenie celu, co czyni je popularnie stosowanym rozwiązaniem. Istotną ich wadą jest natomiast ograniczone pasmo przenoszenia co przekłada się na trudności w realizacji szybkich i precyzyjnych manewrów, szczególnie w końcowej fazie lotu. Układy te charakteryzuje również zależność od prędkości i wysokości lotu rakiety. Aby dysponować odpowiednią efektywnością sterów rakieta musi poruszać się w atmosferze i z dość dużą prędkością. Wymagane są także układy wykonawcze sterowania odpowiedniej mocy, aby mogły się one przeciwstawić momentom zawiasowym na osiach sterów, a także odpowiednie dla nich źródła zasilania. Wszystkie te elementy wpływają oczywiście na zwiększenie kosztu, masy oraz skompli-

kowania całego układu. Sterowanie gazodynamiczne można podzielić na wektorowanie ciągu oraz silniki korekcyjne. Pierwszy typ wymaga działającego silnika głównego, który w przypadku niektórych rakiet może pracować tylko przez kilka pierwszych sekund lotu. Wymaga to także skomplikowanych układów wykonawczych, które muszą powodować odchylenie strug gazów wylotowych od osi rakiety. Drugi typ stanowią układy składające się z silników korekcyjnych, najczęściej wykorzystujących stały materiał pędny. Układy takie nie podlegają ograniczeniom typowym dla sterów aerodynamicznych. Mogą one działać w szerokim zakresie prędkości lotu, zapewniając większą swobodę manewrowania, szczególnie w zakresie bardzo małych prędkości, gdzie sterowanie aerodynamiczne nie może być wykorzystywane. Możliwość uzyskania natychmiastowych impulsów sterujących znacząco poprawia zdolność korygowania trajektorii rakiety w krótkim czasie. Ich zaletą jest także prostota wykonania oraz możliwość działania poza atmosferą. Układy takie głównie wykorzystywane są do zmiany orientacji rakiety zaraz po wyjściu z wyrzutni lub sterowania trajektorią rakiety w górnych partiach atmosfery, gdzie powierzchnie nośne tracą efektywność. Główną wadą tych układów, w zastosowaniu do rakiet typu ziemia-ziemia, jest natomiast możliwość kontrolowania tylko czasu i kierunku działania poszczególnych silników, bez możliwości wpływu na wartość generowanej siły. Powoduje to trudność w odpracowaniu przyspieszeń zadanych przez układ naprowadzania i tym samym problem celności trafienia. Istnieje więc potrzeba opracowania algorytmu sterowania takim układem, zwiększającego celność, aby wykorzystać jego zalety względem układów aerodynamicznych.

Na Politechnice Warszawskiej (PW) od wielu lat rozwijane są technologie rakietowe. Projektowane są zarówno konstrukcje mechaniczne, układy sterowania, a także tworzone są dokładne modele matematyczne wraz z systemami naprowadzania i sterowania. Zespół Politechniki Warszawskiej dysponuje również dużym doświadczeniem planowania i przeprowadzania testów lotnych, a także agregacji i dokładnej analizy wyników strzelań. W ostatnich latach, w ramach projektu badawczego finansowanego przez Narodowe Centrum Badań i Rozwoju, projektowany był gazodynamiczny układ sterowania dla rakiety kalibru 122 mm wraz z algorytmami nawigacji i sterowania, który mógłby być zastosowany na dotychczas niesterowanych pociskach tego kalibru, które należą do wyposażenia Wojska Polskiego. Aby wpasować się w rozwój kompetencji PW w zakresie technologii rakietowych z naciskiem na gazodynamiczne układy sterowania, za cel niniejszej pracy postawiono opracowanie algorytmu sterowania pociskiem rakietowym typu ziemia-ziemia wyposażonym w gazodynamiczny moduł sterujący składający się z zestawu rakietowych silników korekcyjnych na stały materiał pędny. Algorytm ten miałby za zadanie minimalizację błędu trafienia w cel naziemny o znanych współrzędnych geograficznych, a także minimalizację zużycia energii w układzie sterowania, co przekłada się na ilość użytych silników korekcyjnych. Efektywność wykorzystania układu wpływa na masę rakiety. Zmniejszenie ilości niezbędnych silników, zachowując dokładność trafienia, pozwolić może na zwięk-

szenie zasięgu rakiety lub zwiększenie masy głowicy bojowej. Ze względu na impulsowy charakter układu sterowania nie liczyło się bezpośrednie trafienie w cel, lecz zmniejszenie rozrzutu miejsc upadku w stosunku do rakiet niesterowanych oraz sterowanych z użyciem klasycznych algorytmów sterujących, takich jak nawigacja proporcjonalna lub podążanie za trajektorią referencyjną. Opracowany algorytm wykorzystuje metody optymalizacji, aby wyznaczyć czasy i kierunki odpalenia poszczególnych silników korekcyjnych realizując cel minimalizacji. Aby możliwe było zastosowanie metod optymalizacji dedykowanych do układów ciągłych, w celu znalezienia rozwiązania dla impulsowego układu sterowania, niezbędne było podzielenie jej na dwie części. Najpierw układ impulsowy zamieniono na układ ciągły otrzymując ciągłą funkcję sterującą, która podlegała optymalizacji, a następnie jej wynik dyskretyzowano, w celu uzyskania kierunków i czasów odpaleń poszczególnych silników korekcyjnych. Mając dany model matematyczny rakiety uzyskuje się w ten sposób sterowanie w pętli otwartej. Przeprowadzając takie obliczenia dla wielu warunków początkowych otrzymuje się bazę danych składającą się z wynikowych odpaleń silników oraz odpowiadających im trajektorii referencyjnych. Aby zamknąć pętle sprzężenia zwrotnego w docelowym algorytmie sterowania, na podstawie aktualnego stanu rakiety, wyszukuje się w bazie danych rozwiązania najlepiej odpowiadające obecnym warunkom lotu. Aby zwiększyć dokładność trafienia w końcowej fazie lotu wykorzystuje się dodatkowo jedną z klasycznych metod sterowania, która kosztem nieznacznego zwiększenia zużytej energii, zmniejszy błąd trafienia wynikający z różnic między użytą trajektorią referencyjną z bazy danych, a faktyczną trajektorią lotu rakiety.

Struktura pracy została podzielona na następujące części. Rozdział 2 przedstawia przegląd literatury skupiający się na metodach optymalizacji i sterowania optymalnego, a także dotychczasowych podejściach do naprowadzania rakiet ziemia-ziemia na cele stacjonarne, szczególnie wykorzystujących gazodynamiczne układy sterowania. Rozdział 3 wyjaśnia problem badawczy będący celem pracy oraz przedstawia metodykę podejścia do jego rozwiązania. Model matematyczny i symulacyjny rakiety opisane zostały w rozdziale 4, a ich weryfikacja i walidacja bazująca na danych rzeczywistej rakiety oraz wynikach jej testów lotnych przedstawiona została w rozdziale 5. Rozdział 6 poświęcony został metodom optymalizacji. Opisze on wykorzystane oprogramowanie oraz sformułuje problem optymalizacyjny i metody jego rozwiązania. W rozdziale 7 przedstawione zostanie podejście do transformacji wyników optymalizacji na czasy i kierunki odpaleń silników korekcyjnych. Docelowy algorytm sterowania, wraz z metodyką generacji bazy danych, został opisany w rozdziale 8, gdzie przedstawione zostały również wyniki jego działania. Na końcu pracy znajduje się podsumowanie przeprowadzonych badań wraz z wnioskami i planami na przyszłość.

2 Przegląd dotychczasowych rozwiązań

Gazodynamiczne układy sterujące wykorzystujące silniki korekcyjne na stały materiał pędny są często wykorzystywane w końcowej fazie lotu rakiet ze względu na ich szybką odpowiedź, co powoduje zwiększenie niezbędnej w tej fazie lotu manewrowości. Systemy takie podzielić można na te wykorzystujące jednorazowego użytku silniczki rakietowe oraz zestawy wielu dysz umożliwiających sterowanie w sposób ciągły. Układy takie są często wykorzystywane w rakietach przeciwlotniczych. Rakiety PAC-3 amerykańskiego systemu Patriot wykorzystują 180 silników rakietowych do podsterowania w końcowej fazie lotu. Silniczki takie pracując przez setne części sekundy są w stanie wytworzyć duży ciąg pozwalający na szybką korektę trajektorii lotu rakiety. Dzięki temu rakiety te są w stanie niszczyć cele powietrzne bezpośrednim trafieniem. Podobny układ znajduje się także w rakietach SM-3 oraz ukraińskich rakietach systemu Wilcha. Układ ciągły wykorzystujący 8 dysz znajduje się w rakietach amerykańskiego systemu THAAD. Układ taki jest bardziej precyzyjny, gdyż pozwala także na sterowanie wartością generowanej siły, lecz jest dużo droższy i bardziej skomplikowany. Układy gazodynamiczne wykorzystuje się także w rakietach przeciwpancernych oraz moździerzowych. Przykładem takich konstrukcji są szwedzki pocisk Strix, wykorzystujący 12 silników korekcyjnych umieszczonych w pobliżu środka ciężkości, oraz amerykańskie pociski zestawu Dragon, wyposażone w 60 silników korekcyjnych. Innym zastosowaniem gazodynamicznych układów sterujących nieanalizowanym w tej pracy jest obrót rakiety zaraz po wyjściu z wyrzutni, zwany procedurą pionowego zimnego lub miękkiego startu. Polega ona na wypchnięciu rakiety z wyrzutni, a następnie wykorzystaniu silników korekcyjnych do zmiany jej orientacji, przed uruchomieniem silnika głównego, co ze względu na małą prędkość pocisku uniemożliwia użycia powierzchni aerodynamicznych. Procedurę taką wykorzystują brytyjskie rakiety z rodziny CAMM, oraz koreańskie rakiety Cheolmae-2.

Wyposażenie w układ zawierający silniki korekcyjne na stały materiał pędny rakiet typu ziemia-ziemia 122 mm oraz w nisko-kosztowe układy i czujniki nawigacyjne, może pozwolić na znaczącą redukcję dyspersji miejsc upadku. Opracowanie prostego autopilota opartego o tanie czujniki typu MEMS opisano w [4]. Model sterowania gazodynamicznego w zastosowaniu do bomb lotniczych przedstawiono w [5, 6]. Istotnym aspektem jest także zbadanie wpływu działania takiego układu na charakterystyki aerodynamiczne rakiety. Badanie efektów odpaleń silników korekcyjnych na opływ rakiety przedstawiono w [7, 8]. Do klasycznych metod sterowania rakietami wyposażonymi w gazodynamiczne układy sterujące oparte o silniki korekcyjne należą nawigacja proporcjonalna [9] oraz podążanie za trajektorią referencyjną [10, 11, 12]. Powstały także prace wykorzystujące głowice wizyjne pozwalające na wypracowanie komend korektujących trajektorię [13]. Autorzy [14] opracowali natomiast pracę, gdzie porównano w drodze symulacji różne metody naprowadzania. Kombinację predykcji punktu upadku z podążaniem za trajektorią referencyjną

można znaleźć w [15, 16], a z wariantami nawigacji proporcjonalnej w [17]. Wykorzystanie algorytmu genetycznego oraz kontrolera rozmytego opisano w [18]. Kolejną klasą metod są algorytmy wykorzystujące optymalizację oraz teorię sterowania optymalnego.

Problem sterowania optymalnego polega na znalezieniu takich sterowań $u(t)$ zależnych od czasu, które pozwolą układowi dynamicznemu opisanemu zestawem równań różniczkowych pierwszego rzędu zminimalizować pewien wskaźnik jakości, spełniając jednocześnie pewne ograniczenia narzucone na stan układu lub sterowanie. Opis różnych problemów sterowania optymalnego w lotnictwie i kosmonautyce przedstawił autor [19]. Istnieje wiele metod pozwalających na rozwiązanie tego problemu [20, 21, 22]. Do głównych kategorii można zaliczyć metody pośrednie, bezpośrednie, programowanie dynamiczne, uczenie ze wzmocnieniem oraz sterowanie predykcyjne.

W metodzie pośredniej warunki konieczne optymalności wyprowadzane są z zasad rachunku wariacyjnego lub Zasady Maksimum Pontryagina (ZMP) [23]. Prowadzą one do układu równań różniczkowych na zmienne stanu, z warunkiem początkowym, i na zmienne sprzężone, z warunkiem końcowym, które tworzą tak zwane zagadnienie dwubrzegowe. Do typowych metod pośrednich zaliczyć można metodę strzału (ang. shooting) lub metodę gradientów zredukowanych (ang. reduced gradients). Wykorzystanie ZMP i nieciągłej metody Galerkina do rozwiązania problemu optymalnego przedstawiono natomiast w [24]. Metoda bezpośrednia polega na dyskretyzacji całego problemu, lub tylko sterowań, przekształcając go w problem programowania nieliniowego, czy nieliniowej optymalizacji. Do klasycznych metod bezpośrednich zaliczyć można na przykład kolokację.

Metoda strzału opiera się na bezpośrednim rozwiązywaniu warunków koniecznych optymalności, wynikających z ZMP. Cały problem sterowania zamieniany jest w zagadnienie początkowe z warunkami brzegowymi. Zgaduje się na starcie wartości początkowe zmiennych sprzężonych, dokonując "strzału", a następnie oba równania całkuje się w przód. Warunki brzegowe oraz tak zwane warunki transwersalności tworzą pewną funkcję nieliniową, zależną od rozwiązania problemu początkowego, której miejsce zerowe jest rozwiązaniem postawionego zagadnienia. Wartości sterowania otrzymuje się spełniając ZMP poprzez maksymalizację Hamiltonianu. Można wyróżnić metodę pojedynczego strzału (ang. single shooting) [25, 26, 27], gdzie całkowanie odbywa się po całym przedziale czasu oraz wielokrotnego strzału (ang. multiple shooting) [28, 29, 30], gdzie przedział czasu dzielony jest na podprzedziały, w każdym z nich rozwiązywane są zagadnienia początkowe, a wartości zmiennych na krańcach przedziałów dodawane są do warunków brzegowych. Metoda ta jest bardzo dokładna jeśli dysponuje się zbliżonymi do rozwiązania warunkami początkowymi, jest ona jednak dość skomplikowana w implementacji oraz kosztowna obliczeniowo, gdyż wymaga użycia dokładnych algorytmów całkujących [31] i rozwiązujących duże układy równań liniowych. Może być ona realizowana zarówno w wariancie pośrednim i bezpośrednim [32, 33]. W bezpośredniej metodzie strzału (ang. direct shooting) tylko sterowania podlegają parametryzacji i dyskretyzacji, a równanie stanu jest całko-

wane numerycznie [34]. Porównanie bezpośredniej kolokacji i metody strzału do generacji trajektorii optymalnych opisano w [35]. Wykorzystanie metody wielokrotnego strzału do rozwiązania problemu opisanego przez układ równań różniczkowo-algebraicznych pokazano w [36, 37, 38]. Wykorzystanie nieliniowego programowania sprzężonego z metodą wielokrotnego strzału przedstawia [39].

W metodzie gradientów zredukowanych eliminuje się zmienne stanu z parametrów optymalizacyjnych, jako w całości zależne od rozwiązania różniczkowego równania stanu oraz sterowania. Optymalizacja przebiega tylko po parametrach sterowania, w mniejszej przestrzeni, z czego wynika nazwa zredukowane. Całkując równania stanu przy danym sterowaniu otrzymuje się trajektorię, a następnie optymalizuje się sterowania aby zminimalizować zadany wskaźnik jakości. Gradient funkcji celu wyznaczany jest z użyciem zmiennych sprzężonych.

Jednym z podejść metody bezpośredniej jest zastosowanie metod kolokacji, które pozwalają przekształcić nieskończenie wymiarowy (ciągły w czasie) problem sterowania na skończony problem optymalizacji nieliniowej. Trajektorie wektora stanu i sterowania przybliżane są wielomianami, w taki sposób aby wymusić spełnienie równań dynamicznych w pewnych punktach zwanych punktami kolokacji. W ten sposób sprowadza się to zagadnienie do dużego układu równań algebraicznych stanowiących więzy w problemie optymalizacyjnym. Funkcję kosztu przybliża się odpowiednią kwadraturą, a cały problem rozwiązuje się z użyciem optymalizatora. Metody kolokacji można podzielić na lokalne i globalne [40]. W metodzie lokalnej czas dzielony jest na wiele małych przedziałów, gdzie w każdym z nich przybliża się lokalnie wektor stanu i sterowań wielomianami zwykle niskiego stopnia, a także wymusza się ciągłość problemu na granicach tych przedziałów. Charakteryzuje się ona prostą implementacją, lecz wymaga wielu punktów kolokacji w celu osiągnięcia dobrej dokładności rozwiązania. W metodzie globalnej, zwanej też pseudospektralną, czas traktowany jest jako jeden duży przedział, a wektory stanu i sterowania aproksymowane są przez wielomiany ortogonalne wysokiego stopnia, interpolowane w specjalnie wybranych węzłach. Pozwala ona na osiągnięcie bardzo dużej dokładności przy mniejszej liczbie punktów kolokacji. Zasadniczą różnicą pomiędzy różnymi metodami pseudospektralnymi jest wybór tych punktów. Opracowanie trajektorii optymalnej z wykorzystaniem kwadratury Gaussa oraz punktów Legendre-Gauss-Lobatto przedstawiono w [41, 42, 43, 44, 45]. Ta sama metoda uzupełniona o liniową teorię pocisku przedstawiona została w [46]. Podobny proces z wykorzystaniem punktów Legendre-Gaussa pokazano w [47]. Rozwiązanie nieliniowego problemu z więzami wykorzystując metodę pseudospektralną Chebysheva przedstawiono w [48, 49, 50, 51], metodę Legendre, z punktami Legendre-Gauss-Lobatto, w [52, 53], a metodę Jacobiego w [54]. Wykorzystanie punktów Legendre-Gauss-Radau do problemu z nieskończonym horyzontem czasowym opracowano w [55, 56]. Porównanie dwóch wariantów metody strzału i kolokacji pokazano w [57].

Innym podejściem do problemów optymalnego sterowania jest programowanie dy-

namiczne, oparte na zasadzie optymalności Bellmana, która mówi, że każda część trajektorii optymalnej od pewnego stanu pośredniego do stanu końcowego jest także optymalna. Na podstawie tej zasady można zdefiniować funkcję wartości $V(x,t)$, która opisuje minimalny koszt możliwy do uzyskania z danego stanu x w chwili t, aż do końca horyzontu sterowania. Funkcja wartości spełnia nieliniowe równanie cząstkowe znane jako równanie Hamiltona–Jacobiego–Bellmana (HJB) z narzuconym warunkiem końcowym. Rozwiązując to równanie, analitycznie lub numerycznie, można otrzymać funkcję $V(x,t)$, z której można bezpośrednio wyznaczyć optymalne sterowanie jako funkcję stanu. Metoda ta pozwala osiągnąć teoretycznie najbardziej dokładne rozwiązanie, jednak jej główną wadą jest tak zwana klątwa wymiarowości, co uniemożliwia jej zastosowanie do problemów o wymiarach większych niż 3-4. Wykorzystanie programowania dynamicznego do rozwiązania problemu optymalnego przechwytywania na potrzeby obrony przeciwlotniczej pokazano w [58, 59, 60]. Autorzy [61, 62] przedstawili wykorzystanie adaptacyjnego programowania dynamicznego do przechwycenia przez rakietę celu manewrującego. Połączenie podejścia HJB i metody strzału pokazano w [63]. Wykorzystanie równania HJB przy rozwiązywaniu problemów optymalnych inwestycji przedstawiono w [64, 65], a do wyznaczenia optymalnej trajektorii dla rakiety orbitalnej w [66].

Kolejną z metod zyskujących coraz większe zainteresowanie jest uczenie maszynowe, wykorzystujące sieci neuronowe, czy uczenie ze wzmocnieniem. Metody te pozwalają przybliżać rozwiązania problemów sterowania poprzez wypracowywanie odpowiednich polityk sterowania, bazując na uczeniu się modelu wykorzystując akcję i nagrodę. Wykorzystanie głębokiego uczenia ze wzmocnieniem oraz głębokich sieci neuronowych do identyfikacji modelu aerodynamicznego rakiety oraz do opracowanie kontrolera w kanale pochylania przedstawiono w [67]. Naprowadzanie rakiety na cel z użyciem uczenia ze wzmocnieniem pokazano w [68, 69, 70], a scenariusz odwrotny ze zwodzeniem rakiety przechwytującej w [71]. Algorytm ewolucyjny wraz z uczeniem maszynowym wykorzystano w [72] do optymalizacji trajektorii rakiety na stały materiał pędny. Porównanie metody nawigacji proporcjonalnej i uczenia ze wzmocnieniem do naprowadzania rakiety przedstawiono w [73].

Wyróżnia się także, metody oparte na sterowaniu predykcyjnym (ang. Model Predictive Control, MPC). Charakteryzują się one tym, że nie rozwiązują całego problemu na raz, tylko w sposób kroczący optymalizując sterowanie na krótkim horyzoncie i przesuwając go w czasie. Pozwala to, na implementację kontrolera w czasie rzeczywistym, lecz wymaga zazwyczaj uproszczonego, liniowego modelu obiektu. Wykorzystanie tej metody z użyciem zmodyfikowanej liniowej teorii pocisku przedstawia [74]. Dwie strategie predykcyjne połączone z liniową teorią pocisku opisano w [75]. Wykorzystanie metody MPC dla rakiet wyposażonych w silniki korekcyjne przedstawiono w [76, 77].

Istnieją także inne metody oparte o optymalizację. Wykorzystanie programowania całkowitoliczbowego (ang. Mixed-Integer Linear Programming) pozwalającego opra-

cować logikę sterującą przedstawione zostało w [78]. Wykorzystanie uogólnionej, jawnej metody naprowadzania wektorowego (ang. Generalized Vector Explicit Guidance) GENEX do sterowania rakietą przedstawiono w [79]. Wykorzystanie metod optymalizacji do kontroli kąta uderzenia w cel dla rakiety pokazano w [80]. Wykorzystanie rozwinięcia funkcyjnego Fliessa do opracowania logiki sterującej rakiety korzystającej z silników korekcyjnych opisano w [81]. Metoda optymalizacji zarówno czasu jak i kąta uruchomienia silników korekcyjnych została przedstawiona w [82]

W wielu przypadkach rozwiązywanie problemów sterowania optymalnego sprowadza się do rozwiązania nieliniowego problemu optymalizacji (NLP) z ograniczeniami. Do najbardziej znanych metod rozwiązywania takich problemów należą sekwencyjne programowanie kwadratowe (ang. Sequential Quadratic Programming (SQP)) [83, 84] oraz metoda punktu wewnętrznego (ang. Interior-Point (IP)) [85, 86], które wspierane są przez strategie doboru wielkości kroku optymalizacji takie jak liniowe poszukiwanie (ang. line search) [87, 88], region zaufania (ang. trust region) [89, 90] czy metoda filtra [91, 92]. Metoda SQP polega na kolejnych aproksymacjach problemu NLP za pomocą problemów programowania kwadratowego (QP), gdzie funkcja celu przybliżana jest macierzowym równaniem kwadratowym, a ograniczenia są linearyzowane [93]. Wykorzystanie tej metody do rozwiązania problemu sterowania optymalnego pokazano w [94], a zastosowaniu do sterowania rakiet w [95, 96]. W metodzie punktu wewnętrznego przekształca się problem z ograniczeniami nierównościowymi poprzez wprowadzenie tak zwanej funkcji bariery do funkcji celu. Rozwiązuje się następnie serie podproblemów, prowadzących do rozwiązania pierwotnego NLP. Wykorzystanie metody IP w zastosowaniu do sterowania rakiet przedstawiono w [97, 98]

W porównaniu do metod ogólnie stosowanych do wyznaczania optymalnego sterowania dla takiej klasy rakiety, prezentowana w niniejszej pracy metoda nie wymaga uproszczeń modelu symulacyjnego, poza układem sterowania. Możliwe jest wykorzystanie dowolnie skomplikowanego, nieliniowego modelu rakiety i środowiska, co pozwala uzyskiwać bardzo dokładne rozwiązania. Opracowana metoda bazuje na metodzie gradientów zredukowanych, wykorzystując ZMP i zmienne sprzężone, a wynikowy problem optymalizacyjny rozwiązywany jest metodą punktu wewnętrznego wraz ze strategią doboru długości kroku wykorzystującą metodę filtra. W odróżnieniu od kolokacji nie wymaga ona aproksymacji zmiennych wielomianami, a w przeciwieństwie do metody strzału nie wymaga dokładnej znajomości warunków początkowych. Nie może być ona zastosowana "online", jak kontrolery MPC, ale opracowana strategia interpolacji rozwiązań umożliwia opracowanie algorytmu sterowania pracującego w czasie lotu.

3 Problem badawczy

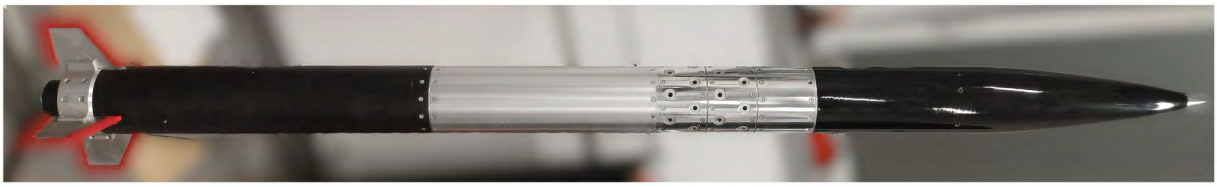
Celem niniejszej pracy jest opracowanie i implementacja metody sterowania pociskiem rakietowym typu ziemia-ziemia, wyposażonym w gazodynamiczny moduł sterujący, która pozwoli na zmniejszenie rozrzutu miejsc upadku i jednocześnie zapewni efektywne wykorzystanie dostępnej w układzie sterowania energii. Będące obecnie na wyposażeniu Sił Zbrojnych RP produkowane w Polsce niekierowane pociski rakietowe kalibru 122 milimetry M-21 FHD "FENIKS"(Rysunek 3.1) charakteryzują się rozrzutem przekraczającym 800 metrów, mierzonym jako promień okręgu zawierającego 50% miejsc upadku. Jak zaznaczono we wstępie wartość ta wynika nie tylko z braku sterowania, lecz również z niedokładności wykonania elementów każdej z rakiet, drgań wyrzutni podczas startu oraz losowości warunków atmosferycznych. Powoduje to bardzo duży wzrost zniszczeń pobocznych oraz koniecznych wystrzelenia wielu rakiet, aby zrealizować cel działań militarnych.

Rysunek 3.1: Pocisk rakietowy kalibru 122 mm M-21 FHD FENIKS

Aby ograniczyć wartość rozrzutu lub nawet zbliżyć się do możliwości precyzyjnego rażenia stacjonarnych obiektów naziemnych, niezbędne jest wyposażenie rakiety w układy nawigacji, naprowadzania i sterowania. Korzystając z zalet układów gazodynamicznych względem aerodynamicznych opisanych we wstępie oraz wykorzystując wiedzę i doświadczenie w zakresie takich układów obecne na Politechnice Warszawskiej, zdecydowano się na opracowanie algorytmu sterującego, wykorzystującego silniki korekcyjne na stały materiał pędny, który umożliwi spełnienie celu jakim jest zmniejszenie rozrzutu rakiety, a jednocześnie zapewni efektywne wykorzystanie dostępnej w układzie energii, które to pozwoli na oszacowanie niezbędnej ilości silników korekcyjnych w jakie należy wyposażyć rakietę, aby osiągnąć rozrzut o określonej wartości.

Aby móc opracować taki algorytm niezbędne jest posiadanie dokładnego modelu matematycznego docelowej rakiety, włączając w to informacje o efektywności modułu gazodynamicznego, a także posiadanie danych uzyskanych w trakcie testów lotnych pozwalających na weryfikację i walidację modelu. Ze względu na to, że informację takie opisujące

rakietę FENIKS nie są dostępne, na cele badań wykorzystano opracowaną na Politechnice Warszawskiej Rakietową Platformę Testową (RPT) ORION (Rysunek 3.2), która posiada gazodynamiczny moduł sterujący z 32 silnikami korekcyjnymi na stały materiał pędny.

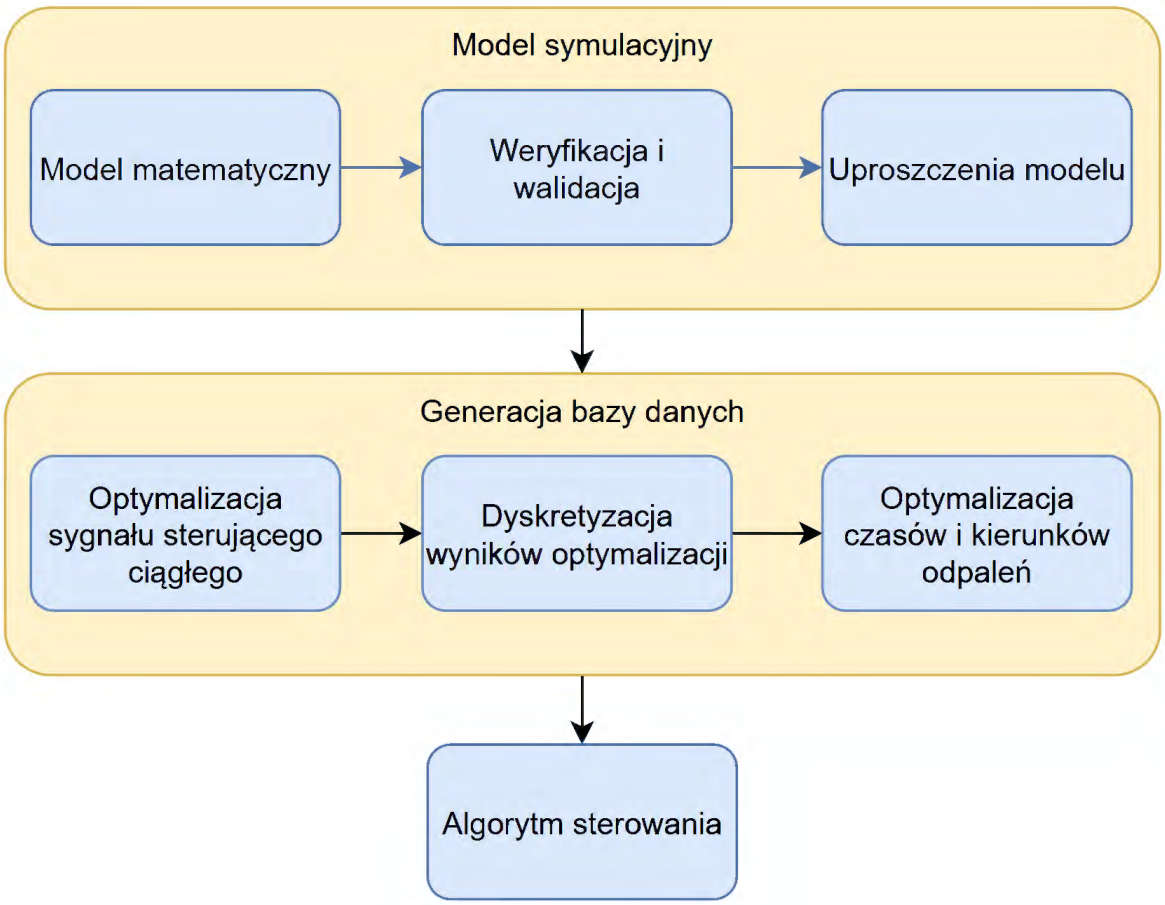


Rysunek 3.2: Rakietowa Platforma Testowa kalibru 122 mm ORION

Jest to rakieta kalibru 122 mm opracowana w celu rozwoju i testowania gazodynamicznych układów sterujących o zasięgu pozwalającym na przeprowadzanie prób lotnych na dostępnych w kraju poligonach. Jest ona krótsza niż rakieta Feniks o około 700 mm, a także porusza się z mniejszymi prędkościami. Startując z wyrzutni prowadnicowej rozpędza się do około 500 m/s, by po osiągnięciu apogeum korzystając z modułu gazodynamicznego naprowadzać się na cel naziemny o znanych współrzędnych. Moduł ten (Rysunek 3.3) składa się z 32 silników korekcyjnych na stały materiał pędny, które są rozłożone na obwodzie rakiety w 4 równoległych płaszczyznach po 8 w każdej. Każdy z silników może być uruchamiany impulsowo, co pozwala na generowanie krótkotrwałych sił bocznych korygujących trajektorię lotu rakiety. Sterowanie w fazie końcowej ma za zadanie precyzyjne naprowadzenie pocisku na cel. Celem działania takiego układu jest redukcja rozrzutu i zwiększenie dokładności trafienia.

Rysunek 3.3: Gazodynamiczny moduł sterujący rakiety RPT ORION

Głównym celem działania algorytmu sterowania jest wybór kierunku oraz czasu odpalenia każdego silnika korekcyjnego, w taki sposób aby zminimalizować błąd trafienia w cel naziemny oraz żeby wykorzystać w tym celu jak najmniejszą ilość silników. Minimalizacja obu tych warunków wymagała przedstawienia tego zagadnienia w postaci problemu optymalizacyjnego. Spełnienie drugiego z warunków, czyli minimalizację zużycia energii w układzie sterowania, uniemożliwiało bezpośredniego zadania jako zmiennych optymalizacyjnych czasów i kierunków odpaleń silników korekcyjnych, gdyż na starcie nie jest znana ich niezbędna ilość. Z tego względu zastosowano specjalną, procedurę optymalizacji, której wynikiem byłaby także minimalna ilość silników korekcyjnych zapewniająca trafienie rakiety w cel z wymaganą dokładnością. Procedura ta składała się z kilku części, których dokładny opis znajduje się w dalszej części pracy. Metodykę rozwiązania problemu badawczego można podzielić na następujące elementy, przedstawione również schematycznie na Rysunku 3.4:



Rysunek 3.4: Metodyka rozwiązania problemu badawczego

1. Opracowanie modelu matematycznego i symulacyjnego rakiety sterowanej wyposażonej w zestaw silników korekcyjnych na stały materiał pędny,

2. Weryfikacja i walidacja modelu wykorzystując dane rzeczywistej konstrukcji oraz

wyników jej testów lotnych,

3. Przygotowanie modelu matematycznego do obliczeń optymalizacyjnych stosując niezbędne uproszczenia,

4. Zamiana dyskretnych impulsów odpaleń silników korekcyjnych na ciągły sygnał sterujący,

5. Optymalizacja sygnału ciągłego w celu minimalizacji błędu trafienia oraz zużycia energii w układzie sterowania,

6. Zamiana optymalnego, ciągłego sygnału sterującego na dyskretne impulsy odpaleń silników korekcyjnych,

7. Optymalizacja czasu i kierunku odpalenia każdego z silników korekcyjnych.

8. Generacja bazy danych składającej się z trajektorii referencyjnych i odpowiadających im optymalnych rozwiązań,

9. Opracowanie docelowego algorytmu sterowania wykorzystującego bazę danych,

10. Oszacowanie wyników redukcji dyspersji miejsc upadku rakiety względem rakiet niesterowanych i sterowanych z użyciem klasycznych metod.

Pierwszym krokiem było opracowanie dokładnego modelu symulacyjnego rakiety sterowanej, który mógłby być wykorzystany do obliczeń optymalizacyjnych. Model ten pierwotnie powstał w środowisku MATLAB/Simulink, a następnie został przetłumaczony na język C++, czego wymagało wykorzystywane oprogramowanie do optymalizacji. Model ten został zweryfikowany oraz zwalidowany wykorzystując w tym celu dane rzeczywistej rakiety RPT ORION, opracowanej na PW, oraz wyników jej testów lotnych. Model ten następnie uproszczono, starając się zachować dokładność, w celu wykonania obliczeń optymalizacyjnych. Założeniem pracy było wykorzystanie standardowych metod optymalizacji, w szczególności sterowania optymalnego, operujących na sygnałach ciągłych. Ze względu na to, że gazodynamiczny moduł sterujący charakteryzuje się impulsowym działaniem należało wpierw dokonać transformacji układu sterowania z ciągłego na dyskretny, a następnie po wykonaniu obliczeń optymalizacyjnych z powrotem wrócić do dyskretnych impulsów sterujących. W pierwszym kroku więc znacząco uproszczono system sterowania, zastępując równo rozłożone na obwodzie rakiety silniki korekcyjne przez dwa parametry sterujące, które mogą wytwarzać ciąg w dwóch prostopadłych płaszczyznach rakiety w obu kierunkach w sposób ciągły. Taki sygnał następnie podlegał optymalizacji. Wykorzystano w tym celu dwa podejścia nazwane metodami bezpośrednią i pośrednią. W metodzie bezpośredniej potraktowano problem jako standardowy problem optymalizacji nieliniowej,

gdzie przygotowaną funkcję kosztu bezpośrednio przekazano do narzędzia optymalizacyjnego (ang. solver), dedykowanego do rozwiązywania nieliniowych problemów optymalizacyjnych NLP (ang. Non-Linear Programming), w celu otrzymania wyników. W podejściu pośrednim, problem optymalizacji zamieniono najpierw na problem sterowania optymalnego wykorzystując w tym celu Zasadę Maksimum Pontryagina (ZMP). Ułożono równania wykorzystujące zmienne sprzężone otrzymując w ten sposób wzór na gradient funkcji celu. Dodatkowo przy liczeniu gradientów i jakobianów wykorzystano automatyczne różniczkowanie, które znacząco zwiększa dokładność wyników. Mając wynik optymalizacji w postaci przebiegów czasowych parametrów sterujących opracowano metodę jego konwersji na czasy i kierunki odpaleń silników korekcyjnych, dostając od razu ich niezbędną ilość. Ze względu, między innymi, na bardzo dynamiczną zmianę parametrów rakiety w trakcie lotu, konwersja ta wprowadzała pewien błąd trafienia względem wyników optymalizacji. Mając jednak ustaloną liczbę silników korekcyjnych oraz dobre oszacowanie początkowe czasu i kierunku odpalenia każdego z nich, opracowano na tej podstawie dodatkowy problem optymalizacyjny, gdzie minimalizacji podlegał już tylko błąd trafienia, a czasy i kierunki odpaleń pełniły rolę zmiennych optymalizacyjnych. Całą te procedurą powtarzano wielokrotnie, dla różnych warunków początkowych modelu rakiety, otrzymując w ten sposób bazę danych, którą wykorzystywać będzie docelowy algorytm sterujący. Aby sprawdzić jego działanie, wyznaczona zostanie dyspersja miejsc upadku da przypadków niesterowanego, sterowanego z użyciem klasycznej metody oraz sterowanego z użyciem opracowanego algorytmu sterowania. Efektem powinien być zmniejszony rozrzut trafień przy sterowaniu z użyciem opracowanej metody. Mając opracowany problem badawczy oraz sprecyzowany cel badań, teza pracy została sformułowana w następujący sposób:

Teza: **Wykorzystanie metod optymalizacji do wyznaczania parametrów sterowania rakietą oraz sterowanie z wykorzystaniem bazy danych pozwoli skuteczniej i efektywniej naprowadzać rakietę na cel stacjonarny.**

4 Model symulacyjny rakiety

Model symulacyjny rakiety jest niezbędnym elementem procesów projektowania i testowania algorytmów sterowania rakiet. Modele takie pozwalają na uzyskanie wstępnych informacji o charakterze lotu oraz osiągach rakiety przed jej testami poligonowymi [99, 100, 101, 102]. Standardowe podejście do opracowywania nowych konstrukcji składa się z kilku faz projektowania i testowania. Pierwszym etapem jest opracowanie modelu fizycznego obiektu, który powinien dość dobrze odzwierciedlać jego rzeczywiste zachowanie w środowisku. Model ten wraz z odpowiednimi warunkami upraszczającymi służy opracowaniu modelu matematycznego, który składa się z zestawu równań i zależności opisujących dany obiekt. Kolejnym krokiem jest wybór odpowiedniego oprogramowania pozwalającego na opracowanie modelu numerycznego, czy symulacyjnego. Taki model stanowi później główne narzędzie pozwalające na testowanie opracowanych algorytmów sterujących [103, 104, 105].

W fazie testów pierwszym krokiem jest przeprowadzenie symulacji typu Model-In-the-Loop (MIL), które są po prostu wykonaniem kodu symulacyjnego [106, 107]. Jeśli wyniki są satysfakcjonujące kolejnym krokiem jest transkrypcja algorytmu sterującego na docelową formę oraz język programowania, który będzie użyty na komputerze pokładowym rakiety. Mając algorytm w takiej formie przeprowadza się testy Software-In-the-Loop (SIL), które polegają na użyciu algorytmu w docelowym formacie w pętli z modelem symulacyjnym rakiety [108, 109, 110, 111, 112]. Można także wyróżnić fazę testów zwaną Processor-In-the-Loop (PIL), gdzie docelowy algorytm sterowania jest wykonywany na docelowym procesorze, gdzie porównuje się, czy mając dane wejścia uzyskane z symulacji, wyjścia algorytmu odpowiadają tym uzyskanym w fazach poprzednich [113, 114, 115, 116]. Ostatnim krokiem przed testami lotnymi są symulacje typu Hardware-In-the-Loop (HIL), gdzie przy wykorzystaniu symulacji czasu rzeczywistego oraz części rzeczywistych modułów obiektu testowanego, takich jak układ sterowania czy komputer pokładowy, można sprawdzić działanie algorytmów sterujących w warunkach najbardziej zbliżonych do rzeczywistych [117, 118, 119, 120].

Jak wynika z powyższych rozważań model symulacyjny jest bardzo istotną częścią procesu projektowania algorytmów sterujących, więc należy przyłożyć szczególną uwagę procesowi jego opracowania, wraz agregacją dokładnych danych opisujących symulowany obiekt oraz procesom jego weryfikacji i walidacji.

4.1 Opis modelu matematycznego

Proces tworzenia modelu matematycznego należy zacząć od sformułowania założeń dotyczących stopnia dokładności odwzorowania zjawisk rzeczywistych występujących w trakcie lotu rakiety. W niniejszej pracy przyjęto następujące założenia:

1. Rakieta jest bryłą sztywną o sześciu stopniach swobody.

2. Masa, położenie środka ciężkości oraz macierz momentów bezwładności zmieniają się w funkcji działania silnika głównego.

3. Model środowiska zgodny jest z modelem atmosfery standardowej z uwzględnieniem wpływu lokalnych warunków panujących w miejscu startu takich jak temperatura, ciśnienie oraz wilgotność powietrza.

4. Uwzględnia się pionowy profil wiatru, jego prędkość i kierunek w funkcji wysokości.

5. Odpalenia kolejnych silników korekcyjnych mają wpływ na zmianę charakterystyk bezwładnościowych rakiety.

6. Przyspieszenie grawitacyjne jest stałe w czasie lotu, równe wartości z modelu WGS-84 dla danych współrzędnych miejsca startu [121].

7. Przyjęto, że Ziemia jest płaska oraz zaniedbano jej ruch obrotowy.

8. Start rakiety odbywa się z wyrzutni prowadnicowej, do której rakieta jest przymocowana za pośrednictwem dwóch ślizgaczy.

9. W czasie ruchu po wyrzutni nie występuje tarcie ślizgaczy o szynę.

4.1.1 Wykorzystywane układy współrzędnych

Do opisu ruchu pocisku rakietowego wykorzystano następujące układy współrzędnych (Rysunek 4.1):

1. Nawigacyjny układ współrzędnych (N) $O_N x_N y_N z_N$ – prawoskrętny, kartezjański, nieruchomy układ współrzędnych związany z Ziemią. Początek układu O_N znajduje się w miejscu startu rakiety. Płaszczyzna $O_N x_N y_N$ jest styczna do powierzchni Ziemi. Oś $O_N x_N$ skierowana jest w stronę północy geograficznej, oś $O_N y_N$ skierowana jest w stronę wschodnią, zaś kierunek i zwrot osi $O_N z_N$ są zgodne z wektorem przyspieszenia ziemskiego.

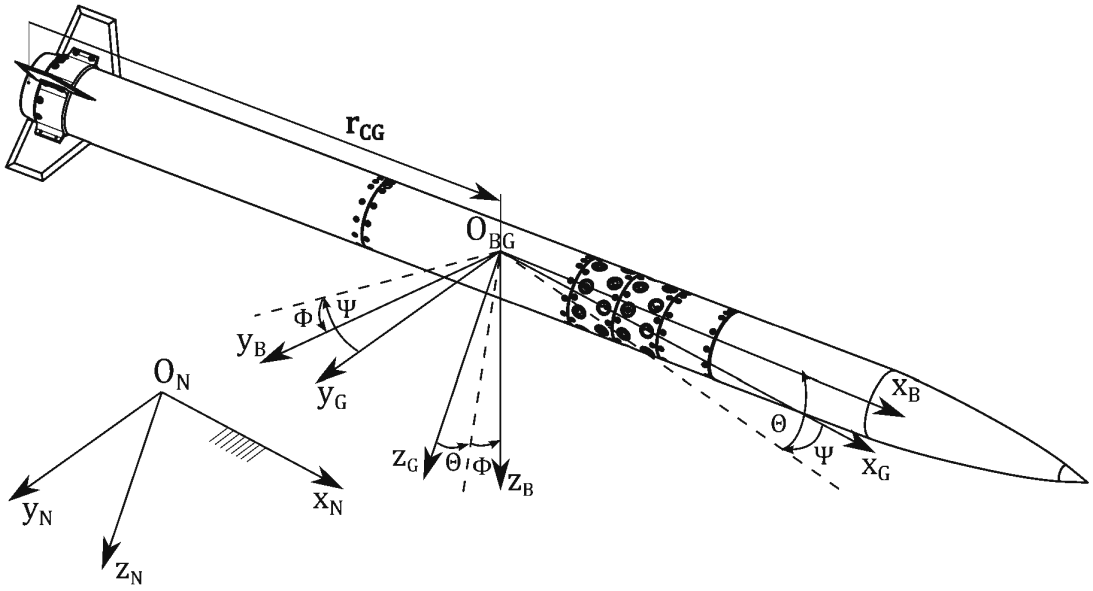
2. Grawitacyjny układ współrzędnych $O_G x_G y_G z_G$ – prawoskrętny, kartezjański, ruchomy układ współrzędnych związany z rakietą. Początek układu O_G znajduje się w środku ciężkości obiektu, a układ w trakcie trwania ruchu pocisku rakietowego pozostaje równoległy do nawigacyjnego układu współrzędnych $O_N x_N y_N z_N$.

3. Układ współrzędnych związany z rakietą wirujący $O_B x_B y_B z_B$ – prawoskrętny, kartezjański układ współrzędnych. Początek układu O_B znajduje się w środku ciężkości obiektu. Oś $O_B x_B$ pokrywa się z osią podłużną i jest skierowana do przodu kadłuba rakiety, oś $O_B y_B$ skierowana jest w prawą stronę patrząc wzdłuż osi $O_B x_B$, zaś oś

$O_B z_B$ stanowi dopełnienie prawoskrętnego układu współrzędnych. Orientację przestrzenną układu (B) względem grawitacyjnego układu współrzędnych (G) opisują lotnicze kąty Eulera: odchylenia Ψ, pochylenia Θ i przechylenia Φ.

4. Układ współrzędnych związany z rakietą niewirujący $O_{NB}x_{NB}y_{NB}z_{NB}$ – prawoskrętny, kartezjański układ współrzędnych. Początek układu O_{NB} znajduje się w środku ciężkości obiektu. Oś $O_{NB}x_{NB}$ pokrywa się z osią podłużną i jest skierowana do przodu kadłuba rakiety, oś $O_{NB}y_{NB}$ skierowana jest w prawą stronę patrząc wzdłuż osi $O_B x_B$ w momencie startu rakiety i nie obraca się razem z rakietą, zaś oś $O_{NB}z_{NB}$ stanowi dopełnienie prawoskrętnego układu współrzędnych. Układ (NB) pokrywa się z układem (B) w momencie startu.

Wszystkie zmienne wykorzystywane w modelu symulacyjnym są zapisywane w układzie rakiety (B) chyba, że zaznaczono inaczej. Punktem referencyjnym dla wektorów położenia, na przykład środka ciężkości, punktu referencyjnego momentów aerodynamicznych, czy położenia płaszczyzn silników korekcyjnych jest koniec dyszy rakiety.



Rysunek 4.1: Układy współrzędnych

4.1.2 Dynamiczne równania ruchu

Dynamiczne równania ruchu rakiety opisują ruch dowolny bryły sztywnej o sześciu stopniach swobody i zmiennych charakterystykach bezwładnościowych względem nieruchomego układu nawigacyjnego. Podstawowe zależności dynamiczne i kinematyczne można znaleźć w pozycjach [122, 123, 124, 125]. Ze względu na spalanie materiału pędnego w czasie lotu zmianie ulegają charakterystyki bezwładnościowe rakiety. Różne podejścia

do uwzględnienia tego efektu w równaniach ruchu można znaleźć w [126, 127, 128]. W poniższej pracy zdecydowano się zastosować podejście przedstawione w [129] uwzględniając człon $\dot{\mathbf{I}}\boldsymbol{\omega}$ ze względu na dużą prędkość kątową rakiety. Wektor stanu rakiety składa się z 13 zmiennych: trzech składowych prędkości linowej, trzech składowych prędkości kątowej, trzech składowych położenia w układzie nawigacyjnym oraz czterech składowych kwaternionu opisującego orientację przestrzenną układu (B) względem układu (N). Równania ruchu zapisywane są i rozwiązywane w układzie (B). Dynamiczne równanie zmiany pędu w czasie ma postać:

$$m\dot{\mathbf{v}} + m(\boldsymbol{\omega} \times \mathbf{v}) = \mathbf{F}_B \tag{4.1}$$

gdzie: m - masa rakiety, $\mathbf{v} = \begin{bmatrix} U & V & W \end{bmatrix}^T$ - wektor prędkości liniowej rakiety, $\boldsymbol{\omega} = \begin{bmatrix} P & Q & R \end{bmatrix}^T$ - wektor prędkości kątowej rakiety, $\mathbf{F}_B = \begin{bmatrix} X_B & Y_B & Z_B \end{bmatrix}^T$ - wektor sił działających na rakietę, stanowiących sumę sił of grawitacji, aerodynamiki, ciągu silnika głównego oraz układu sterowania zgodnie ze wzorem:

$$\mathbf{F}_B = \mathbf{F}_G + \mathbf{F}_A + \mathbf{F}_T + \mathbf{F}_{SK} \tag{4.2}$$

Dynamiczne równanie zmiany krętu w czasie ma postać:

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \dot{\mathbf{I}}\boldsymbol{\omega} = \mathbf{M_B} \tag{4.3}$$

gdzie: $\mathbf{I}$ - macierz momentów bezładności, którą można przedstawić zgodnie z zależnością:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{4.4}$$

a $\mathbf{M_B} = \begin{bmatrix} L_B & M_B & N_B \end{bmatrix}^T$ - wektor momentów działających na rakietę, będący sumą momentów od aerodynamiki, ciągu silnika głównego oraz układu sterowania, zgodnie ze wzorem:

$$\mathbf{M}_B = \mathbf{M}_A + \mathbf{M}_T + \mathbf{M}_{SK} \tag{4.5}$$

Równania dynamiczne uzupełniono zależnościami kinematycznymi pozwalającymi na wyznaczenia trajektorii i orientacji przestrzennej rakiety. Równanie trajektorii ma postać:

$$\dot{\mathbf{R}} = (\mathbf{C}_N^B)^T \mathbf{v} \tag{4.6}$$

gdzie: $\mathbf{R} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ - wektor położenia rakiety w układzie nawigacyjnym, a $\mathbf{C}_N^B$ jest macierzą transformacji z układu (N) do układu (B). Orientację przestrzenną rakiety wyznacza się za pomocą kwaternionów zgodnie z równaniem zawierającym człon norma-

lizujący [130]:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{\Omega}\mathbf{q} + (1 - \mathbf{q}\cdot\mathbf{q})\mathbf{q} \tag{4.7}$$

gdzie: $\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$ - kwaternion, a macierz $\mathbf{\Omega}$ dana jest zależnością:

$$\mathbf{\Omega} = \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} \tag{4.8}$$

Człon normalizujący ma za zadanie zachować normę kwaternionu wynoszącą 1, a która w wyniku błędów zaokrągleń podczas całkowania numerycznego może ulec zmianie. Dodatkowo można wykorzystać dodatkowe zależności pozwalające wyznaczyć kwaternion za pomocą kątów Eulera:

$$q_0 = \cos\left(\frac{1}{2}\Phi\right)\cos\left(\frac{1}{2}\Theta\right)\cos\left(\frac{1}{2}\Psi\right) + \sin\left(\frac{1}{2}\Phi\right)\sin\left(\frac{1}{2}\Theta\right)\sin\left(\frac{1}{2}\Psi\right) \tag{4.9}$$

$$q_1 = \sin\left(\frac{1}{2}\Phi\right)\cos\left(\frac{1}{2}\Theta\right)\cos\left(\frac{1}{2}\Psi\right) - \cos\left(\frac{1}{2}\Phi\right)\sin\left(\frac{1}{2}\Theta\right)\sin\left(\frac{1}{2}\Psi\right) \tag{4.10}$$

$$q_2 = \cos\left(\frac{1}{2}\Phi\right)\sin\left(\frac{1}{2}\Theta\right)\cos\left(\frac{1}{2}\Psi\right) + \sin\left(\frac{1}{2}\Phi\right)\cos\left(\frac{1}{2}\Theta\right)\sin\left(\frac{1}{2}\Psi\right) \tag{4.11}$$

$$q_3 = \cos\left(\frac{1}{2}\Phi\right)\cos\left(\frac{1}{2}\Theta\right)\sin\left(\frac{1}{2}\Psi\right) - \sin\left(\frac{1}{2}\Phi\right)\sin\left(\frac{1}{2}\Theta\right)\cos\left(\frac{1}{2}\Psi\right) \tag{4.12}$$

kąty Eulera za pomocą kwaternionu:

$$\Phi = \operatorname{atan2}\left(2\left(q_0q_1 + q_2q_3\right), 1 - 2\left(q_1^2 + q_2^2\right)\right) \tag{4.13}$$

$$\Theta = \operatorname{asin}\left(2\left(q_0q_2 - q_1q_3\right)\right) \tag{4.14}$$

$$\Psi = \operatorname{atan2}\left(2\left(q_0q_3 + q_1q_2\right), q_0^2 + q_1^2 - q_2^2 - q_3^2\right) \tag{4.15}$$

oraz macierz transformacji $\mathbf{C}_N^B$ za pomocą kwaternionu:

$$\mathbf{C}_N^B = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_3q_0 + q_1q_2\right) & 2\left(q_1q_3 - q_0q_2\right) \\ 2\left(q_1q_2 - q_3q_0\right) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\left(q_0q_1 + q_2q_3\right) \\ 2\left(q_0q_2 + q_1q_3\right) & 2\left(q_2q_3 - q_0q_1\right) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{4.16}$$

4.1.3 Model wyrzutni prowadnicowej

W pierwszej fazie lotu rakieta porusza się po wyrzutni prowadnicowej za pomocą dwóch ślizgaczy. Model tego ruchu został opracowany posługując się równaniami Lagrange'a I rodzaju, przy założeniu że więzy są doskonałe i nieholonomiczne [131]. Rów-

nania (4.1) oraz (4.3) można przekształcić do postaci:

$$\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \mathbf{M}^{-1} \left(\mathbf{f}_0 + \mathbf{f}_\mathbf{C} \right) \tag{4.17}$$

gdzie jako $\mathbf{f}_0$ oznaczono obciążenia czynne, a jako $\mathbf{f}_C$ dodatkowe obciążenia reakcji od wyrzutni. Macierz masowa $\mathbf{M}$ dana jest zależnością (oznaczając jako $\mathbf{1}$ macierz jednostkową):

$$\mathbf{M} = \begin{bmatrix} m\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{4.18}$$

Obciążenia czynne $\mathbf{f}_0$ dane są jako:

$$\mathbf{f}_0 = \begin{bmatrix} \mathbf{F}_B - m\boldsymbol{\omega} \times \mathbf{v} \\ \mathbf{M}_B - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} - \dot{\mathbf{I}}\boldsymbol{\omega} \end{bmatrix} \tag{4.19}$$

Zgodnie z zasadą d'Alemberta obciążenia od więzów można zapisać jako:

$$\mathbf{f}_C = (\nabla \mathcal{R})^T \boldsymbol{\lambda} \tag{4.20}$$

gdzie $\mathcal{R} = \mathcal{R}(\mathbf{v}, \boldsymbol{\omega}) = \mathbf{0}$ jest równaniem więzów nieholonomicznych, a $\boldsymbol{\lambda}$ jest wektorem mnożników Lagrange'a. Zakładając, że równania więzów są liniowe względem wektora stanu można zastosować zapis:

$$\mathcal{R} = \nabla \mathcal{R} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{0} \tag{4.21}$$

Zmiennymi do wyznaczenia w równaniu (4.20) są mnożniki Lagrange'a. Wyznaczyć je można różniczkując po czasie równanie (4.21) otrzymując zależność:

$$\dot{\mathbf{J}} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} + \mathbf{J} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \mathbf{0} \tag{4.22}$$

Podstawiając do równania (4.22) zależność (4.17) oraz (4.20), a następnie przekształcając w celu obliczenia mnożników Lagrange'a otrzymano wzór:

$$\boldsymbol{\lambda} = - \left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \right)^{-1} \left(\dot{\mathbf{J}} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} + \mathbf{J}\mathbf{M}^{-1}\mathbf{f}_0 \right) \tag{4.23}$$

Następnie posługując się zależnością (4.20) można wyznaczyć siły i momenty reakcji w czasie ruchu po wyrzutni, a następnie dodać je do pozostałych obciążeń działających na rakietę $\mathbf{F}_B$ oraz $\mathbf{M}_B$, by następnie rozwiązać dynamiczne równania ruchu. Sam ruch po wyrzutni podzielono na dwie części. W pierwszej oba ślizgacze znajdują się na wyrzutni.

Przyjęto, że reakcje od obu ślizgaczy można zastąpić jednym ślizgaczem zastępczym, który odbiera wszystkie obciążenia. W tym przypadku możliwy jest tylko ruch postępowy wzdłuż osi rakiety, prędkości liniowe poprzeczne oraz wszystkie prędkości kątowe są równe zero. Macierz $\mathbf{J}$ występująca w równaniu (4.21) dla tego przypadku ma postać:

$$\mathbf{J} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.24}$$

a jej pochodna jest macierzą zerową o wymiarach 5x6. W drugiej fazie ruchu przedni ślizgacz opuścił szynę, a rakieta może wykonywać dodatkowo ruch zawiasowy na tylnym ślizgaczu. W tym przypadku prędkość kątowa przechylania wynosi zero oraz składowe prędkości punktu kontaktu tylnego ślizgacza z szyną wyrzutni w kierunku prostopadłym do szyny są równe zero. Drugi warunek można zapisać w postaci równania:

$$[\mathbf{v}_R]_R = \mathbf{C}_B^R (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_b) \tag{4.25}$$

gdzie zapis $[\mathbf{v}_R]_R$ oznacza prędkość punktu R, czyli miejsca styku tylnego ślizgacza z szyną, wyrażoną w układzie wyrzutni, który pokrywa się z układem (B) w chwili początkowej, $\mathbf{r}_b$ jest położeniem tylnego ślizgacza względem układu rakiety, a $\mathbf{C}_B^R$ jest macierzą transformacji z układu rakiety do układu wyrzutni. Sprowadzając to równanie wraz w warunkiem zerowania prędkości kątowej przechylania do postaci (4.21) macierz $\mathbf{J}$ można wyrazić jako:

$$\mathbf{J} = \begin{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}_B^R & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}_B^R [\mathbf{r}_b]_x \\ 0 \quad 0 \quad 0 & 1 \quad 0 \quad 0 \end{bmatrix} \tag{4.26}$$

a pochodną macierzy $\mathbf{J}$ jako:

$$\dot{\mathbf{J}} = \begin{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}_B^R [\boldsymbol{\omega}]_x & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}_B^R [\boldsymbol{\omega}]_x [\mathbf{r}_b]_x \\ 0 \quad 0 \quad 0 & 0 \quad 0 \quad 0 \end{bmatrix} \tag{4.27}$$

gdzie oznaczenie $[]_x$ jest macierzą antysymetryczną pozwalającą zamienić mnożenie wektorowe na mnożenie macierzowe. Wektor $\mathbf{A} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$ można przedstawić w tej postaci jako:

$$[\mathbf{A}]_x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{4.28}$$

Po opuszczeniu wyrzutni przez rakietę, czyli zejściu z prowadnicy drugiego ślizgacza, porusza się ona ruchem swobodnym (bez więzów).

4.1.4 Model środowiska

Na model środowiska składa się model atmosfery standardowej uzupełniony o wpływ wilgotności powietrza oraz lokalne warunki, w postaci temperatury i ciśnienia, panujące w miejscu startu, a także pionowy profil wiatru. Dodatkowo wyznaczane są aerodynamiczne kąty napływu pozwalające wyznaczyć orientację rakiety względem napływającego powietrza. Profil wiatru wczytywany jest jako tabela prędkości oraz kierunku meteorologicznego (kierunek skąd wieje) wiatru w funkcji wysokości nad ziemią. Jeśli jako v_w oznaczymy prędkość wiatru, a jako ψ_w kierunek meteorologiczny, wektor prędkości wiatru w układzie nawigacyjnym (N) będzie miał postać:

$$\mathbf{v}_w = \begin{bmatrix} -v_w \cos(\psi_w) & -v_w \sin(\psi_w) & 0 \end{bmatrix} \tag{4.29}$$

Aby uzyskać prędkość rakiety względem powietrza $\mathbf{v}_A = \begin{bmatrix} u_A & v_A & w_A \end{bmatrix}$ należy wykonać operację:

$$\mathbf{v}_A = \mathbf{v} - \mathbf{C}_N^B \mathbf{v}_w \tag{4.30}$$

Mając prędkość względem napływu można wyznaczyć aerodynamiczne kąty napływu [124]. Kąt natarcia α zdefiniowany jest jako:

$$\alpha = \operatorname{atan2}(w_A, u_A) \tag{4.31}$$

Kąt ślizgu β dany jest zależnością:

$$\beta = \operatorname{asin}\left(\frac{v_A}{|\mathbf{v}_A|}\right) \tag{4.32}$$

Dodatkowo kąty napływu można zdefiniować w innej konwencji przydatnej przy opisie ciał osiowosymetrycznych: Całkowity kąt natarcia α_T opisuje wzór:

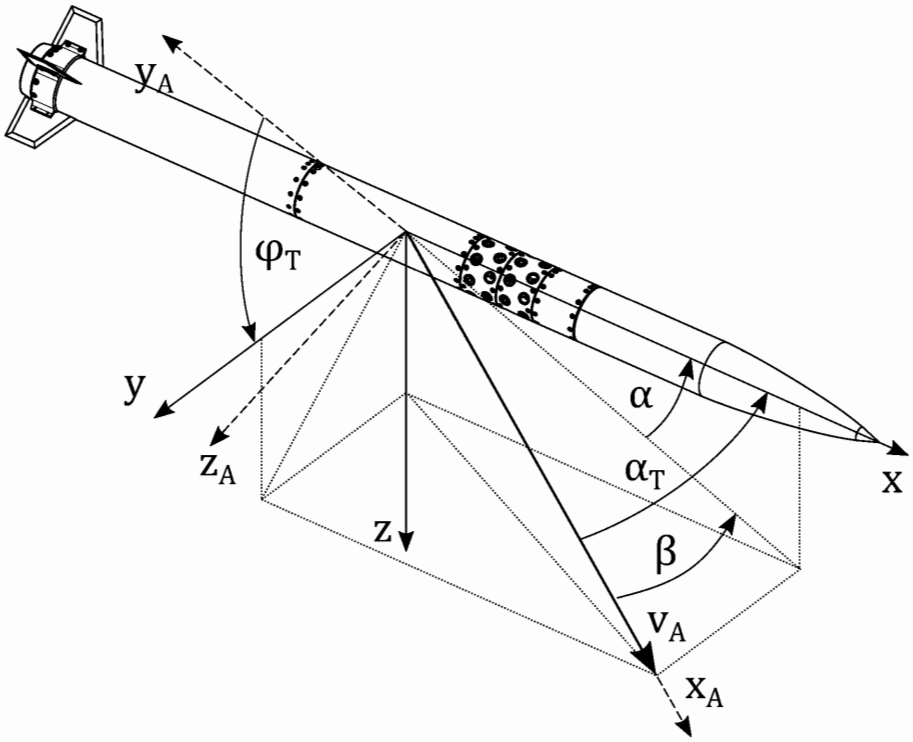
$$\alpha_T = \operatorname{acos}\left(\frac{u_A}{|\mathbf{v}_A|}\right) \tag{4.33}$$

Aerodynamiczny kąt przechylenia ϕ_T dany jest zależnością:

$$\phi_T = \operatorname{atan2}(v_A, w_A) \tag{4.34}$$

Definicję kątów napływu przedstawia także Rysunek 4.2. Osiami x, y, z zaznaczono tam układ rakiety (B), a osiami x_A, y_A, z_A układ aerodynamiczny związany z wektorem prędkości rakiety względem powietrza, powstały poprzez obrócenie układu (B) wykorzystując

kąty α_T, ϕ_T.



Rysunek 4.2: Definicja aerodynamicznych kątów napływu

.

Model atmosfery standardowej (ISA) pozwala na wyznaczenie temperatury, ciśnienia, gęstości oraz prędkości dźwięku w zależności od wysokości. Ze względu na osiągi rozpatrywanej rakiety ograniczono się tylko do zależności występujących dla troposfery, czyli do wysokości 11 km. Dodatkowo uwzględniono warunki panujące w miejscu startu, temperaturę, ciśnienie oraz wilgotność powietrza, bazując na opisie z [132]. Wysokość nad ziemią h można przekształcić do wysokości geopotencjalnej h_{geo} posługując się wzorem:

$$h_{geo} = \frac{r_e(\phi)h}{r_e(\phi) + h}\left(\frac{g_0(\phi)}{g_0}\right) \tag{4.35}$$

gdzie $g_0 = 9,80665\frac{m}{s^2}$ jest przyspieszeniem grawitacyjnym zgodnym z ISA, a $g_0(\phi)$ jest to wartość przyspieszenia Ziemskiego w funkcji szerokości geograficznej miejsca startu ϕ, która można obliczyć posługując się równaniem Lamberta:

$$g_0(\phi) = 9,80616\left(1 - 0,0026373\cos\left(2\phi\right) + 0,0000059\cos^2\left(2\phi\right)\right) \tag{4.36}$$

Dodatkowo $r_e(\phi)$ jest to odległość dowolnego punktu na powierzchni ziemi od środka Ziemi, dana zależnością:

$$r_e(\phi) = \sqrt{\frac{a^4 + c^4\tan^2\phi}{a^2 + c^2\tan^2\phi}} \tag{4.37}$$

gdzie $a = 6378137$ m jest promieniem Ziemi na równiku, a $c = 6356752,3$ m promieniem Ziemi na biegunie. Transformacja wysokości jest niezbędna, aby uniezależnić równania atmosfery od zmian wartości przyspieszenia grawitacyjnego wraz z wysokością. Zmianę temperatury wraz z wysokością opisuje zależność:

$$T = T_0 - 0,0065 h_{geo} \tag{4.38}$$

gdzie T_0 jest temperaturą w miejscu startu. Zmiana ciśnienia dana jest wzorem:

$$p = p_0 \left(\frac{T}{T_0} \right)^{\frac{g0}{0,0065R}} \tag{4.39}$$

gdzie p_0 jest ciśnieniem w miejscu startu, a R jest stałą gazową. Gęstość powietrza można wyznaczyć korzystając ze wzoru:

$$\rho = \frac{p}{RT} \tag{4.40}$$

a prędkość dźwięku opisuje zależność:

$$a = \sqrt{1,4RT} \tag{4.41}$$

Uwzględnienie wilgotności powietrza ma wpływ na stałą gazową, która opisuje mieszaninę powietrza i pary wodnej. Można ją wyznaczyć ze wzoru:

$$R = \frac{R_0 p}{p - 0,377998 p_v} \tag{4.42}$$

gdzie $R_0 = 287,05287 \frac{J}{molK}$ jest stałą gazową dla powietrza według ISA, a p_v jest ciśnieniem cząstkowym pary wodnej danym równaniem:

$$p_v = p_s RH \tag{4.43}$$

gdzie RH jest względną wilgotnością powietrza, a p_s ciśnieniem pary wodnej w stanie saturacji. W zależności od stosunku temperatury powietrza do temperatury krzepnięcia T/T_0 ciśnienie saturacji dane jest wzorem:

$$p_s = \left(\frac{T}{T_0} \right)^{a_w} 10^{b_w \left(\frac{T_0}{T} \right) + c_w} \tag{4.44}$$

ze stałymi $a_w = -4,927432$, $b_w = -10,752935$, $c_w = 11,538976$ jeśli stosunek ten jest większy niż 1 lub wzorem:

$$p_s = \left(\frac{T}{T_0} \right)^{a_i} 10^{b_i \left(\frac{T_0}{T} \right) + c_i} \tag{4.45}$$

ze stałymi $a_i = -0,322862$, $b_i = -9,903888$, $c_i = 10,689717$ w przeciwnym wypadku.

Dodatkowo wyznacza się wartości liczby Macha:

$$Ma = \frac{|\mathbf{v}_A|}{a} \tag{4.46}$$

oraz ciśnienia dynamicznego:

$$q_{dyn} = \frac{1}{2}\,\rho\,|\mathbf{v}_A|^2 \tag{4.47}$$

4.1.5 Charakterystyki bezwładnościowe

Wraz ze spalaniem materiału pędnego silnika głównego oraz uruchamianiem kolejnych silników korekcyjnych zmianie ulegają parametry bezwładnościowe rakiety. Masę rakiety $m(t)$ w chwili czasu t można opisać zależnością:

$$m(t) = m_0 - (m_0 - m_k)\,\frac{\int_0^t F_T(t)dt}{I_c} - \sum_{i=1}^{N_{sk}} m_{sk_i}\frac{\int_{t_{0_i}}^t F_{T_{sk_i}}(t - t_{0_i})dt}{I_{c_{sk_i}}} \tag{4.48}$$

gdzie N_{sk} jest liczbą silników korekcyjnych, m_0 jest masą startową, m_k masą po wypaleniu materiału pędnego silnika głównego, m_{sk_i} masą materiału pędnego i-tego silnika korekcyjnego, I_c oraz $I_{c_{sk_i}}$ impulsem całkowitym odpowiednio silnika głównego oraz i-tego silnika korekcyjnego, $F_T(t)$ siłą ciągu w czasie t silnika głównego, a $F_{T_{sk_i}}(t - t_{0_i})$ siłą ciągu i-tego silnika korekcyjnego od czasu jego uruchomienia t_{0_i}. Ruch środka ciężkości rakiety wynikający ze spalania materiału pędnego silnika głównego można wyznaczyć z zależności:

$$\mathbf{r}_{cg_R}(t) = \mathbf{r}_{cg_0} - (\mathbf{r}_{cg_0} - \mathbf{r}_{cg_k})\,\frac{\int_0^t F_T(t)dt}{I_c} \tag{4.49}$$

gdzie $\mathbf{r}_{cg_0}$ i $\mathbf{r}_{cg_k}$ są odpowiednio położeniem środka ciężkości rakiety w chwili startu i rakiety bez materiału pędnego. Przy założeniu, że geometria ziarna materiału pędnego silnika korekcyjnego ma postać wydrążonego cylindra oraz że spalanie ziarna następuje symetrycznie od wewnętrznej powierzchni cylindra do zewnętrznej, położenie środka ciężkości pojedynczego silnika korekcyjnego $\mathbf{r}_{cg_{sk_i}}$ nie ulega zmianie w czasie pracy silnika. Ruch środka ciężkości całej rakiety można następnie przedstawić zgodnie z równaniem:

$$\mathbf{r}_{cg}(t) = \frac{m(t)\mathbf{r}_{cg_R}(t) - \sum_{i=1}^{N_{sk}} m_{sk_i}\mathbf{r}_{cg_{sk_i}}\frac{\int_{t_{0_i}}^t F_{T_{sk_i}}(t-t_{0_i})dt}{I_{c_{sk_i}}}}{m(t)} \tag{4.50}$$

Zmianę macierzy momentów bezwładności rakiety wynikającą ze spalania materiału pędnego silnika głównego opisuje zależność:

$$\mathbf{I}_R(t) = \mathbf{I}_0 - (\mathbf{I}_0 - \mathbf{I}_k)\,\frac{\int_0^t F_T(t)dt}{I_c} \tag{4.51}$$

gdzie $\mathbf{I}_0$ i $\mathbf{I}_k$ są odpowiednio macierzą momentów bezwładności w chwili startu i rakiety bez materiału pędnego silnika głównego. Macierz momentów bezwładności każdego z silników korekcyjnych, przy poczynionych założeniach geometrycznych, można zapisać w układzie odpowiedniego silnika korekcyjnego jako:

$$[\mathbf{I}_{sk_i}]_{sk} = \begin{bmatrix} I_{xx_{sk}} & 0 & 0 \\ 0 & I_{yy_{sk}} & 0 \\ 0 & 0 & I_{zz_{sk}} \end{bmatrix} \tag{4.52}$$

W układzie silnika oś x pokrywa się z wysokością cylindra. Każdy silnik jest pochylony względem układu rakiety o 90^o, a także znajduję się w konkretnym miejscu na obwodzie rakiety. Z tego względu należy najpierw macierz $\mathbf{I}_{sk_i}$ przetransformować do układu rakiety, a następnie korzystając z twierdzenia Steinera przenieść ją do środka ciężkości rakiety. Zapisując jako $\mathbf{r}_{cg_{w_i}} = \mathbf{r}_{cg_{sk_i}} - \mathbf{r}_{cg_R}$ położenie środka ciężkości i-tego silnika korekcyjnego względem środka ciężkości rakiety, można to zrobić posługując się zależnością:

$$\mathbf{I}_{sk_i} = \mathbf{C}^B_{SK_i} [\mathbf{I}_{sk_i}]_{sk} \left(\mathbf{C}^B_{SK_i}\right)^T + m_{sk_i} \left(\mathbf{1}\mathbf{r}^T_{cg_{w_i}}\mathbf{r}_{cg_{w_i}} - \mathbf{r}_{cg_{w_i}}\mathbf{r}^T_{cg_{w_i}}\right) \tag{4.53}$$

Macierz transformacji $\mathbf{C}^B_{SK_i}$ dana jest wzorem:

$$\mathbf{C}^B_{SK_i} = \begin{bmatrix} 0 & 0 & -1 \\ -\sin\psi_{sk_i} & \cos\psi_{sk_i} & 0 \\ \cos\psi_{sk_i} & \sin\psi_{sk_i} & 0 \end{bmatrix} \tag{4.54}$$

gdzie ψ_{sk_i} jest położeniem kątowym i-tego silnika korekcyjnego względem układu (B). Całkowita zmiana momentów bezwładności może być zapisana jako (uwzględniając fakt że zarówno $\mathbf{I}_0$ jak i $\mathbf{I}_k$ zawierają masę silników korekcyjnych):

$$\mathbf{I}(t) = \mathbf{I}_R(t) - \sum_{i=1}^{N} \mathbf{I}_{sk_i} \frac{\int_{t_{0_i}}^{t} F_{T_{sk_i}}(t - t_{0_i})dt}{I_{c_{sk_i}}} \tag{4.55}$$

4.1.6 Obciążenia aerodynamiczne

Charakterystyki aerodynamiczne rakiety determinują w dużym stopniu jej osiągi oraz charakter lotu, wymagane jest więc opracowanie dobrego modelu uwzględniającego możliwie wiele czynników wpływających na lot rakiety. Wykorzystany model aerodynamiczny powstał w oparciu o pół-empiryczny model wykorzystywany w oprogramowaniu PRODAS [133], który uwzględnia efekty związane z wirowaniem rakiety wyposażonej w cztery stateczniki. Model ten specyfikuje współczynniki w funkcji liczby Macha Ma z podziałem na części statyczną, dynamiczną oraz wyższego rzędu. Założono także, że rakieta jest symetryczna w płaszczyznach pochylania i odchylania. Siłę aerodynamiczną można

obliczyć z równania:

$$\mathbf{F}_A = q_{dyn} S_{ref} \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \tag{4.56}$$

gdzie S_{ref} jest powierzchnią odniesienia, tu powierzchnią przekroju poprzecznego rakiety, a C_X, C_Y, C_Z współczynnikami odpowiednio siły osiowej, bocznej i normalnej. Współczynnik siły osiowej dany jest równaniem:

$$C_X = -\left(C_{X_{0_{on}}}(Ma) + \left(C_{X_{0_{off}}}(Ma) - C_{X_{0_{on}}}(Ma) \right) \frac{1 - P_{eng}}{T_{eng} s + 1} + C_{X_{\alpha^2}}(Ma) \sin^2 \alpha_T \right) \tag{4.57}$$

gdzie $C_{X_{0_{on}}}(Ma)$, $C_{X_{0_{off}}}(Ma)$ są współczynnikami oporu dla zerowego kąta natarcia odpowiednio w czasie pracy silnika i po pracy silnika, $C_{X_{\alpha^2}}(Ma)$ jest pochodną współczynnika siły osiowej po kwadracie całkowitego kąta natarcia, P_{eng} ma wartość 1 gdy silnik pracuje i 0 w przeciwnym wypadku, a T_{eng} jest stałą czasową w modelu pierwszego rzędu, która ma za zadanie wygładzić współczynnik oporu w momencie zakończenia pracy silnika. Współczynnik siły bocznej można wyznaczyć z zależności:

$$C_Y = -C_{N_\alpha}(Ma) \frac{v_A}{|\mathbf{v}_A|} + C_{N_Q}(Ma) \frac{R d_{ref}}{2 |\mathbf{v}_A|} + C_{Y_{\gamma\alpha}}(Ma) \sin^2 \alpha_T \sin 4\phi_T \frac{w_A}{|\mathbf{v}_A|} \tag{4.58}$$

gdzie $C_{N_\alpha}(Ma)$ jest pochodną siły normalnej po kącie natarcia, $C_{N_Q}(Ma)$ jest w tym przypadku pochodną siły bocznej po prędkości kątowej odchylania, a $C_{Y_{\gamma\alpha}}(Ma)$ jest pochodną wyższego rzędu uwzględniającą wpływ wirów schodzących z noska rakiety i uderzających w stateczniki podczas lotu z dużą prędkością wirowania. Współczynnik siły normalnej można wyznaczyć z zależności:

$$C_Z = -C_{N_\alpha}(Ma) \frac{w_A}{|\mathbf{v}_A|} + C_{N_Q}(Ma) \frac{Q d_{ref}}{2 |\mathbf{v}_A|} - C_{Y_{\gamma\alpha}}(Ma) \sin^2 \alpha_T \sin 4\phi_T \frac{v_A}{|\mathbf{v}_A|} \tag{4.59}$$

gdzie $C_{N_Q}(Ma)$ jest w tym przypadku pochodną siły normalnej po prędkości kątowej pochylania. Ze względu na symetrię rakiety pochodne aerodynamiczne we wzorach (4.58) i (4.59) mają takie same wartości. Moment aerodynamiczny dany jest zależnością:

$$\mathbf{M}_A = q_{dyn} S_{ref} d_{ref} \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} + (\mathbf{r}_{aero} - \mathbf{r}_{cg}) \times \mathbf{F}_A \tag{4.60}$$

gdzie d_{ref} jest długością odniesienia, tu średnicą rakiety, a C_l, C_m, C_n współczynnikami odpowiednio momentu przechylającego, pochylającego i odchylającego. Współczynniki momentów aerodynamicznych podawane są względem punktu referencyjnego, którego położenie względem początku układu (B) opisuje wektor $(\mathbf{r}_{aero} - \mathbf{r}_{cg})$, stąd niezbędna jest

poprawka przenosząca momenty do środka ciężkości rakiety. Współczynnik momentu przechylającego dany jest zależnością:

$$C_l = C_{l_0}(Ma) + C_{l_P}(Ma)\frac{Pd_{ref}}{2\left|\mathbf{v}_A\right|} + C_{l_{\gamma\alpha}}(Ma)\sin^2\alpha_T \sin 4\phi_T \tag{4.61}$$

gdzie $C_{l_0}(Ma)$ jest współczynnikiem wymuszającym wirowanie rakiety w osi podłużnej, $C_{l_P}(Ma)$ jest pochodną tłumiącą momentu przechylającego, a $C_{l_{\gamma\alpha}}(Ma)$ jest pochodną wyższego rzędu uwzględniającą wpływ wirów schodzących z noska rakiety i uderzających w stateczniki podczas lotu z dużą prędkością wirowania. Współczynnik momentu pochylającego dany jest zależnością:

$$C_m = C_{m_\alpha}(Ma)\frac{w_A}{\left|\mathbf{v}_A\right|} + C_{m_Q}(Ma)\frac{Qd_{ref}}{2\left|\mathbf{v}_A\right|} + C_{n_{\gamma\alpha}}(Ma)\sin^2\alpha_T \sin 4\phi_T \frac{v_A}{\left|\mathbf{v}_A\right|} \tag{4.62}$$

gdzie $C_{m_\alpha}(Ma)$ jest pochodną współczynnika momentu pochylającego po kącie natarcia, $C_{m_Q}(Ma)$ jest pochodną tłumiącą momentu pochylającego, a $C_{n_{\gamma\alpha}}(Ma)$ jest pochodną wyższego rzędu uwzględniającą wpływ wirów schodzących z noska rakiety i uderzających w stateczniki podczas lotu z dużą prędkością wirowania. Współczynnik momentu odchylającego dany jest zależnością:

$$C_n = -C_{m_\alpha}(Ma)\frac{v_A}{\left|\mathbf{v}_A\right|} + C_{m_Q}(Ma)\frac{Rd_{ref}}{2\left|\mathbf{v}_A\right|} + C_{n_{\gamma\alpha}}(Ma)\sin^2\alpha_T \sin 4\phi_T \frac{w_A}{\left|\mathbf{v}_A\right|} \tag{4.63}$$

Pochodne aerodynamiczne w równaniu (4.63) co do wartości są takie same jak w równaniu (4.62).

4.1.7 Obciążenia od zespołu napędowego

Siła ciągu silnika głównego dana jest wzorem:

$$\mathbf{F}_T = \mathbf{C}_T^B \begin{bmatrix} F_T(t) + A_e\left(p_e - p\right) P_{eng} \\ 0 \\ 0 \end{bmatrix} \tag{4.64}$$

gdzie $F_T(t)$ jest ciągiem silnika w funkcji czasu, A_e powierzchnią wylotową dyszy, a p_e ciśnieniem odniesienia dla ztabelaryzowanej wartości ciągu. Model pozwala także uwzględnić nieosiowe działanie ciągu poprzez kąty odchylenia osi ciągu od osi rakiety Θ_T oraz Φ_T. Macierz transformacji z układu siły ciągu do układu (B) dana jest wzorem:

$$\mathbf{C}_T^B = \begin{bmatrix} \cos\Theta_T & 0 & -\sin\Theta_T \\ \sin\Theta_T \sin\Phi_T & \cos\Phi_T & \cos\Theta_T \sin\Phi_T \\ \sin\Theta_T \cos\Phi_T & -\sin\Phi_T & \cos\Theta_T \cos\Phi_T \end{bmatrix} \tag{4.65}$$

Moment od siły ciągu można otrzymać z zależności:

$$\mathbf{M}_T = -\mathbf{r}_{cg} \times \mathbf{F}_T \tag{4.66}$$

4.1.8 Obciążenia od grawitacji

Równania ruchu rozwiązywane są w układzie związanym ze środkiem masy obiektu. Z tego względu grawitacja nie powoduje momentu obracającego rakietę. Siłę od grawitacji można opisać zależnością:

$$\mathbf{F}_g = m\mathbf{C}_N^B \begin{bmatrix} 0 \\ 0 \\ g_{acc} \end{bmatrix} \tag{4.67}$$

gdzie g_{acc} jest przyspieszeniem grawitacyjnym zależnym od współrzędnych geograficznych miejsca startu zgodnie z modelem WGS-84 oraz stałym przez cały okres lotu.

4.1.9 Obciążenia od układu sterowania gazodynamicznego

Uruchomienie kolejnych silników korekcyjnych następuje po wysłaniu sygnału sterującego z układu sterowania. Silniczki umieszczone są równomiernie na obwodzie rakiety, w kilku płaszczyznach prostopadłych do osi podłużnej rakiety. Siła generowana przez układ sterowania może być wyrażona wzorem:

$$\mathbf{F}_{SK} = \begin{bmatrix} 0 \\ \sum_{i=1}^{N} F_{T_{sk_i}}(t - t_{0_i}) \sin \psi_{sk_i} \\ -\sum_{i=1}^{N} F_{T_{sk_i}}(t - t_{0_i}) \cos \psi_{sk_i} \end{bmatrix} \tag{4.68}$$

Moment generowany przez układ sterowania dany jest równaniem:

$$\mathbf{M}_{SK} = \sum_{i=1}^{N} (\mathbf{r}_{sk_i} - \mathbf{r}_{cg}) \times \mathbf{F}_{sk_i} \tag{4.69}$$

gdzie $\mathbf{r}_{sk_i}$ jest położeniem płaszczyzny działania i-tego silnika korekcyjnego względem punktu referencyjnego (końca dyszy rakiety).

4.1.10 Model jednostki nawigacji inercjalnej

Symulacja zawiera także model czujników w postaci giroskopów, akcelerometrów oraz czujnika ciśnienia, a także uproszczony model systemu nawigacji inercjalnej INS. Dokładny opis modelu można znaleźć w [134]. Każdy z czujników został obarczony modelem saturacji, kwantyzacji oraz czasu próbkowania. Dla czujnika ciśnienia są to jedyne

uwzględniane błędy. Przyspieszenie środka masy rakiety można wyznaczyć jako:

$$\mathbf{a}_{cg} = \frac{\mathbf{F}_B}{m} \tag{4.70}$$

Ze względu na to, że akcelerometry zazwyczaj nie są umieszczone w środku masy rakiety, należy przyspieszenie z równania (4.70) przeliczyć do położenia czujnika zgodnie ze wzorem:

$$\mathbf{a}_{IMU} = \mathbf{a}_{cg} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\mathbf{r}_{IMU} - \mathbf{r}_{cg})) + \dot{\boldsymbol{\omega}} \times (\mathbf{r}_{IMU} - \mathbf{r}_{cg}) - \mathbf{C}_N^B \begin{bmatrix} 0 \\ 0 \\ g_{acc} \end{bmatrix} \tag{4.71}$$

gdzie $\mathbf{r}_{IMU}$ jest położeniem czujnika względem punktu referencyjnego. Dodatkowo uwzględniono błędy sprzężeń skrośnych między osiami c_{ij}, przesunięcia zera b_i, skali s_i, szum σ_a oraz przyjęto, że czujnik można zamodelować jako układ drugiego rzędu ze współczynnikiem tłumienia ξ_a oraz częstością drgań własnych ω_{n_a}. Wynikowe przyspieszenie wskazywane przez czujnik daje wzór:

$$\mathbf{a}_{meas} = \frac{\omega_{n_a}^2}{s^2 + 2\xi_a\omega_{n_a}s + \omega_{n_a}^2} \left(\begin{bmatrix} s_x & -c_{xy} & c_{xz} \\ c_{xy} & s_y & -c_{yz} \\ -c_{xz} & c_{yz} & s_z \end{bmatrix} \mathbf{a}_{IMU} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \right) + \sigma_a \tag{4.72}$$

Przyjmując analogiczne założenia, prędkości kątowe uzyskane z modelu giroskopów przedstawia równanie:

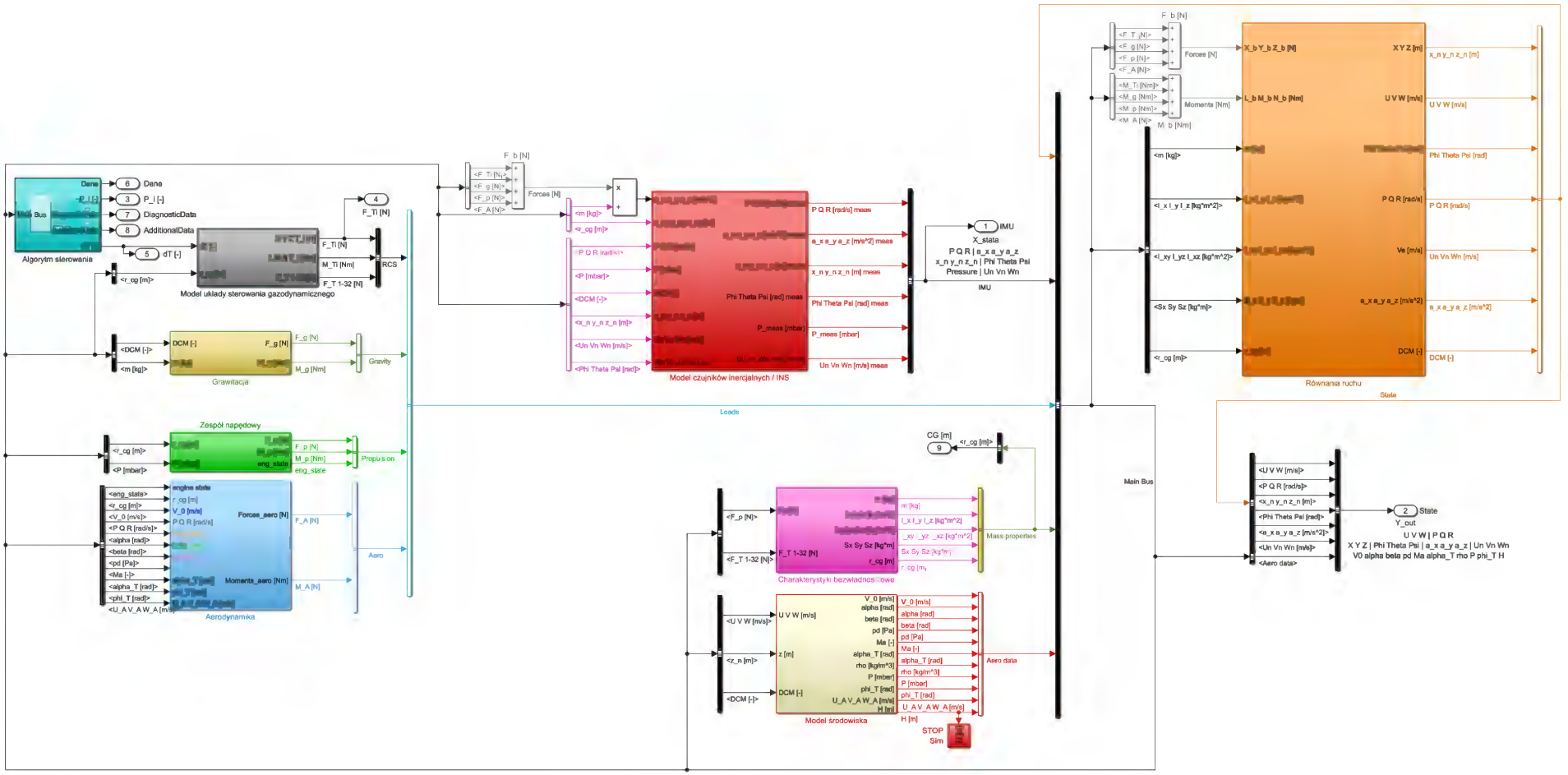
$$\boldsymbol{\omega}_{meas} = \frac{\omega_{n_g}^2}{s^2 + 2\xi_g\omega_{n_g}s + \omega_{n_g}^2} \left(\begin{bmatrix} s_x & -c_{xy} & c_{xz} \\ c_{xy} & s_y & -c_{yz} \\ -c_{xz} & c_{yz} & s_z \end{bmatrix} \boldsymbol{\omega} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \mathbf{G}\mathbf{a}_{cg} \right) + \sigma_g \tag{4.73}$$

gdzie $\mathbf{G}$ jest macierzą uwzględniającą wpływ przyspieszeń na wskazania giroskopów. Aby uniknąć potrzeby projektowania skomplikowanych algorytmów filtracji w celu wyznaczania orientacji przestrzennej, prędkości i pozycji na podstawie danych z czujników na potrzeby układu INS, zdecydowano się na zaburzanie rezultatów symulacji błędami szumu oraz próbkowania. Takie podejście pozwala w dość dobry sposób dostroić wyniki modelu INS do rzeczywistych danych uzyskanych z systemu nawigacyjnego bez potrzeby komplikacji modelu matematycznego.

4.2 Opis modelu symulacyjnego

Model symulacyjny został zaprojektowany w środowisku MATLAB/Simulink w wersji R2023a. Modele opisane w poprzednich rozdziałach zaprogramowano w Simulinku zgodnie z Rysunkiem 4.3. Dane wejściowe są wczytywane skryptem napisanym w MATLA-

Bie, w którym to również napisano skrypty do analizy wyników. Model układu sterowania, a także weryfikacja oraz walidacja modelu zostanie opisana w kolejnym rozdziale. Przedstawiony model symulacyjny składa się z bloków wyznaczających obciążenia od układu sterowania, grawitacji, aerodynamiki oraz zespołu napędowego, bloków wyznaczających parametry środowiska oraz charakterystyki bezwładnościowe, a także bloków odpowiadających za rozwiązywanie dynamicznych równań ruchu oraz opisujących model czujników bezwładnościowych. Wejściem do modelu układu sterowania jest macierz sygnałów odpaleń poszczególnych silników korekcyjnych oraz aktualne położenie środka ciężkości, a wyjściem siły i momenty generowane przez układ sterowania. Opis modelu znajduje się w rozdziale 4.1.9. Model obciążeń od grawitacji opisano w rozdziale 4.1.8. Model wymaga podania aktualnej masy rakiety oraz macierzy transformacji z układu nawigacyjnego do układu rakiety, a zwraca siły i momenty wynikające z działania grawitacji. Zespół napędowy wymaga podania aktualnego położenia środka ciężkości oraz lokalnego ciśnienia atmosferycznego, a zwraca obciążenia wynikające z działania silnika głównego oraz flagę informującą o działaniu silnika. Model opisany został w rozdziale 4.1.7. Obciążenia aerodynamiczne wyznaczane są zgodnie z zależnościami opisanymi w rozdziale 4.1.6. Jako dane wejściowe model wymaga stanu pracy silnika głównego, aktualnego położenia środka ciężkości, prędkości liniowej rakiety względem powietrza, prędkości kątowych, kątów napływu, ciśnienia dynamicznego oraz aktualnej liczby Macha. Model zwraca siły i momenty aerodynamiczne. Wejściem do modelu czujników bezwładnościowych są aktualne przyspieszenia i prędkości kątowe rakiety, położenie środka ciężkości, położenie i prędkość rakiety w układzie nawigacyjnym, kąty orientacji przestrzennej, macierz transformacji oraz ciśnienie atmosferyczne, a wyjściem te same parametry zaburzone przez model czujników opisany w rozdziale 4.1.10. Zmiany charakterystyk bezwładnościowych wyznaczane są na podstawie aktualnego ciągu silnika głównego oraz ciągu poszczególnych silników korekcyjnych. Model zwraca aktualną masę, położenie środka ciężkości oraz macierz momentów bezwładności wyznaczane zgodnie z zależnościami opisanymi w rozdziale 4.1.5. Wejściem do modelu środowiska opisanego w rozdziale 4.1.4 są prędkości liniowe rakiety, wysokość oraz macierz transformacji. Wyjściem z modelu są kąty napływu, liczba Macha, prędkości linowe względem powietrza, ciśnienie dynamiczne, gęstość oraz ciśnienie otoczenia. Dynamiczne równania ruchu opisane w rozdziałach 4.1.2 oraz 4.1.3 wymagają podania sił i momentów działających na rakietę oraz aktualnych charakterystyk bezwładnościowych. Wyjściem z modelu jest pełen wektor stanu rakiety: prędkości liniowe i kątowe, położenie rakiety oraz kąty orientacji przestrzennej uzupełnione o przyspieszenia liniowe oraz macierz transformacji. Parametry niezbędne do symulacji modelu zostały podane w rozdziale następnym.



Rysunek 4.3: Schemat blokowy modelu symulacyjnego opracowany w środowisku Simulink

5 Badania poligonowe Rakietowej Platformy Badawczej

W ramach prac autorowi przypadło w udziale uczestnictwo w projekcie badawczo-rozwojowym o tytule „*Opracowanie gazodynamicznego modułu sterującego, precyzyjnego naprowadzania dla pocisku rakietowego*", finansowanym przez Narodowe Centrum Badań i Rozwoju w ramach umowy nr DOB-SZAFIR/03/B/002/01/2021, w ramach programu „*Rozwój nowoczesnych, przełomowych technologii służących bezpieczeństwu i obronności państwa*" – SZAFIR 1, który realizowany był w okresie 01.06.2021 r. – 31.05.2024 r. Celem projektu było wykonanie oraz badania demonstratora układu sterowania pociskiem rakietowym, który składa się z kilkudziesięciu rakietowych silników korekcyjnych na stały materiał pędny, zarówno od strony sprzętu jak i oprogramowania umożliwiającego precyzyjne naprowadzania na wcześniej zdefiniowany cel naziemny. Wyposażenie rakiety w taki układ powinno znacząco wpłynąć na poprawę jej celności i prawdopodobieństwa trafienia celu, co umożliwiłoby zmniejszenie niezbędnej ilości wystrzelonych rakiet, a więc również zmniejszenie kosztów operacyjnych działania jednostek wojskowych.

Opracowany moduł sterujący wraz z algorytmami nawigacji i sterowania wymagał przetestowania w warunkach poligonowych. W tym celu opracowano także Rakietową Platformę Testową (RPT) ORION. Jest to rakieta kalibru 122 mm o długości ponad 2 m o zasięgu około 10 km, co umożliwia jej testowanie na poligonach wojskowych dostępnych w kraju. W czasie trwania projektu udało się przeprowadzić łącznie 14 strzałów, które pozwoliły na dopracowanie konstrukcji oraz przetestowanie efektywności działania opracowanego modułu sterującego. Dodatkowo możliwe było przeprowadzenie weryfikacji i walidacji opracowanego modelu symulacyjnego. Weryfikacja miała na celu sprawdzenie czy przygotowany model symulacyjny odzwierciedlał zachowanie poruszającego się z dużą szybkością i wirującego pocisku rakietowego. Wymagał on dostarczenia danych geometrycznych, masowych, osiągowych i aerodynamicznych obiektu. Analizie podlegały trajektoria, orientacja oraz wartości prędkości liniowych i kątowych obiektu. Po pozytywnej weryfikacji, proces walidacji miał na celu sprawdzenie, czy symulowany obiekt zachowuje się jak badana rakieta ORION. Dodatkowo bazując na dostępnych danych telemetrycznych możliwe było dostrojenie parametrów obiektu, szczególnie aerodynamicznych, aby wynik symulacji modelu odpowiadał z wymaganą dokładnością danym uzyskanym z testów lotnych.

5.1 Weryfikacja modelu matematycznego

W próbach poligonowych brało udział łącznie 14 rakiet. Pierwsze sześć prób poligonowych miało na celu dopracowanie konstrukcji samej rakiety, a kolejne osiem testy modułu sterowania. Każda wyprodukowana rakieta była ważona, mierzony był jej śro-

dek ciężkości, a także główne momenty bezwładności. Po ustaleniu geometrii zewnętrznej docelowej konstrukcji wyznaczono jej charakterystyki aerodynamiczne. Zmierzono także parametry bezwładnościowe silników korekcyjnych, ich ciąg oraz ciąg silnika głównego. Na podstawie ostatnich ośmiu strzałów opracowano generyczny model rakiety ORION, uzupełniony o dane niezbędne do działania modelu symulacyjnego. Weryfikacja modelu miała na celu sprawdzenie, czy zachowanie się w locie symulowanego obiektu będzie odpowiadać oczekiwaniom. Jakościowym ocenom podlegały przebiegi trajektorii, prędkości liniowych i kątowych oraz kąty orientacji przestrzennej.

5.1.1 Charakterystyki bezwładnościowe

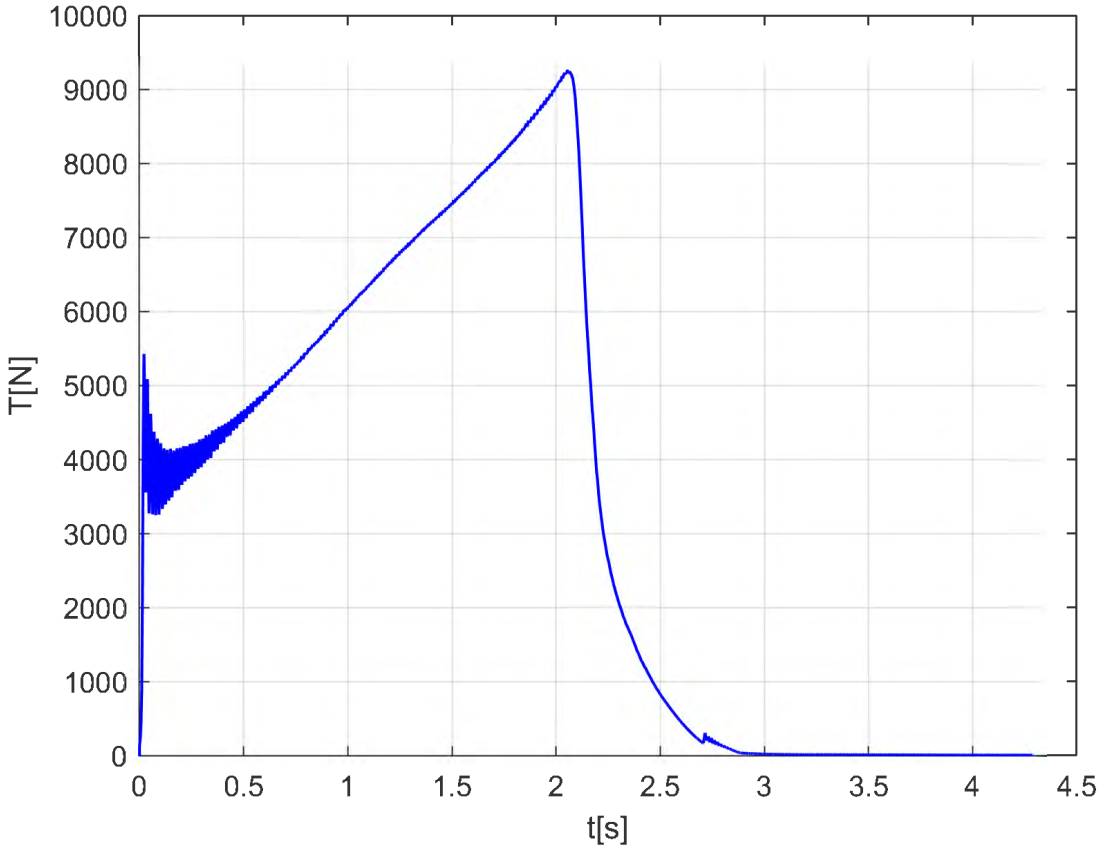
Masę, położenie środka ciężkości rakiety oraz jej macierz momentów bezwładności w chwili startu oraz po wypaleniu się materiału pędnego silnika głównego przedstawia Tabela 5.1. Część wartości zostało zmierzonych eksperymentalnie, takie jak masa startowa, masa materiału pędnego, podłużne położenie środka ciężkości oraz główne momenty bezwładności. Pozostałe dane zaczerpnięto z przygotowanego modelu CAD rakiety. Położenie środka ciężkości badane było poprzez zawieszenie rakiety na cienkiej lince w taki sposób, aby znalazła się ona w położeniu równowagi. Momenty bezwładności zmierzono za pomocą wahadła zawieszonego na czterech linkach.

Tabela 5.1: Charakterystyki bezwładnościowe rakiety ORION

Parametr	Symbol	Wartość	Jednostka	Źródło
Masa startowa	m_0	$31,74$	kg	Eksperyment
Masa materiału pędnego	m_p	$6,35$		Eksperyment
Położenie środka ciężkości w chwili startu	x_{cg_0}	$0,818$	m	Eksperyment
	y_{cg_0}	$-6,988e^{-4}$		Model CAD
	z_{cg_0}	$-1,775e^{-4}$		Model CAD
Położenie środka ciężkości po wypaleniu materiału pędnego	x_{cg_k}	$0,932$		Eksperyment
	y_{cg_k}	$-8,707e^{-4}$		Model CAD
	z_{cg_k}	$-2,212e^{-4}$		Model CAD
Momenty bezwładności w chwili startu	I_{xx_0}	$0,069$	kg·m^2	Model CAD
	I_{yy_0}	$9,223$		Eksperyment
	I_{zz_0}	$9,199$		Eksperyment
Momenty dewiacyjne w chwili startu	I_{xy_0}	$-5408,240e^{-6}$		Model CAD
	I_{yz_0}	$220,032e^{-6}$		Model CAD
	I_{xz_0}	$-183,511e^{-6}$		Model CAD
Momenty bezwładności po wypaleniu materiału pędnego	I_{xx_k}	$0,065$		Model CAD
	I_{yy_k}	$7,272$		Eksperyment
	I_{zz_k}	$7,312$		Eksperyment
Momenty dewiacyjne po wypaleniu materiału pędnego	I_{xy_k}	$-3638,069e^{-6}$		Model CAD
	I_{yz_k}	$219,130e^{-6}$		Model CAD
	I_{xz_k}	$266,195e^{-6}$		Model CAD

5.1.2 Parametry zespołu napędowego

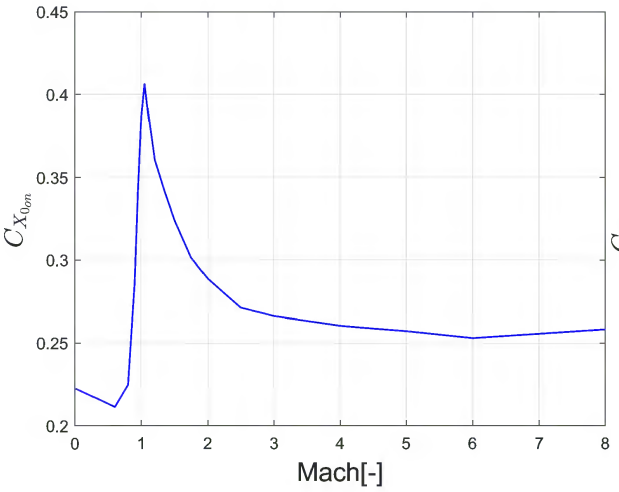
Ciąg silnika głównego został zmierzony eksperymentalnie, wykorzystując do tego specjalne stanowisko zwane hamownią. Przeprowadzono kilka eksperymentów, na podstawie których oszacowano przebieg ciągu. Rysunek 5.1 przedstawia uśrednioną i wygładzoną krzywą ciągu zależną od czasu. Założono, że w czasie eksperymentów ciśnienie otoczenia wynosiło $p_e = 101325$ Pa, a średnica wylotowa dyszy wynosi $d_e = 68$ mm.



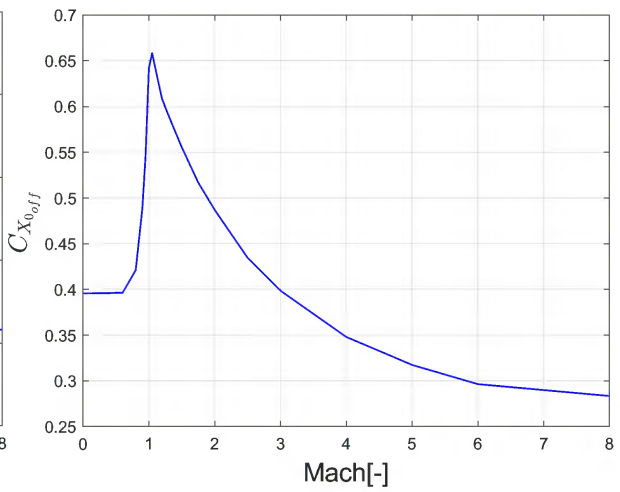
Rysunek 5.1: Ciąg silnika głównego w funkcji czasu

5.1.3 Charakterystyki aerodynamiczne

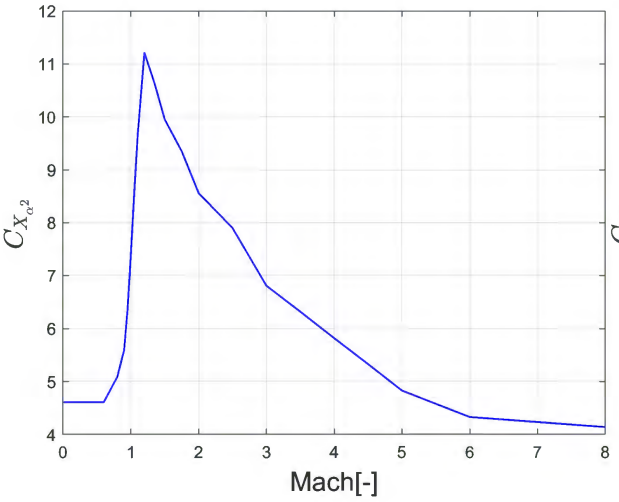
Charakterystyki aerodynamiczne, w postaci pochodnych, zostały obliczone jako funkcje liczby Macha. Wykorzystano w tym celu specjalne oprogramowanie bazujące tylko na geometrii zewnętrznej obiektu, które wykorzystuje w większości pół-empiryczne zależności w celu szacowania parametrów aerodynamicznych rakiet i pocisków. Współczynniki aerodynamiczne użyte w modelu przedstawia Rysunek 5.2. Wewnątrz symulacji są one interpolowane w zależności od aktualnej liczby Macha.



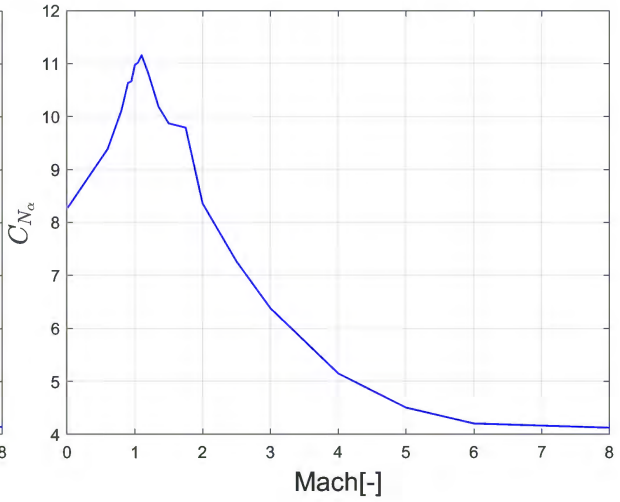
(a) Współczynnik oporu dla zerowego kąta natarcia przy włączonym silniku



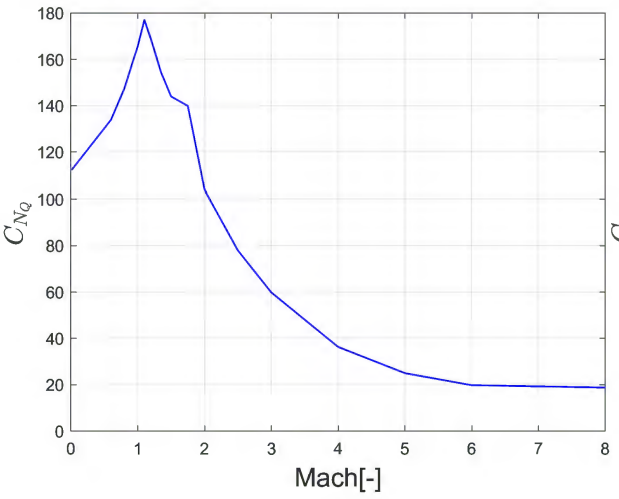
(b) Współczynnik oporu dla zerowego kąta natarcia przy wyłączonym silniku



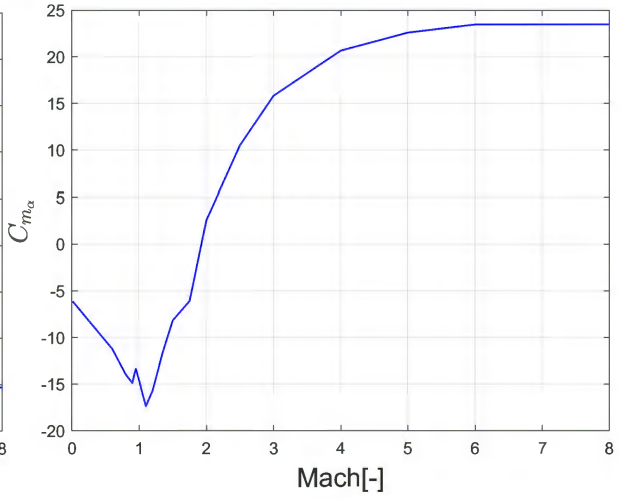
(c) Pochodna współczynnika oporu po kwadracie kąta natarcia



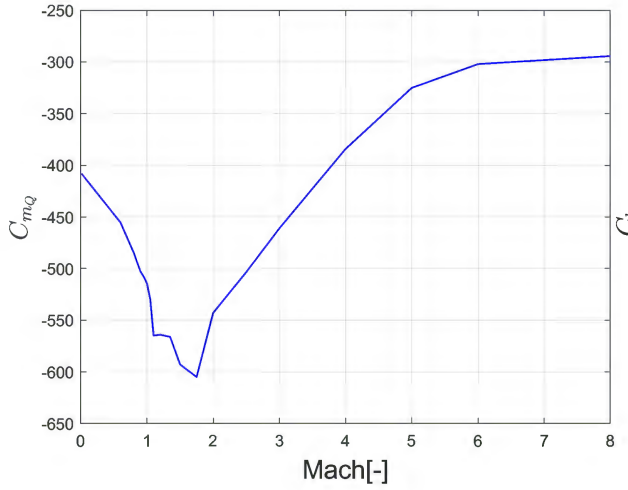
(d) Pochodna współczynnika siły normalnej po kącie natarcia



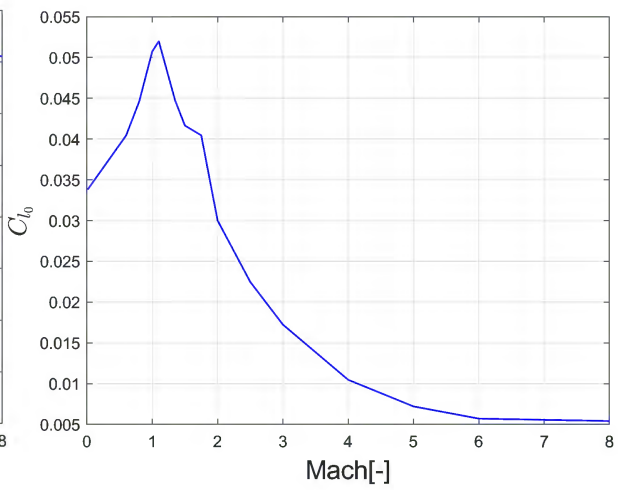
(e) Pochodna tłumiąca współczynnika siły normalnej po prędkości kątowej



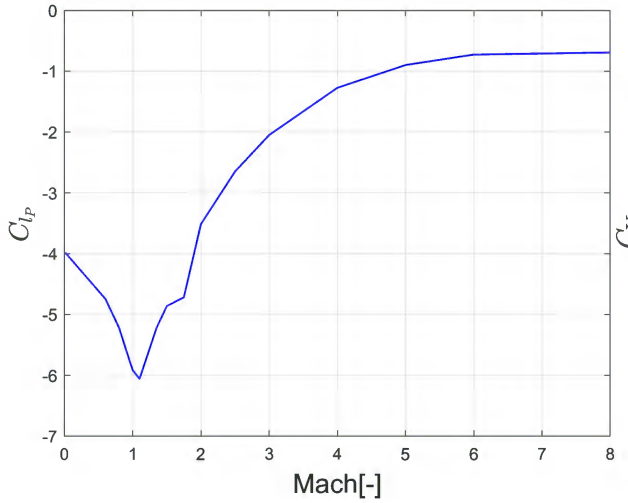
(f) Pochodna współczynnika momentu pochylającego po kącie natarcia



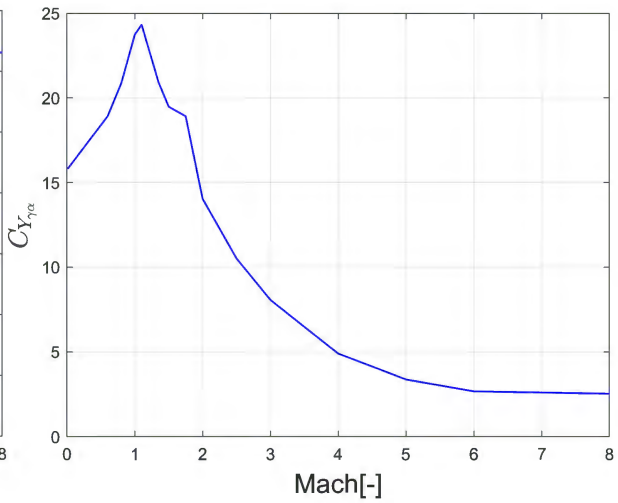
(g) Pochodna tłumiąca współczynnika momentu pochylającego po prędkości kątowej



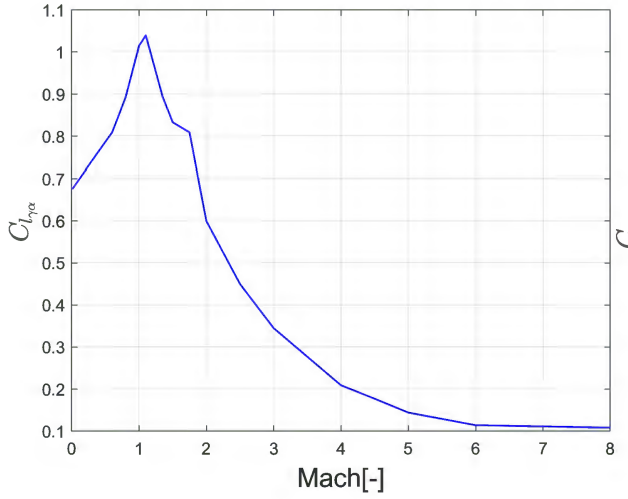
(h) Współczynnik momentu przechylającego wymuszający rotację



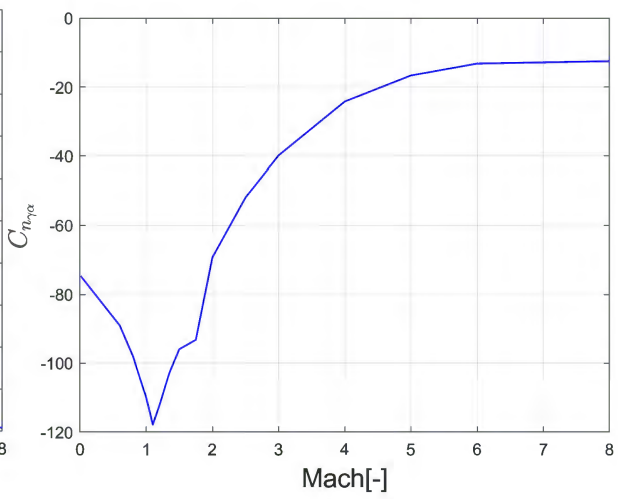
(i) Pochodna tłumiąca współczynnika momentu przechylającego po prędkości kątowej



(j) Pochodna współczynnika siły bocznej wyższego rzędu



(k) Pochodna współczynnika momentu przechylającego wyższego rzędu

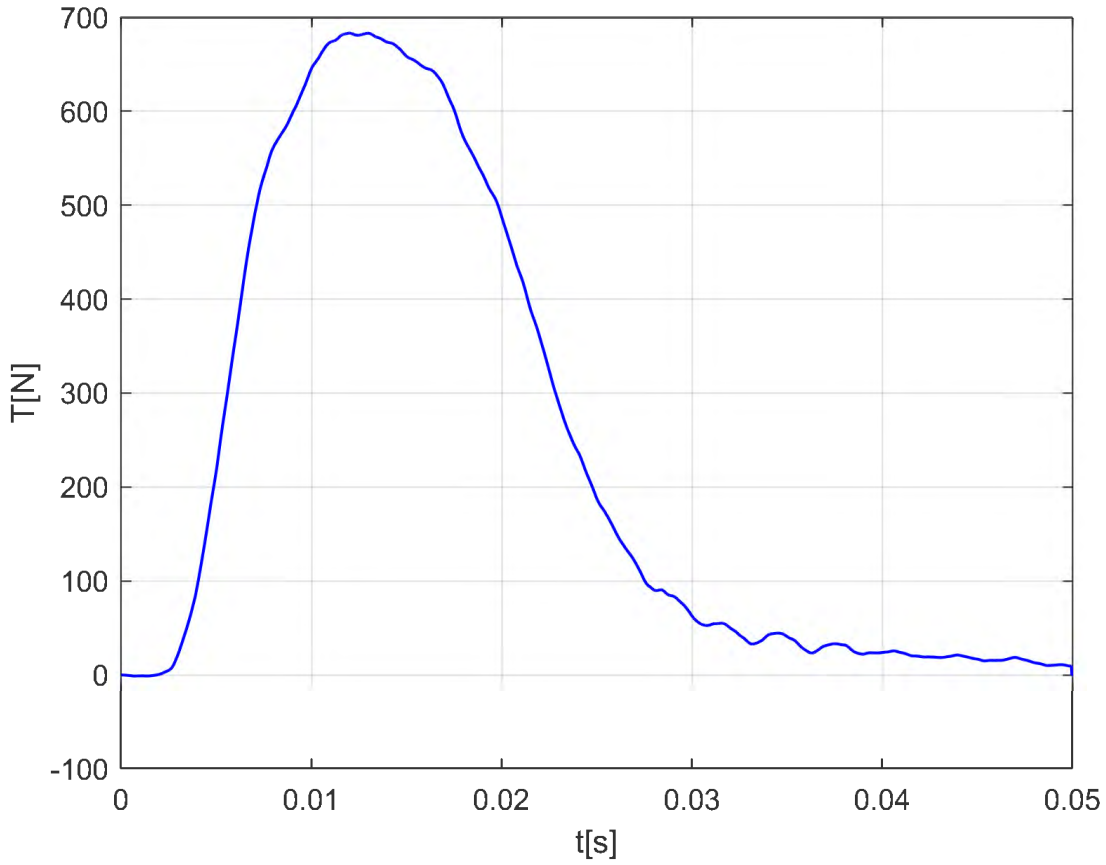


(l) Pochodna współczynnika momentu odchylającego wyższego rzędu

Rysunek 5.2: Współczynniki aerodynamiczne wykorzystywane w modelu symulacyjnym w funkcji liczby Macha

5.1.4 Parametry silników korekcyjnych

Charakterystyka ciągu pojedynczego silnika korekcyjnego w funkcji czasu przedstawiona została na Rysunku 5.3. Dane te uzyskano uśredniając wyniki z kilku pomiarów przeprowadzonych na hamowni. Przebieg ciągu charakteryzuje się stosunkowo dużą wartością maksymalną, sięgającą prawie 700 N, biorąc pod uwagę masę i wymiary ziarna, oraz bardzo krótkim czasem palenia, rzędu setnych części sekundy, co jest bardzo pożądaną cechą dla takich układów sterowania. Dzięki temu wypadkowa siła sterująca działająca na rakietę zakreśla niewielki kąt w czasie obrotu rakiety, co pozwala na osiągnięcie dużej precyzji kątowej strzału.



Rysunek 5.3: Ciąg silnika korekcyjnego w funkcji czasu

Rakietowa platforma testowa ORION wyposażona została w gazodynamiczny moduł sterujący składający się łącznie z 32 silniczków korekcyjnych równo rozmieszonych na obwodzie rakiety w czterech równoległych płaszczyznach po 8 w każdej. Każda z płaszczyzn była przesunięta kątowo względem pozostałych o $11,25$ stopnia. Numer każdego silniczka wraz z jego położeniem kątowym, położeniem płaszczyzny, w której się znajduje mierzonym od punktu referencyjnego (koniec dyszy rakiety) oraz kolejnością uruchamiania przedstawia Tabela 5.2.

Tabela 5.2: Położenie oraz kolejność uruchamiania silników korekcyjnych

Numer silniczka	Kolejność uruchamiania	Położenie kątowe w płaszczyźnie [deg]	Położenie płaszczyzny względem punktu referencyjnego [m]
1	1	0	1,1027
2	5	180	1,1027
3	3	90	1,1027
4	7	270	1,1027
5	2	45	1,1027
6	6	225	1,1027
7	4	135	1,1027
8	8	315	1,1027
9	9	22,5	1,1462
10	13	202,5	1,1462
11	11	112,5	1,1462
12	15	292,5	1,1462
13	10	67,5	1,1462
14	14	247,5	1,1462
15	12	157,5	1,1462
16	16	337,5	1,1462
17	17	11,25	1,1898
18	21	191,25	1,1898
19	19	101,25	1,1898
20	23	281,25	1,1898
21	18	56,25	1,1898
22	22	236,25	1,1898
23	20	146,25	1,1898
24	24	326,25	1,1898
25	25	33,75	1,2332
26	29	213,75	1,2332
27	27	123,75	1,2332
28	31	303,75	1,2332
29	26	78,75	1,2332
30	30	258,75	1,2332
31	28	168,75	1,2332
32	32	348,75	1,2332

Charakterystyki bezwładnościowe pojedynczego silnika korekcyjnego, w postaci masy, momentów bezwładności oraz położenia środka ciężkości względem osi rakiety, które umożliwiają uwzględnienie wpływu spalania materiału pędnego poszczególnych silników na bezwładność całej rakiety, uzyskane z pomiarów i uzupełnione danymi z modelu CAD przedstawia Tabela 5.3.

Tabela 5.3: Charakterystyki bezwładnościowe pojedynczego silnika korekcyjnego

Parametr	Symbol	Wartość	Jednostka	Źródło
Masa materiału pędnego	m_{sk}	$5,8$	g	Eksperyment
Odległość środka ciężkości ziarna od osi rakiety	r_{sk}	39	mm	Model CAD
Momenty bezwładności w układzie ziarna	$I_{xx_{sk}}$	$0,209e^{-6}$	kg·m^2	Model CAD
	$I_{yy_{sk}}$	$0,331e^{-6}$		Model CAD
	$I_{zz_{sk}}$	$0,331e^{-6}$		Model CAD

5.1.5 Parametry modelu czujników inercjalnych

Parametry czujników inercjalnych zostały wyznaczone na podstawie karty katalogowej czujników ADIS16467 użytych w rakiecie [135]. Przyjęto, że jednostka IMU znajduje się w osi rakiety w odległości $1,536$ m od punktu referencyjnego. Parametry wykorzystywane w modelu akcelerometrów przedstawia Tabela 5.4, a w modelu giroskopów Tabela 5.5. Dodatkowo przyjęto częstotliwości przekazywania parametrów nawigacyjnych przez układ INS, dla kątów orientacji przestrzennej było to 25 Hz, dla położenia i prędkości 5 Hz.

Tabela 5.4: Parametry modelu akcelerometrów

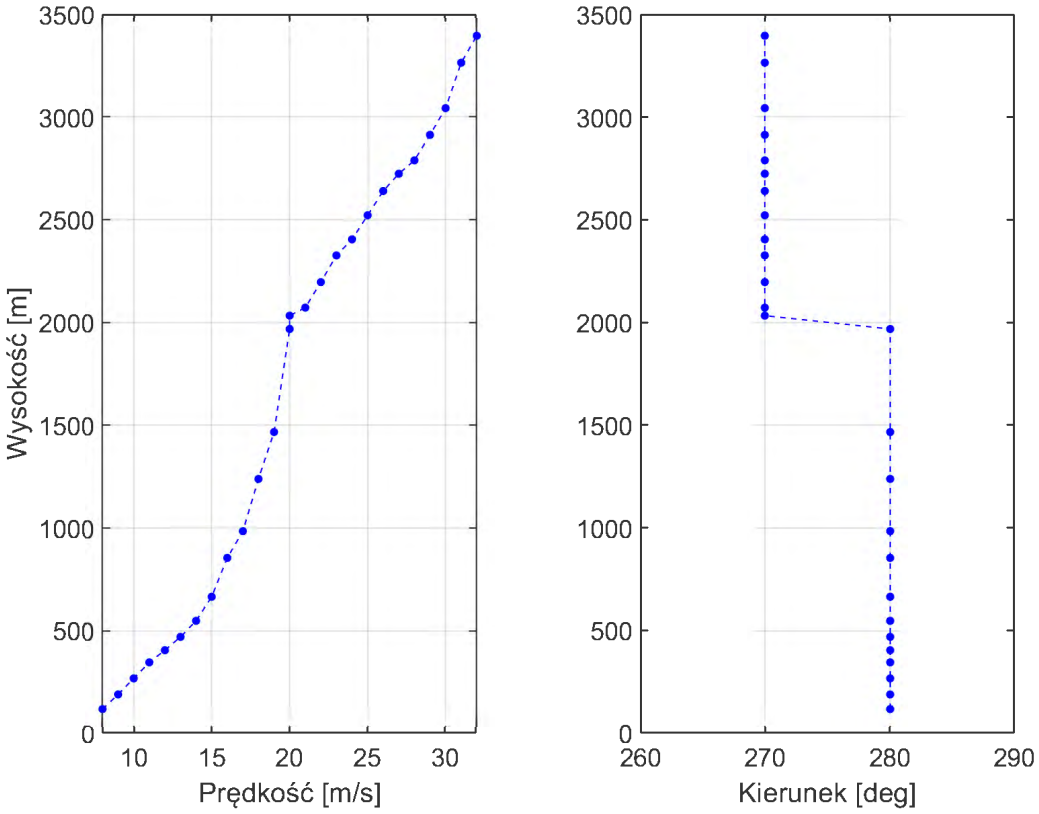
Parametr	Wartość	Jednostka
Częstość własna	$127,32$	rad/s
Współczynnik tłumienia	$0,698$	-
Błąd skali	1	m/s^2
Nieprostopadłość osi	$\sigma = 0.05$	deg
Przesunięcie zera	$\sigma = 13e^{-6}g_{acc}$	m/s^2
Saturacja	$40g_0$	m/s^2
Gęstość szumu	$1e^{-8}g_{acc}$	$(m/s^2)/\sqrt{Hz}$
Okres próbkowania	200	Hz
Kwantyzacja	32	Bit

Tabela 5.5: Parametry modelu giroskopów

Parametr	Wartość	Jednostka
Częstość własna	$116,71$	rad/s
Współczynnik tłumienia	$0,698$	-
Błąd skali	1	m/s^2
Nieprostopadłość osi	$\sigma = 0,05$	deg
Przesunięcie zera	$\sigma = 0,0017$	deg/s
Saturacja	2000	deg/s
Gęstość szumu oś X	$1,6e^{-6}$	$(deg/s)/\sqrt{Hz}$
Gęstość szumu oś Y i Z	$4,2e^{-6}$	$(deg/s)/\sqrt{Hz}$
Okres próbkowania	200	Hz
Kwantyzacja	32	Bit

5.1.6 Wyniki symulacji

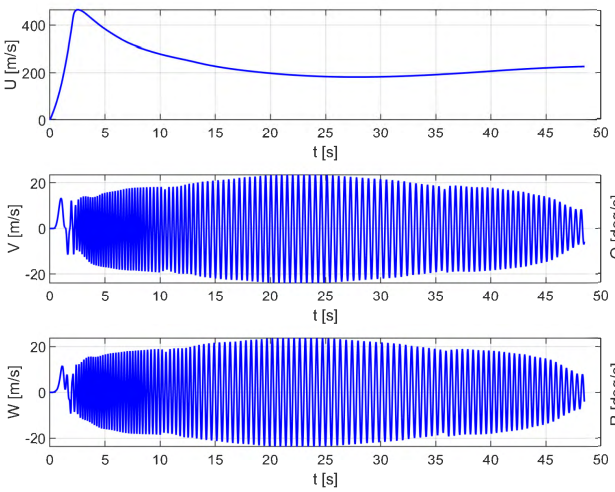
W celu przeprowadzenia weryfikacji modelu symulacyjnego, dane przedstawione w poprzednim rozdziale uzupełniono o parametry związane ze startem z wyrzutni, parametry środowiska oraz dane miejsca startu. Przyjęto, że długość szyny wyrzutni wynosi 5,94 m, a położenie ślizgaczy odpowiednio 0,04 m dla tylnego ślizgacza i 0,68 m dla przedniego, mierząc od punktu referencyjnego. Wyrzutnia znajdowała się w miejscu o współrzędnych 53,35648 stopni szerokości geograficznej, 15,63743 stopni długości geograficznej i wysokości nad elipsoidą WGS-84 wynoszącej 116 m. W czasie startu panowała temperatura 15^oC, ciśnienie 1023 hPa, a wilgotność powietrza wynosiła 50%. Pionowy profil wiatru przedstawia Rysunek 5.4.



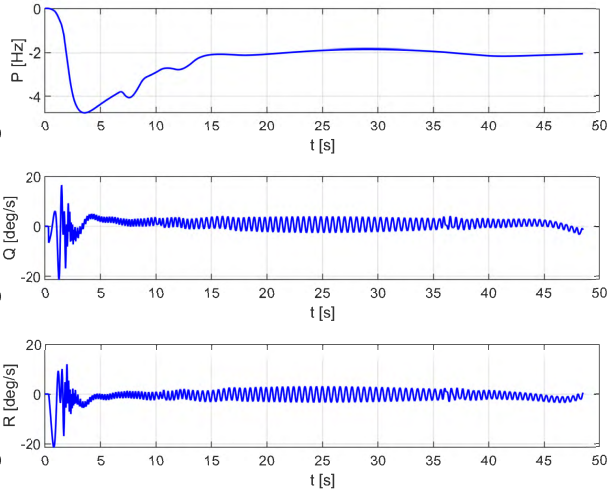
Rysunek 5.4: Pionowy profil wiatru

Po wprowadzeniu niezbędnych danych przeprowadzono symulację lotu rakiety. Wyniki symulacji przedstawia Rysunek 5.5. Część (a) pokazuje przebieg składowych prędkości linowej. Rakieta rozpędza się do ponad 460 m/s, a uderza w ziemię z prędkością około 230 m/s. Oscylację prędkości poprzecznych wynikają z ruchu obrotowego rakiety. Część (b) przedstawia prędkości kątowe. Maksymalna prędkość wirowania wyniosła ponad 4,7 Hz. Na części (c) widoczne są składowe położenia rakiety w układzie nawigacyjnym. Zasięg lotu wyniósł około 9300 m, a maksymalna wysokość lotu niecałe 3000 m. Kąty orientacji przestrzennej widoczne są na części (d). Rakieta skręca po starcie o około 10 stopni względem azymutu startowego, kąt strzału wynosił 45 stopni, a przyziemienie nastąpiło

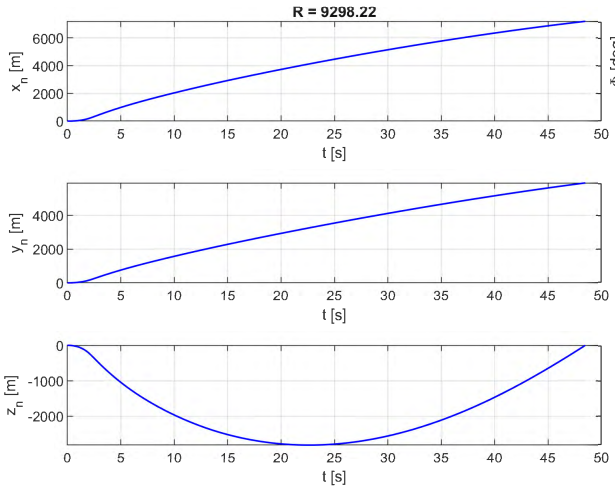
pod kątem około -60 stopni. Dwuwymiarowa oraz trójwymiarowa trajektoria rakiety widoczne są w części (e) oraz (f). Część (g) przedstawia zmianę parametrów atmosfery w czasie lotu wraz z prędkością Macha, która maksymalnie wyniosła około $1,4$. Kąty napływu przedstawia część (h). W częściach od (i) do (l) przedstawiono porównanie przyspieszeń, prędkości kątowych, składowych położenia oraz kątów orientacji przestrzennej zwracanych przez model nawigacji, w kolorze czerwonym, do danych wynikających z rozwiązania równań ruchu, w kolorze niebieskim. Można zauważyć dość dobrą zgodność wyników. Analizując otrzymane wyniki zauważyć można, że trajektoria rakiety odpowiada krzywej balistycznej, a przebiegi prędkości liniowych, kątowych i kątów orientacji przestrzennej odpowiadają lotowi wirującego obiektu o dużej prędkości. Przedstawione wyniki symulacji potwierdzają poprawność zaimplementowanego modelu rakiety.



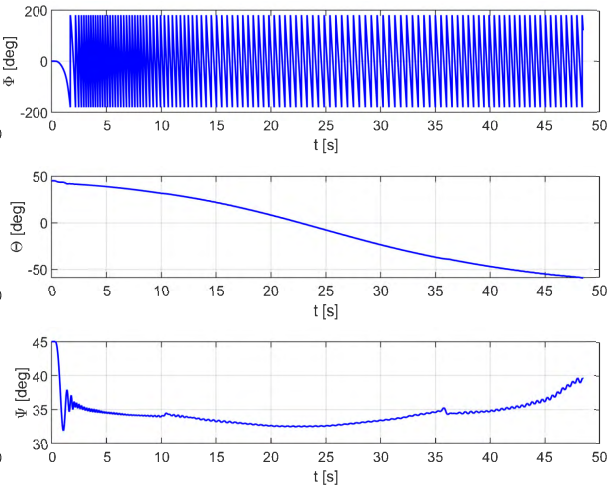
(a) Składowe prędkości liniowej



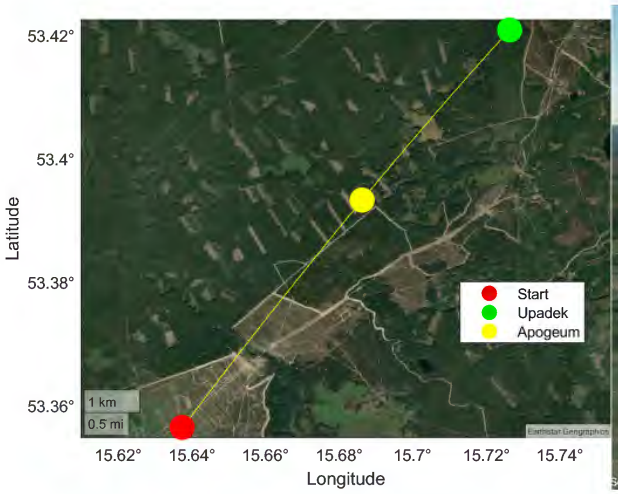
(b) Składowe prędkości kątowej



(c) Składowe położenia

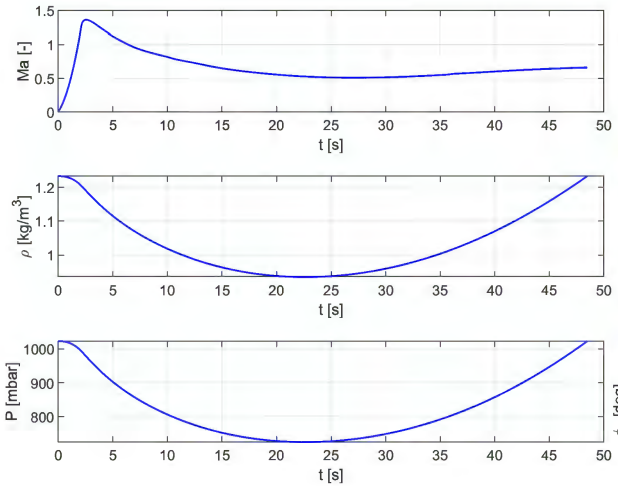


(d) Kąty orientacji przestrzennej

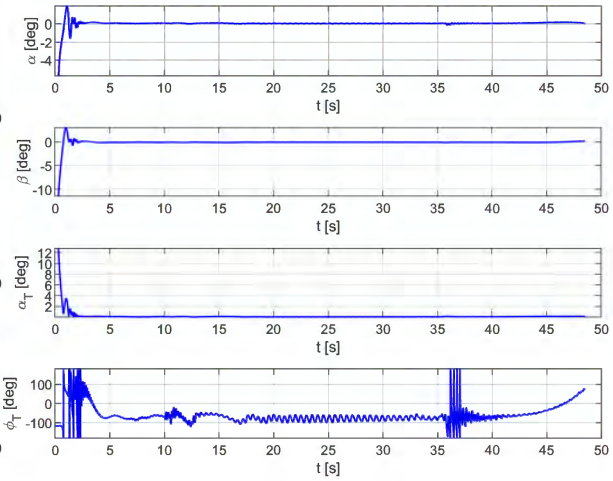


(e) Mapa lotu

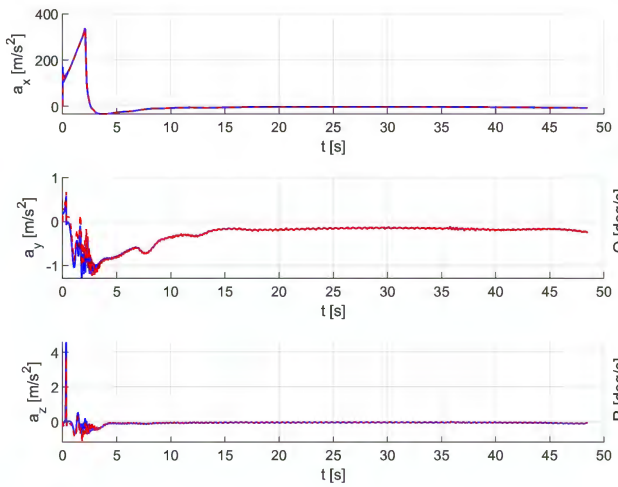
(f) Trajektoria lotu



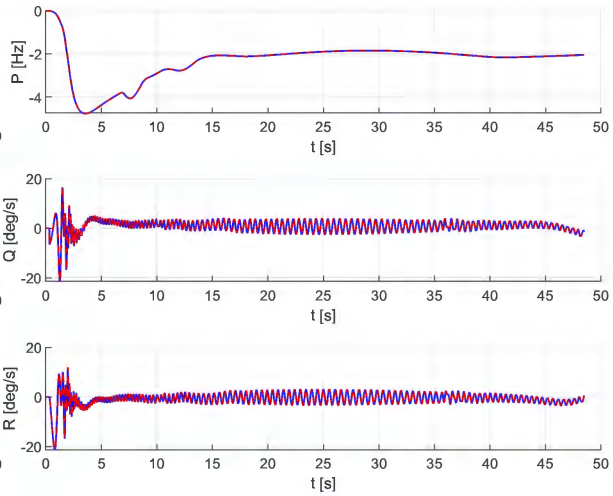
(g) Parametry atmosfery



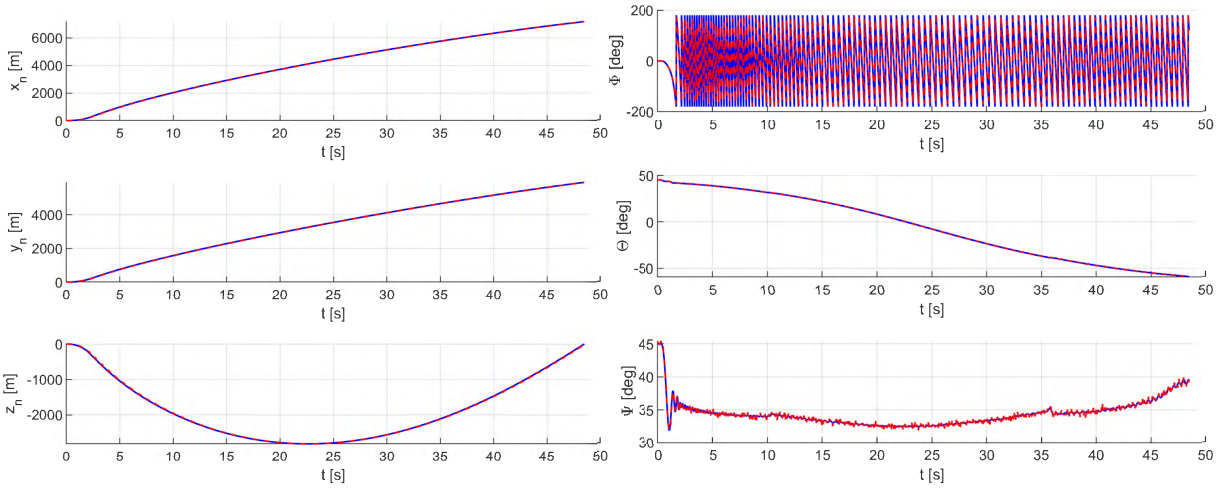
(h) Kąty napływu



(i) Składowe przyspieszenia liniowego - porównanie z modelem nawigacji



(j) Składowe prędkości kątowej - porównanie z modelem nawigacji



(k) Składowe położenia - porównanie z modelem nawigacji

(l) Kąty orientacji przestrzennej - porównanie z modelem nawigacji

Rysunek 5.5: Wyniki symulacji weryfikującej poprawność modelu

5.2 Opracowanie bazowego algorytmu sterującego

Na potrzeby projektu, w celu przetestowania opracowanego modułu sterującego, należało opracować algorytm sterowania i naprowadzania na nieruchomy cel naziemny. Zdecydowano się algorytm oprzeć o klasyczne rozwiązania w postaci wariantu nawigacji proporcjonalnej. Dokładny opis algorytmu oraz metodyki jego testowania w celu zastosowania na rakiecie można znaleźć w [136]. Opracowany algorytm składa się z trzech głównych części, ekstrapolacji danych nawigacyjnych, algorytmu naprowadzania opartego o nawigację proporcjonalną oraz algorytmu wyboru silników korekcyjnych. Wejściami do algorytmu są czas, prędkości liniowe w układzie nawigacyjnym, składowe położenia w układzie nawigacyjnym, prędkości kątowe oraz kąty orientacji przestrzennej. W pierwszej kolejności algorytm uwzględnia fakt, że dane o prędkości odnoszą się do punktu zamocowania układu pomiarowego i przelicza je do położenia środka ciężkości rakiety zgodnie ze wzorem:

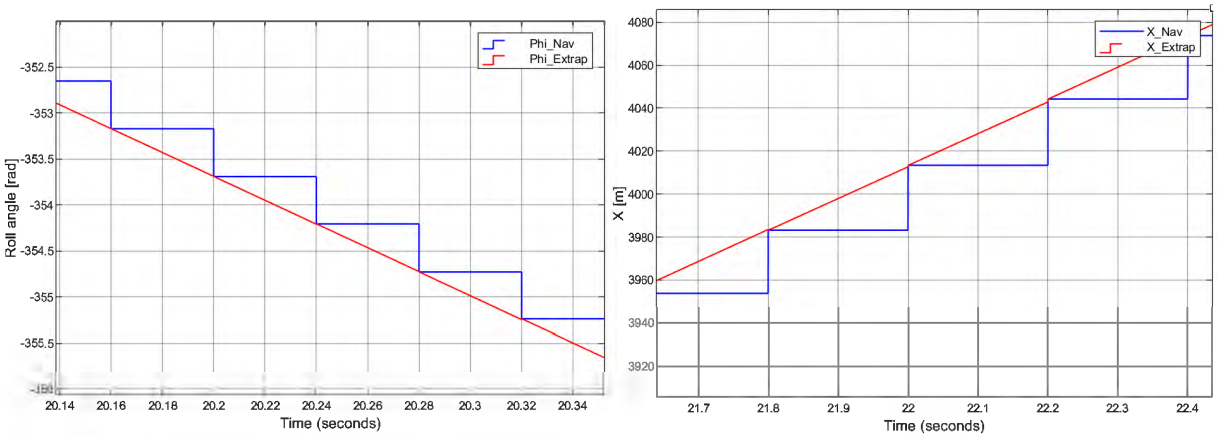
$$\mathbf{v}_{N_{CG}} = \mathbf{v}_{N_{IMU}} + \boldsymbol{\omega} \times (\mathbf{r}_{CG} - \mathbf{r}_{IMU}) \tag{5.1}$$

gdzie prędkości $\mathbf{v}_{N_{CG}}$ i $\mathbf{v}_{N_{IMU}}$ odnoszą się odpowiednio do środka ciężkości i środka układu pomiarowego w układzie nawigacyjnym, $\mathbf{r}_{CG}$ jest położeniem środka ciężkości, które jest wczytywane do algorytmu jako tabela składowych położenia w funkcji czasu otrzymana na podstawie symulacji, a $\mathbf{r}_{IMU}$ jest położeniem układu pomiarowego. Duży wpływ na dokładność działania całego algorytmu ma dokładność zadawania kierunku odpalenia silnika korekcyjnego. Aby ją zachować wymagana jest bardzo duża częstotliwość przesyłania danych nawigacyjnych, która nie może być zapewniona przez sam układ nawigacji. Z tego powodu posługując się prostą ekstrapolacją liniową zwiększana jest częstotliwość infor-

macji o kącie przechylenia oraz składowych położenia rakiety zgodnie ze wzorami:

$$\Phi_{k+1} = \Phi_k + T_s \cdot P_k \tag{5.2}$$
$$X_{k+1} = X_k + T_s \cdot U_{n_k} \tag{5.3}$$
$$Y_{k+1} = Y_k + T_s \cdot V_{n_k} \tag{5.4}$$
$$Z_{k+1} = Z_k + T_s \cdot W_{n_k} \tag{5.5}$$

gdzie T_s jest krokiem czasowym wykonywania algorytmu, a U_n, V_n, W_n są składowymi prędkości środka masy w układzie nawigacyjnym. Algorytm ten zakłada, że prędkości kątowe i liniowe nie zmieniają się pomiędzy krokami algorytmu i pozwala bardzo dokładnie wygładzić przebiegi kąta przechylenia i składowe położenia jak przedstawia Rysunek 5.6. Kolorem niebieskim zaznaczone są wartości sygnałów z układu nawigacji, a kolorem czerwonym wartości po ekstrapolacji. Zauważyć można wyraźną dyskretyzację sygnałów niebieskich spowodowaną niską częstotliwością dostarczania informacji nawigacyjnych, a także wygładzone przebiegi sygnałów ekstrapolowanych.



(m) Kąt przechylenia (n) X składowa położenia

Rysunek 5.6: Porównanie danych nawigacyjnych przed i po algorytmie ekstrapolacji

Algorytm nawigacji proporcjonalnej w trzech wymiarach można znaleźć w [137, 138, 139]. Bazuje on na założeniu, że rakieta osiągnie swój cel jeśli orientacja linii wizowania (linii łączącej rakietę z celem) względem pewnego układu odniesienia nie będzie się zmieniać w czasie. Na podstawie względnego położenia celu $\mathbf{R}_{TM} = \begin{bmatrix} X_{TM} & Y_{TM} & Z_{TM} \end{bmatrix}^T$ wyznacza się kierunek niezbędnego przyspieszenia normalnego do linii wizowania, aby rakieta leciała do przewidywanego punktu spotkania z celem. Przyspieszenie to opisuje wzór:

$$n_{cmd_{ij}} = NV_{c_{ij}}\dot{\lambda_{ij}} \tag{5.6}$$

gdzie $i, j = x, y, z$, N jest stałą nawigacji proporcjonalnej, $\mathbf{V}_c = \begin{bmatrix} V_{c_{xy}} & V_{c_{xz}} & V_{c_{yz}} \end{bmatrix}^T$ jest

prędkością zbliżania, a $\dot{\boldsymbol{\lambda}} = \begin{bmatrix} \dot{\lambda}_{xy} & \dot{\lambda}_{xz} & \dot{\lambda}_{yz} \end{bmatrix}^T$ jest prędkością kątową linii wizowania. Przyspieszenie to wyliczane jest osobno w każdej płaszczyźnie układu nawigacyjnego. Składowe prędkości zbliżania mogą być wyznaczone ze wzorów:

$$V_{c_{xy}} = \frac{X_{TM}U_n + Y_{TM}V_n}{\sqrt{X_{TM}^2 + Y_{TM}^2}} \tag{5.7}$$

$$V_{c_{xz}} = \frac{X_{TM}U_n + Z_{TM}W_n}{\sqrt{X_{TM}^2 + Z_{TM}^2}} \tag{5.8}$$

$$V_{c_{yz}} = \frac{Y_{TM}V_n + Z_{TM}W_n}{\sqrt{Y_{TM}^2 + Z_{TM}^2}} \tag{5.9}$$

Kąty linii wizowania z odpowiednimi płaszczyznami układu nawigacyjnego wyrażają wzory:

$$\lambda_{xy} = \text{atan2}\left(Y_{TM}, X_{TM}\right) \tag{5.10}$$

$$\lambda_{xz} = \text{atan2}\left(Z_{TM}, X_{TM}\right) \tag{5.11}$$

$$\lambda_{yz} = \text{atan2}\left(Z_{TM}, Y_{TM}\right) \tag{5.12}$$

Pochodne tych kątów względem czasu można obliczyć z zależności:

$$\dot{\lambda_{xy}} = -\frac{X_{TM}V_n - Y_{TM}U_n}{X_{TM}^2 + Y_{TM}^2} \tag{5.13}$$

$$\dot{\lambda_{xz}} = -\frac{X_{TM}W_n - Z_{TM}U_n}{X_{TM}^2 + Z_{TM}^2} \tag{5.14}$$

$$\dot{\lambda_{yz}} = -\frac{Y_{TM}W_n - Z_{TM}V_n}{Y_{TM}^2 + Z_{TM}^2} \tag{5.15}$$

Następnie przyspieszenie normalne do linii wizowania z równania (5.6) musi zostać przetransformowane do układu rakiety oraz skompensowane o grawitację za pomocą zależności:

$$\mathbf{a}_{cmd} = \mathbf{C}_N^B \left(\mathbf{C}_{LOS}^N \mathbf{n}_{cmd} - \mathbf{g}\right) \tag{5.16}$$

gdzie $\mathbf{C}_{LOS}^N$ jest macierzą transformacji z układu związanego z linią wizowania do układu nawigacyjnego daną zależnością:

$$\mathbf{C}_{LOS}^N = \begin{bmatrix} -\sin\left(\lambda_{xy}\right) & -\sin\left(\lambda_{xz}\right) & 0 \\ \cos\left(\lambda_{xy}\right) & 0 & -\sin\left(\lambda_{yz}\right) \\ 0 & \cos\left(\lambda_{xz}\right) & \cos\left(\lambda_{yz}\right) \end{bmatrix} \tag{5.17}$$

Następnym krokiem algorytmu jest sprawdzenie szeregu warunków odpowiadających za odpalenie silników korekcyjnych. Są to:

1. Dany silnik korekcyjny nie został jeszcze wykorzystany

2. Czas, który upłynął od chwili uruchomienia ostatniego silnika korekcyjnego t_{last} jest

dłuższy niż wartość $\tau \in (0; \infty)$

$$t - t_{last} > \tau \tag{5.18}$$

3. Silnik korekcyjny musi być uruchomiony w takiej chwili czasu, aby wypadkowa siła ciągu była skierowana w kierunku zadanego przemieszczenia bocznego, co znaczy, że wartość bezwzględna z różnicy fazy błędu γ i kąta ϕ_{sk_i}, na którym znajduje się dany silnik korekcyjny pomniejszona o kąty wyprzedzenia sterowania była mniejsza lub równa pewnej wartości progowej γ_t:

$$|\gamma - \phi_{sk_i} - \pi - P\left(\tau_d + \tau_{sk}\right)| \leq \gamma_t \tag{5.19}$$

gdzie τ_d jest opóźnieniem spłonki silnika, τ_{sk} jest czasem, po którym silnik wykorzysta połowę swojej energii, $\gamma = \text{atan2}\left(a_{cmd_y}, a_{cmd_z}\right)$ jest zadanym kierunkiem działania.

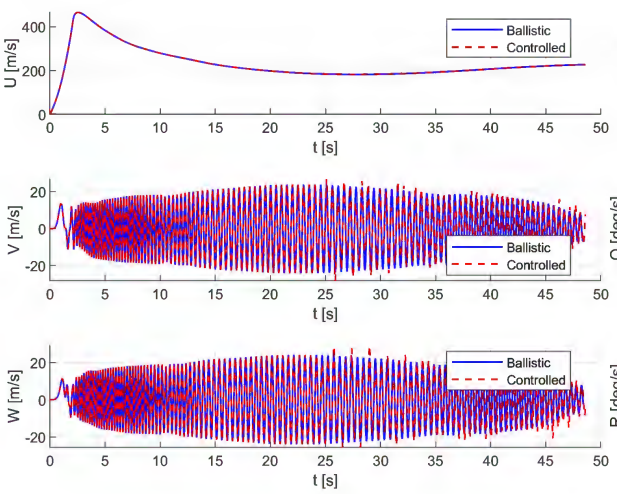
4. Zadziałanie układu sterowania następuje, gdy wartość kąta pochylenia pocisku Θ jest mniejsza lub równa zadanej wartości Θ_g oraz gdy czas t mierzony od początku lotu jest większy lub równy t_g:

$$\Theta \leq \Theta_g \wedge t \geq t_g \tag{5.20}$$

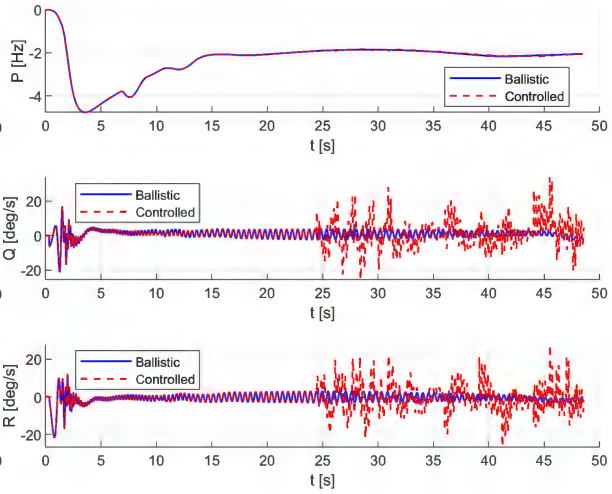
5. Zadane przyspieszenie boczne osiągnie pewną wartość progową a_{th}:

$$\sqrt{\left(a_{cmd_y}^2 + a_{cmd_z}^2\right)} \geq a_{th} \tag{5.21}$$

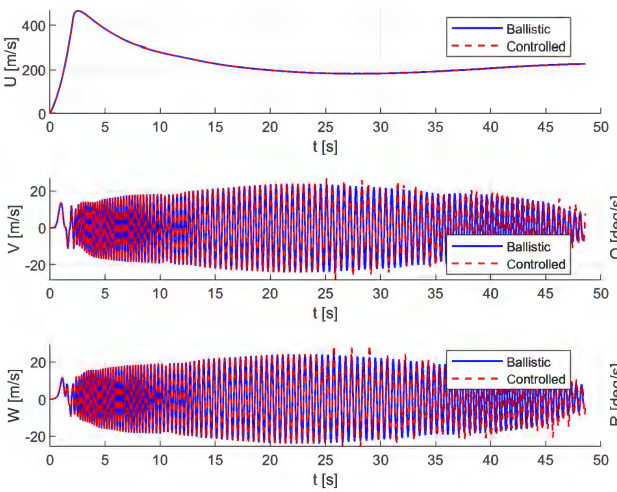
Algorytm wyboru silników korekcyjnych został dokładnie opisany w pozycjach [140, 141]. Każdy silnik korekcyjny posiada swój numer oraz kąt na obwodzie rakiety, na którym się znajduje, a także kolejność uruchomienia, zgodnie z Tabelą 5.2. Idąc zgodnie z tą kolejnością dla każdego silnika sprawdzane są warunki opisane wyżej. Jeśli wszystkie warunki są spełnione dane silnik zostaje zaznaczony jako uruchomiony, a algorytm przechodzi do sprawdzania tych warunków dla kolejnego silnika z tabelki. Uruchomiony silnik zaczyna produkować ciąg zgodny Rysunkiem 5.3 po czasie równym opóźnieniu spłonki. Po implementacji algorytmu w środowisku Simulink przeprowadzono symulację jego działania. W tym celu wygenerowano trajektorię referencyjną, której ostatni punkt przyjęto jako cel trafienia, a następnie zaburzono początkowe kąty strzału o kilka stopni i ponownie puszczono symulację z włączonym algorytmem sterowania. Wyniki lotu sterowanego przedstawiono na Rysunku 5.7.



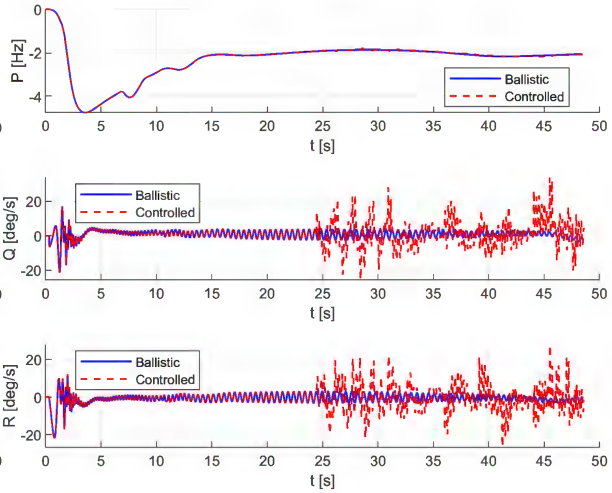
(a) Składowe prędkości liniowej



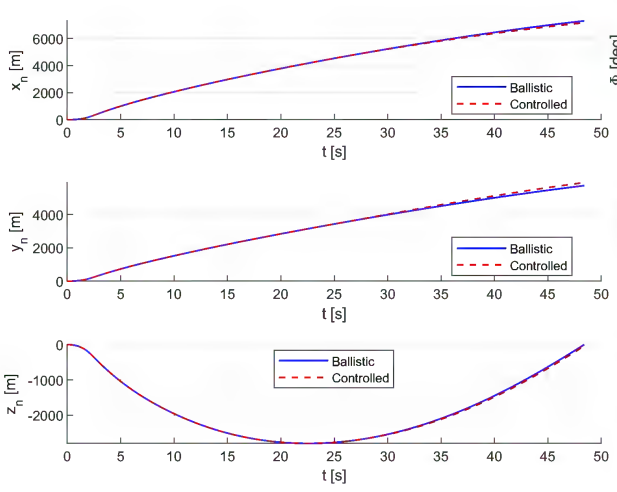
(b) Składowe prędkości kątowej



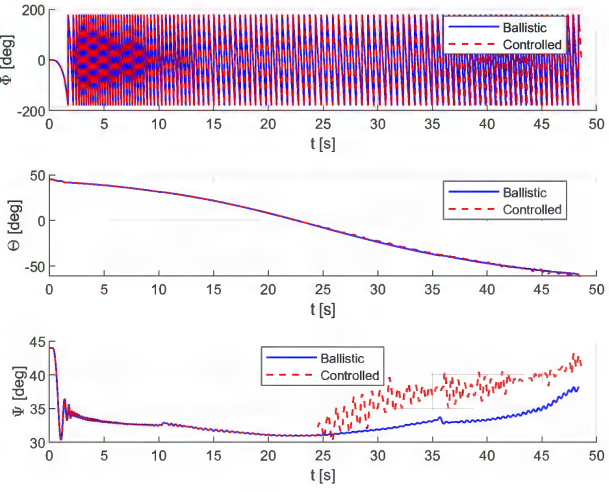
(c) Składowe prędkości liniowej



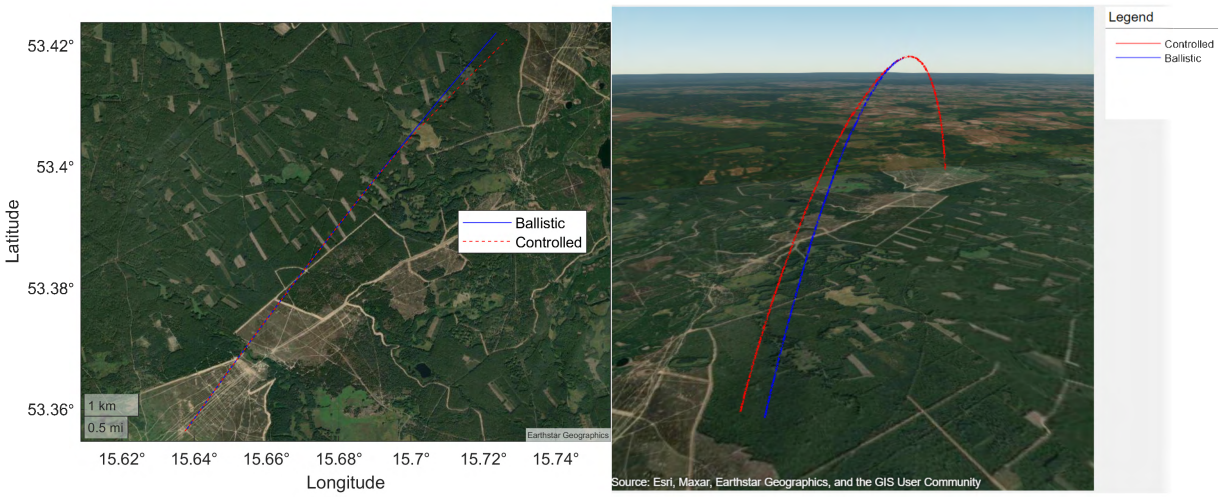
(d) Składowe prędkości kątowej



(e) Składowe położenia



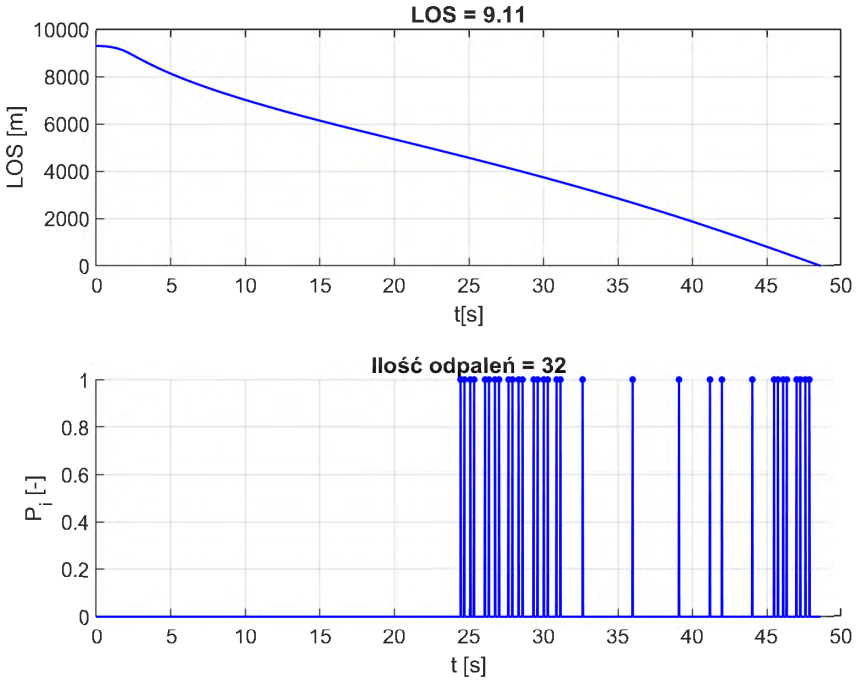
(f) Kąty orientacji przestrzennej



(g) Mapa lotu (h) Trajektoria lotu

Rysunek 5.7: Wyniki symulacji weryfikującej poprawność implementacji algorytmu sterowania

Kolorem niebieskim oznaczono na wykresach lot niesterowany, a kolorem czerwonym sterowany. Porównano przebiegi prędkości liniowych (a), prędkości kątowych (b), składowych położenia (c) oraz kątów orientacji przestrzennej (d). Największy wpływ działania układu sterowania widać na wykresie prędkości kątowych, gdzie pojawiają się charakterystyczne oscylację wynikające z reakcji rakiety na kolejne odpalenia silniczków. Widoczna jest również zmiana kąta odchylenia rakiety spowodowana naprowadzaniem się na cel. Zmianę trajektorii wynoszącą względem lotu niesterownego około 200 m, pokazują mapy przedstawione w częściach (e) i (f). Wynikowy błąd trafienia wyniósł nieco ponad 9 m przy użyciu wszystkich dostępnych silników korekcyjnych. Wykres przedstawiający odległość do punktu trafienia w czasie lotu oraz historię odpaleń kolejnych silników korekcyjnych przedstawia Rysunek 5.8.

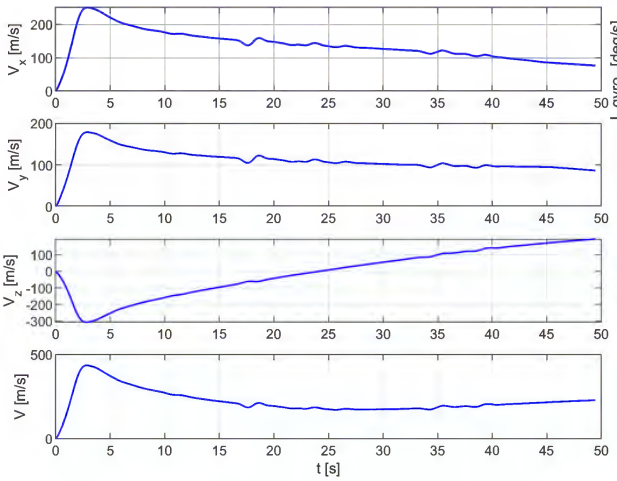


Rysunek 5.8: Przebieg odległości do celu w czasie (góra) oraz historia odpaleń silników korekcyjnych (dół)

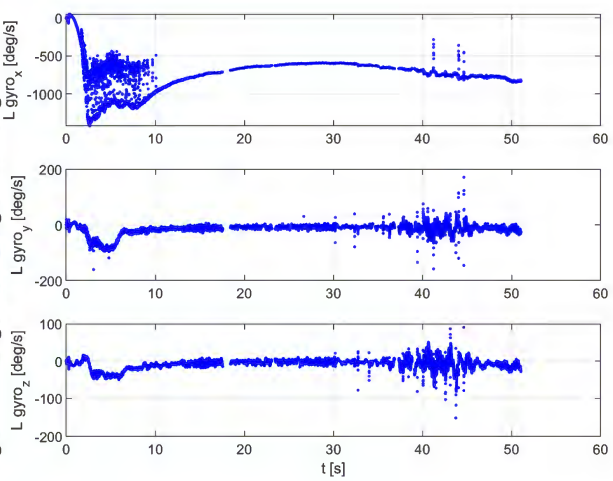
5.3 Walidacja modelu symulacyjnego

Po pozytywnej weryfikacji implementacji modelu rakiety, próby poligonowe zostały wykorzystane także w celu jego walidacji. Testy przeprowadzano na dwóch poligonach wojskowych: w Ośrodku Szkolenia Poligonowego Wojsk Lądowych w Żaganiu oraz w Centrum Szkolenia Wojsk Lądowych w Drawsku Pomorskim. Opisywanym próbom podlegały dwa egzemplarze rakiety ORION o numerach 011, testowana w Drawsku, oraz 014, testowana w Żaganiu. Rakietowa platforma testowa ORION wyposażona została w radiowy moduł telemetrii, który na bieżąco w trakcie lotu przesyłał parametry lotu rakiety. Celem lotu nr 011 było sprawdzenie efektywności odsterowania rakiety w jednym kierunku. W tym celu wykorzystano uproszczony algorytm sterowania, bez algorytmu naprowadzania, którego zadaniem było wystrzelenie wszystkich silników korekcyjnych w ustaloną wcześniej jedną, lewą patrząc wzdłuż rakiety, stronę. Wyrzutnię ustawiono pod kątem strzału wynoszącym 45 stopni w elewacji, a azymut ustawiono w taki sposób, aby lot odbywał się wzdłuż pasa taktycznego. Na przewidywanej trasie lotu ustawiono 4 maszty telemetryczne, które miały za zadanie zebrać jak najwięcej danych wysyłanych przez rakietę w czasie lotu. Sterowanie miało zacząć się mniej więcej po osiągnięciu przez rakietę apogeum, które wypadało w okolicach 30 sekundy lotu. Lot rakiety nr 014, który odbył się na poligonie w Żaganiu, testował pełen algorytm sterowania przy czym cel trafienia celowo znajdował się poza zasięgiem rakiety. Wyrzutnia była tym razem ustawiona pod kątem 58 stopni w elewacji ze względu na krótszy obszar poligonu. Azymut lotu został ustawiony tak, aby rakieta spadła w ustalonym przez dowódcę poligonu obszarze. Dane uzyskane z obu prób

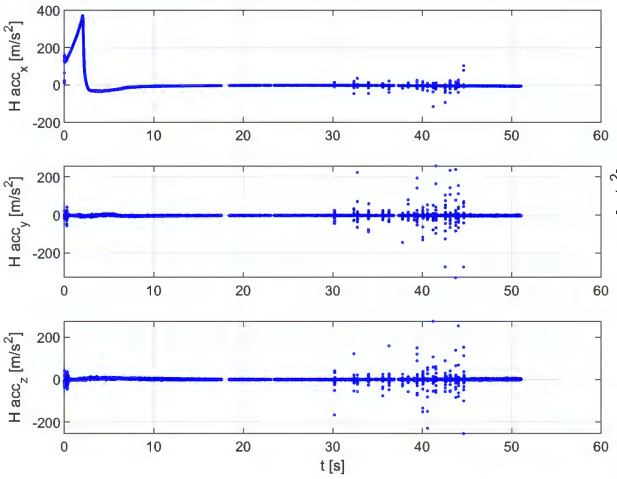
poligonowych przedstawiają Rysunek 5.9 oraz Rysunek 5.10.



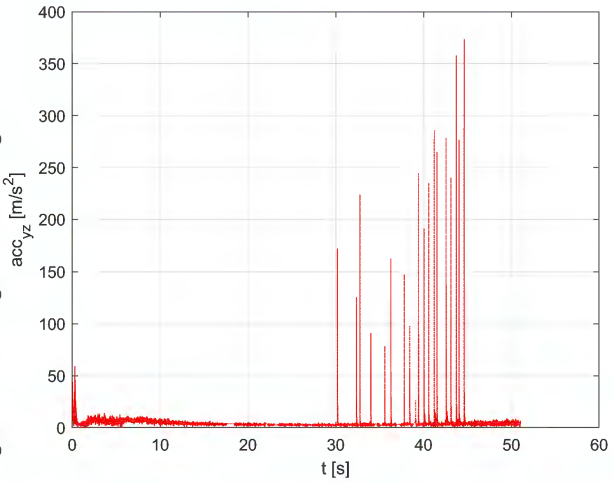
(a) Prędkość w układzie nawigacyjnym



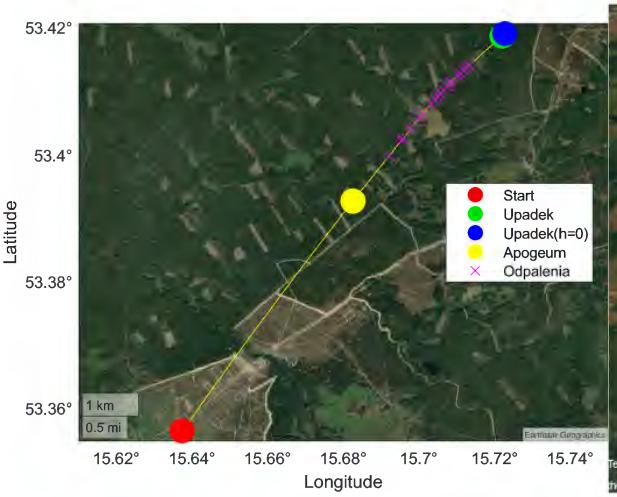
(b) Prędkości kątowe



(c) Przyspieszenia liniowe



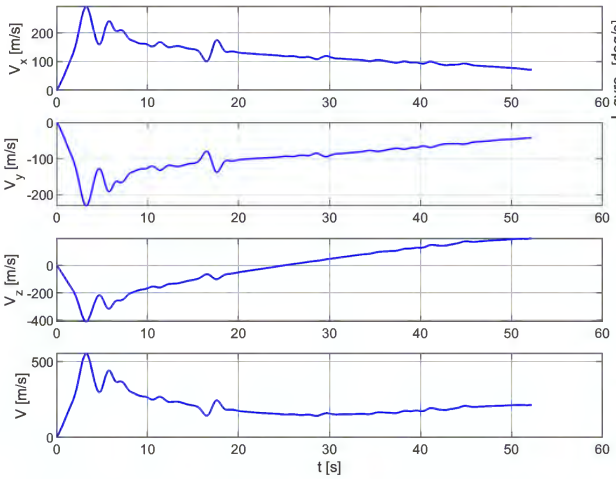
(d) Moduł przyspieszeń poprzecznych



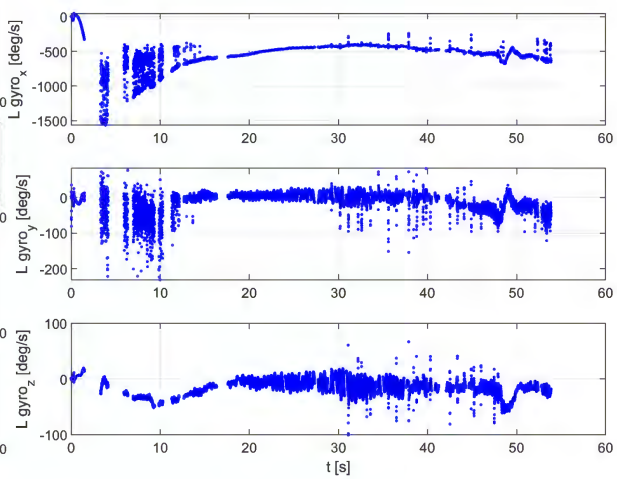
(e) Mapa lotu

(f) Trajektoria lotu

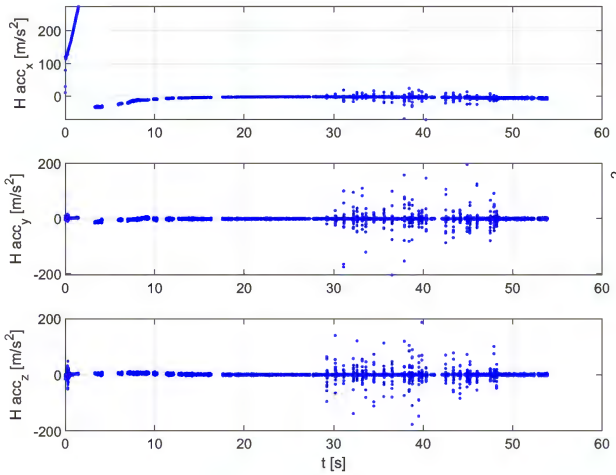
Rysunek 5.9: Dane telemetryczne z lotu RPT ORION nr 011



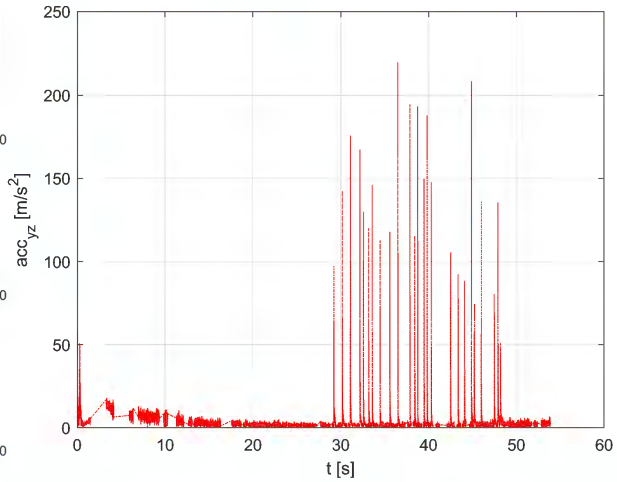
(a) Prędkość w układzie nawigacyjnym



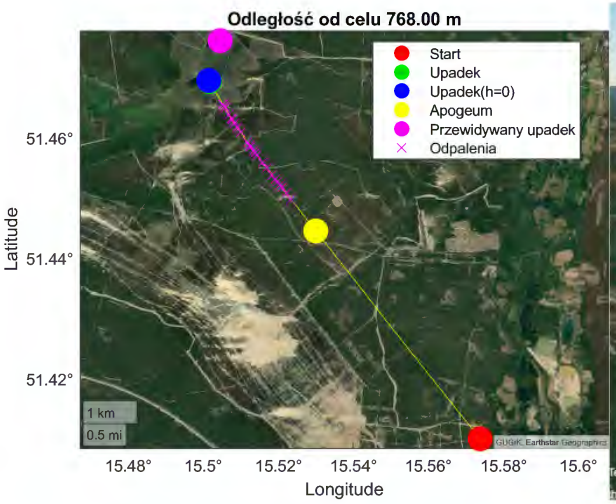
(b) Prędkości kątowe



(c) Przyspieszenia liniowe



(d) Moduł przyspieszeń poprzecznych



(e) Mapa lotu

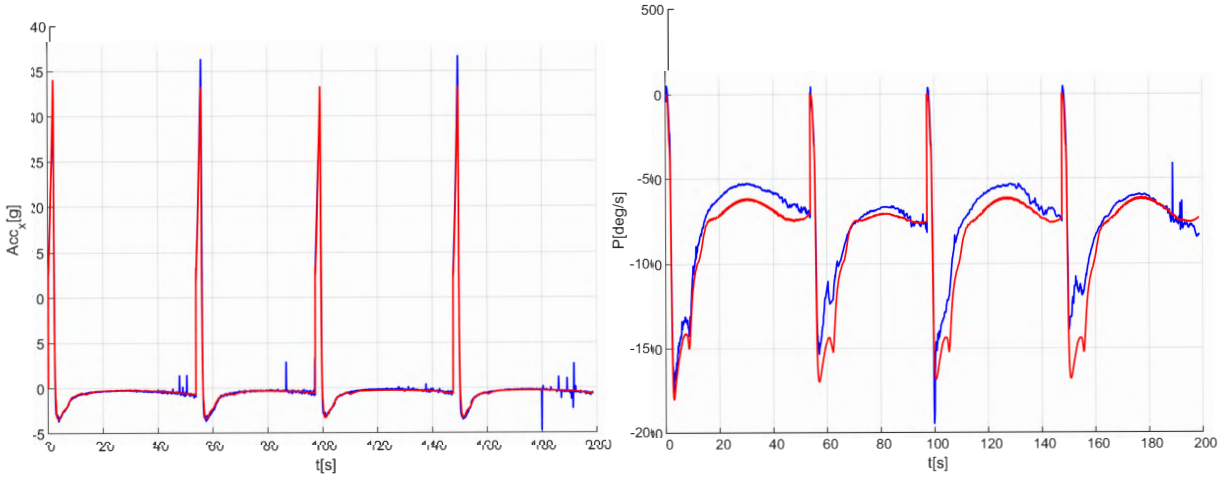
(f) Trajektoria lotu

Rysunek 5.10: Dane telemetryczne z lotu RPT ORION nr 014

Część (a) obu rysunków przedstawia prędkości liniowe w układzie nawigacyjnym, które zostały oszacowane na podstawie trajektorii GPS. W obu przypadkach prędkość

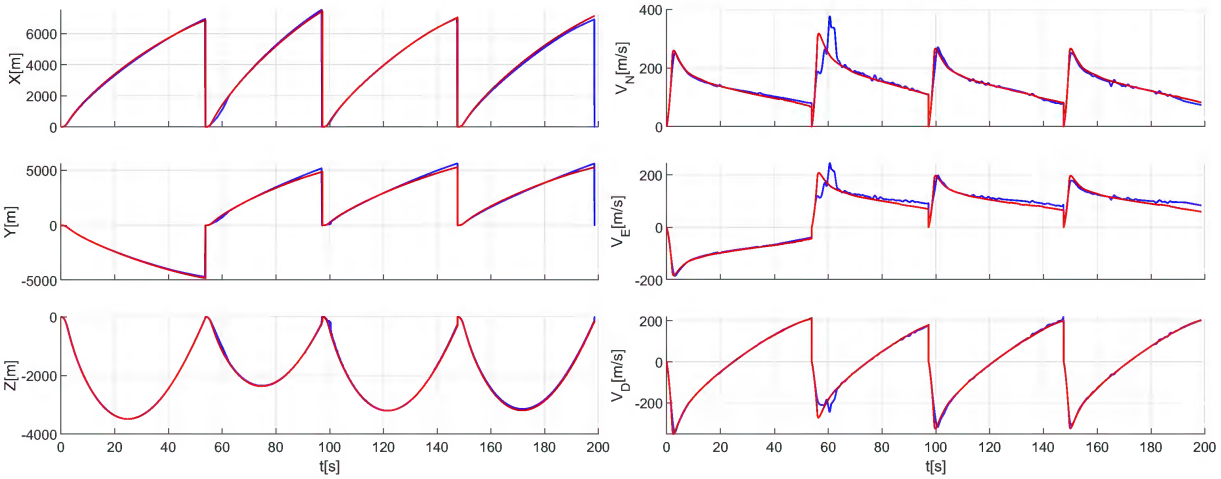
maksymalna wyniosła około 500 m/s. W drugim locie oscylację na wykresie prędkości spowodowane są utratą części ramek telemetrycznych, która miała duży wpływ na dokładność szacowania prędkości. Część (b) pokazuje prędkości kątowe. Można zauważyć charakterystyczne oscylacje podczas działania układu sterowania. Również widoczna jest utrata ramek, szczególnie w pierwszej części lotu, a także dość duży szum giroskopów, szczególnie widoczny w kanale przechylenia. Na części (c) oraz (d) widoczne są przyspieszenia liniowe. Charakterystyczne piki widoczne są w miejscach odpaleń silników korekcyjnych. Części (e) oraz (f) przedstawiają trajektorie rakiet naniesione na mapę wraz z punktami odpaleń silników korekcyjnych. Widoczne jest na nich ugięcie trajektorii pod wpływem działania układu sterowania. Obie próby zakończyły się pełnym sukcesem, układy sterowania zadziały w poprawny sposób, a uzyskane z lotów dane posłużyły do poprawy jakości i dokładności modelu symulacyjnego.

Parametrami, które bardzo wpływają na osiągi rakiety oraz jej zachowanie w locie, a które jest stosunkowo trudno otrzymać są charakterystyki aerodynamiczne. Kilka ostatnich lotów zostało wykorzystanych w celu identyfikacji niektórych parametrów aerodynamicznych. W tym celu wykorzystano autorskie oprogramowanie stworzone na bazie książki [142]. Oprogramowanie to implementuje metodę błędu wyjścia oraz metodę błędu filtra pozwalając na dopasowanie parametrów modelu, głównie charakterystyk aerodynamicznych, w taki sposób, aby wyjścia symulacji odpowiadały danym eksperymentalnym. Wyniki działania oprogramowania można znaleźć w artykule [143]. Do identyfikacji wybrano cztery loty z najlepszym pod względem jakości zestawem danych telemetrycznych. Szacowaniu podlegały głównie współczynnik oporu oraz momentu przechylającego jako funkcje liczby Macha, ale również niektóre inne współczynniki. Porównywano wartości przyspieszenia podłużnego, prędkości kątowej przechylania, składowych położenia oraz prędkości w układzie nawigacyjnym. Wyniki identyfikacji przedstawia Rysunek 5.11.



(a) Przyspieszenia liniowe

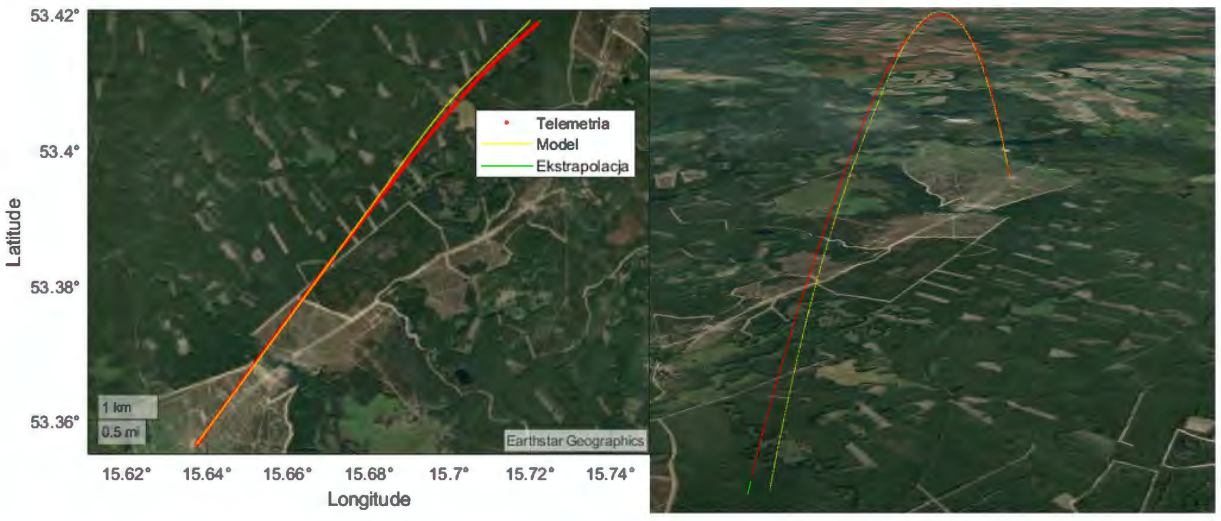
(b) Prędkości kątowe



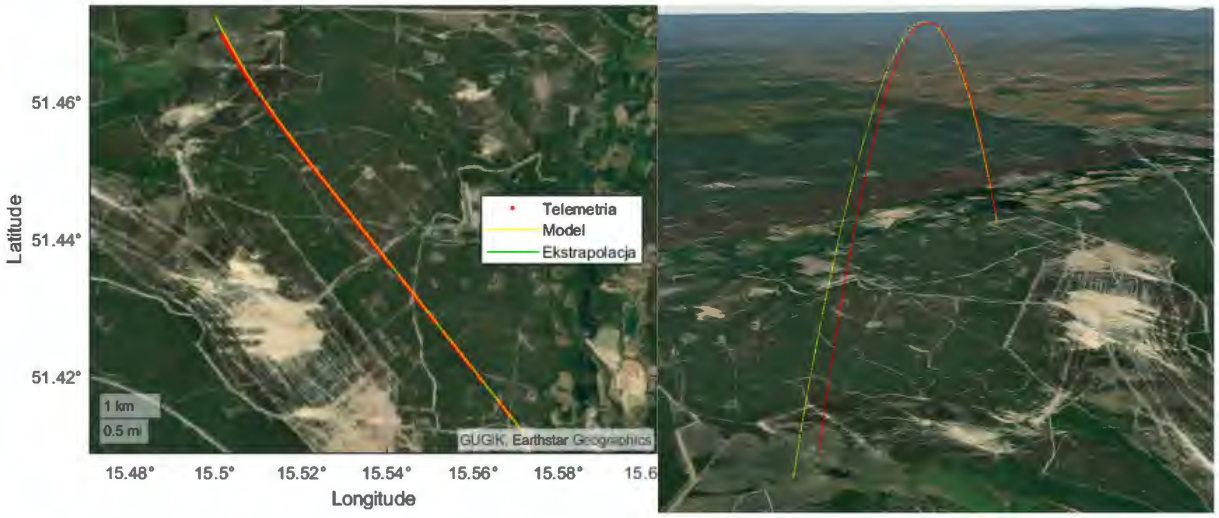
(c) Składowe położenia w układzie nawigacyjnym (d) Prędkości liniowe w układzie nawigacyjnym

Rysunek 5.11: Wyniki dopasowania modelu po identyfikacji charakterystyk aerodynamicznych

Część (a) Rysunku 5.11 przedstawia przyspieszenie liniowe podłużne. Widoczny na osi X czas jest to złączony czas wszystkich czterech zestawów danych. Charakterystyczne piki przedstawiają moment działania silnika głównego. Część (b) przedstawia prędkość kątową przechylania, a części (c) i (d) odpowiednio składowe położenia i prędkości w układzie nawigacyjnym. Kolorem niebieskim oznaczone są dane telemetryczne, a kolorem czerwonym odpowiedź modelu. Po procesie identyfikacji widoczne jest dość dobre dopasowanie obu krzywych na wszystkich wykresach. Proces ten pozwolił dostroić współczynniki aerodynamiczne modelu w taki sposób, aby model jeszcze lepiej potrafił przewidywać zachowanie rakiety w locie. Po wprowadzeniu zaktualizowanych danych do modelu symulacyjnego przeprowadzono symulacje mające na celu sprawdzenie dokładności odwzorowania rzeczywistej trajektorii rakiety przez model. Na Rysunku 5.12 przedstawiono mapy trajektorii z telemetrii oraz z symulacji z nowymi współczynnikami aerodynamicznymi.



(a) Dane dla lotu RPT ORION nr 011



(b) Dane dla lotu RPT ORION nr 014

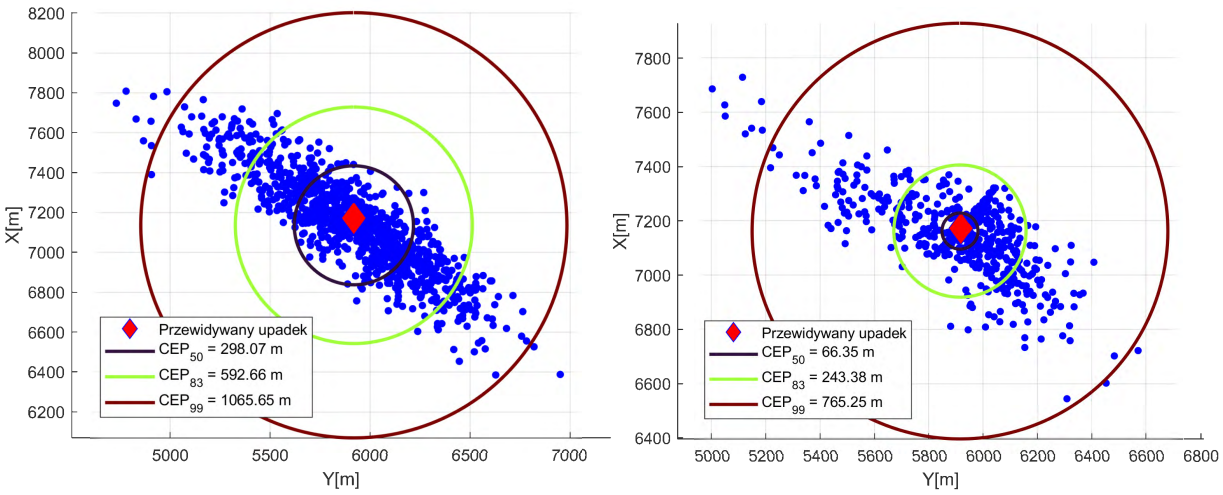
Rysunek 5.12: Porównanie trajektorii rzeczywistej z symulacją po procesie identyfikacji

Rysunek 5.12 (a) przedstawia porównanie trajektorii z zapisów telemetrii oraz z modelu. Odległość punktu lądowania z modelu od punktu lądowania z telemetrii wynosi $176,53$ m. Czas lotu różni się o $-0,14$ s. Różnica maksymalnej prędkości postępowej wyniosła $10,7$ m/s, a maksymalnej prędkości kątowej $-0,77$ Hz. Maksymalne przyspieszenie startowe różniło się o $3,48$ g. Symulacja przewidziała apogeum wyższe o $53,11$ m, a zasięg mniejszy o $113,58$ m. W przypadku tego strzału trajektoria z telemetrii schodzi z kursu trochę wcześniej niż przewidywał model, natomiast ponownie ugięcie trajektorii jest bardzo zbliżone dla obu przypadków. Model przewidział nieco mniejszy zasięg rakiety. Trzeba wziąć pod uwagę, że w czasie tego strzału wiał dość silny wiatr, którego dokładnej prędkości i kierunku nie dało się uwzględnić w modelu. Dla danych z Rysunku 5.12 (b) odległość punktu lądowania z modelu od punktu lądowania z telemetrii wynosi $324,14$ m. Czas lotu różni się o $-0,06$ s. Różnica maksymalnej prędkości postępowej wyniosła $54,21$ m/s, a maksymalnej prędkości kątowej $-0,48$ Hz. Maksymalne przyspieszenie startowe różniło się o $-6,89$ g. Symulacja przewidziała apogeum niższe o $15,28$ m, a zasięg mniejszy o $309,75$ m. Obie trajektorie bardzo dobrze pokazują zagięcie trajektorii wynikające z działania układu sterowania. Obie trajektorie praktycznie się pokrywają w rzucie z góry. Analizując wyniki obu porównań można uznać, że dokładność modelu jest satysfakcjonująca, biorąc pod uwagę trudność szacowania wiatru oraz stosunkowo niewielką ilość strzałów pozwalającą na dokładniejszą identyfikację modelu. Stwierdzono więc, że walidacja przygotowanego modelu wraz danymi przebiegła poprawnie.

5.4 Wyznaczenie efektywności modułu sterującego

Mając do dyspozycji dokładny model rakiety wraz z danymi wejściowymi oraz opracowany i przetestowany algorytm naprowadzania i sterowania, opisany w rozdziale

5.2, przystąpiono do wyznaczenia możliwości redukcji dyspersji miejsc upadku rakiety. Na podstawie pomiarów charakterystyk bezwładnościowych ostatnich ośmiu jednakowych konstrukcji, pomiarów ciągu silnika głównego oraz silniczków korekcyjnych oszacowano niepewności tych parametrów pod przeprowadzenie symulacji Monte Carlo. Dodatkowo oszacowano także niepewności charakterystyk aerodynamicznych, niepewności dla profilu prędkości i kierunku wiatru oraz wyznaczono rzeczywiste odchylenie kątów zejścia z wyrzutni względem jej początkowego ustawienia. Symulacje Monte Carlo polegały na puszczeniu wielu przypadków, w każdym z nich losując parametry modelu korzystając z odpowiedniego rozkładu, najczęściej normalnego, na podstawie przyjętych odchyleń standardowych. Przeprowadzono 1000 symulacji dla przypadku niesterowanego oraz sterowanego z użyciem opisanego wyżej algorytmu sterowania. Wyznaczono dyspersję dla rakiety dysponującej 32 silnikami korekcyjnymi. Wyniki przedstawiono na Rysunku 5.13.



(a) Dyspersja dla lotów niesterowanych

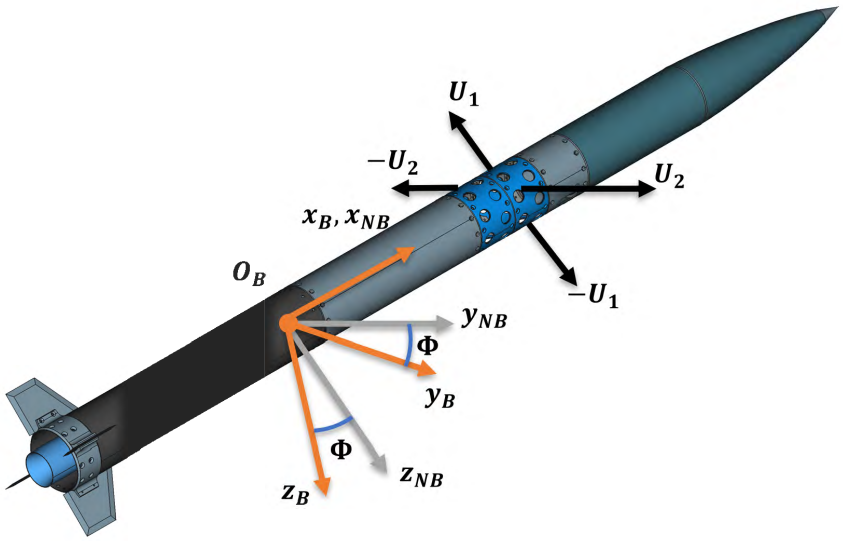
(b) Dyspersja dla lotów sterowanych

Rysunek 5.13: Dyspersja miejsc upadku

Część (a) Rysunku 5.13 przedstawia rozrzuty dla przypadku niesterowanego, a część (b) dla przypadku sterowanego. Czerwony diament jest celem trafienia, niebieskie punkty oznaczają miejsca upadku każdej z 1000 rakiet, a kolorowe okręgi oznaczają parametr zwany kołowym błędem trafienia CEP_x (ang. Circular Error Probable). Jest to promień okręgu w który zawiera się $x\%$ trafień. Najszerzej spotykaną miarą są CEP_{50} oraz CEP_{83}. Dodatkowo pokazano także wartość CEP_{99} traktując ją jako wszystkie trafienia, z pominięciem wartości wyraźnie ostających. Można zauważyć dla przypadku (a), promień CEP_{50} wyniósł około 300 m, CEP_{83} około 600 m, a CEP_{99} ponad 1 km. Przy wykorzystaniu 32 silniczków z algorytmie sterującym udało się osiągnąć redukcję wartości CEP_{50} o prawie 80%, wartości CEP_{83} o około 60%, a CEP_{99} o niecałe 30%. Wartość CEP_{50} znajduje się poniżej 100 metrów co uznaje się za dobry rezultat. Celem algorytmu opracowanego w niniejszej pracy będzie otrzymanie mniejszych lub porównywalnych wartości błędów CEP zachowując lub zmniejszając liczbę wykorzystanych silników korekcyjnych.

6 Optymalizacja

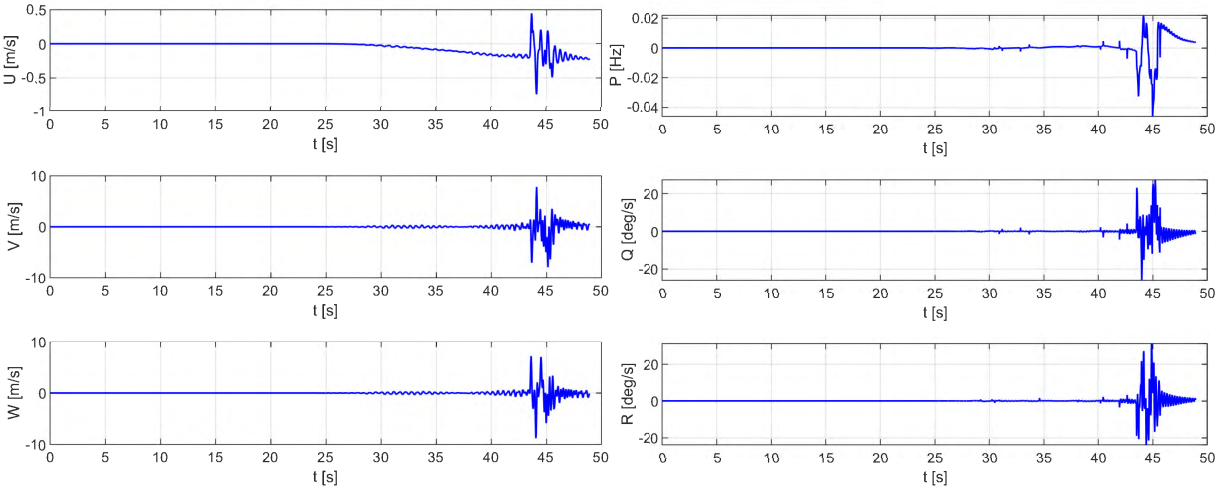
Mając opracowany, zweryfikowany i zwalidowany model symulacyjny przystąpiono do przygotowania metody optymalizacji w celu uzyskania sterowania, które powinno zapewnić trafienie w cel z wymaganą dokładnością oraz zużycie w tym celu minimalnej liczby silników korekcyjnych. Jak zaznaczono we wstępie spełnienie drugiego warunku uniemożliwiało bezpośrednie wyznaczenie czasów i kierunków odpaleń silników korekcyjnych, gdyż to wymagałoby znajomości ich liczby. Dodatkowo założeniem było wykorzystanie metod optymalizacyjnych dedykowanych dla funkcji ciągłych. Z tego względu niezbędna była zamiana układu sterowania z dyskretnego na ciągły, który mógłby następnie podlegać optymalizacji. W tym celu dyskretne impulsy odpaleń silników korekcyjnych zamieniono na dwa parametry sterujące U_1 oraz U_2, które mają możliwość generować ciąg w dwóch prostopadłych kierunkach, zgodnie z osiami y_{NB} oraz z_{NB} układu niewirującego, w obie strony, mogą więc przyjmować wartości dodatnie oraz ujemne. Sytuację tę przedstawiono na Rysunku 6.1.



Rysunek 6.1: Parametry sterujące podlegające optymalizacji

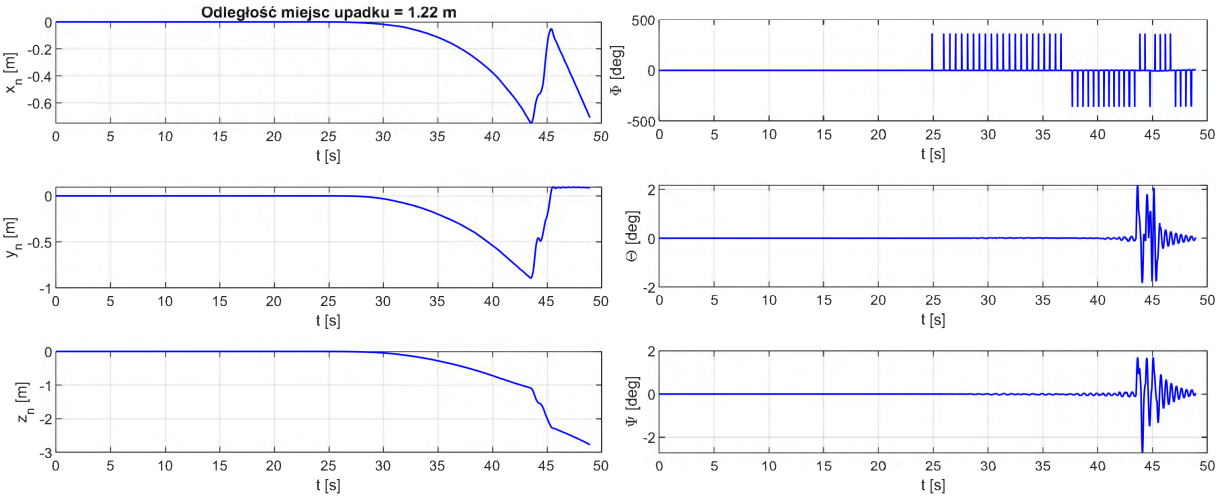
Ze względu na zmianę charakterystyki sterowania zdecydowano się zaniedbać wpływ charakterystyk bezwładnościowych silników korekcyjnych. Założono, że ciąg generowany przez parametry sterujące U_1, U_2 nie powoduje ubytku masy rakiety. Przyjęto również, że oba parametry działają w jednej płaszczyźnie w odległości $\mathbf{r}_{sk}$ od punktu referencyjnego (końca dyszy rakiety). Zdecydowano się także, bez utraty ogólności, wykonywać obliczenia dla atmosfery standardowej bez uwzględniania wpływu wilgotności powietrza. Pominięto również model czujników bezwładnościowych, założono do obliczeń optymalizacyjnych, że algorytmy sterujące dysponują idealnymi danymi dotyczącymi stanu rakiety. Wpływ założenia o nieważkości układu sterowania na zmianę parametrów lotu rakiety pokazuje

Rysunek 6.2. Do obliczeń użyto symulacji z impulsowym układem sterowania, gdzie w trakcie lotu wykorzystano 32 silniki sterujące.



(a) Składowe prędkości liniowej

(b) Składowe prędkości kątowej



(c) Składowe położenia

(d) Kąty orientacji przestrzennej

Rysunek 6.2: Wyniki porównania wpływu nie uwzględniania zmian charakterystyk bezwładnościowych układu sterowania na lot rakiety

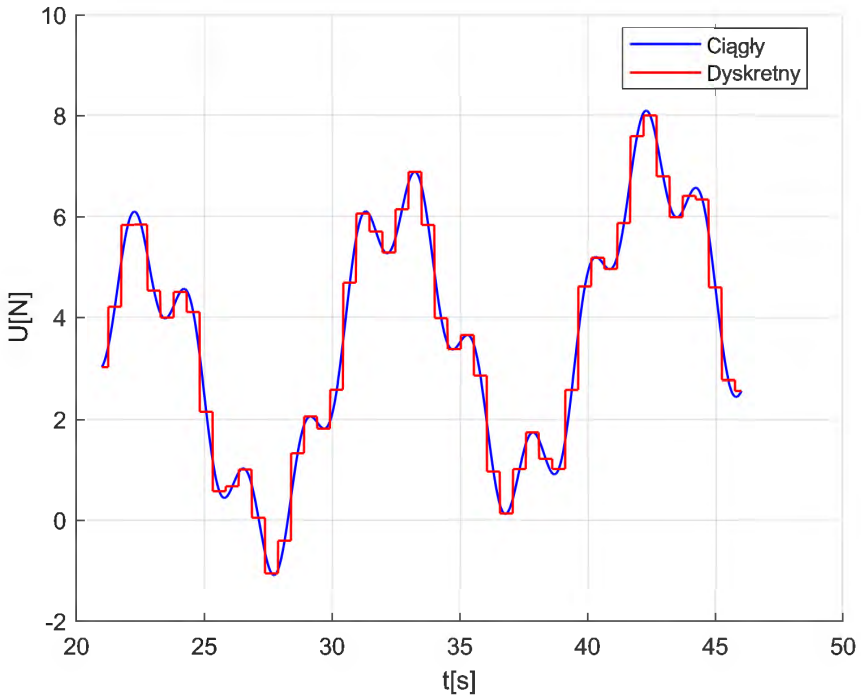
Wykresy na Rysunku 6.2 przedstawiają różnice między wartościami poszczególnych parametrów dla symulacji uwzględniającej i nieuwzględniającej zmianę masy rakiety spowodowaną działaniem układu sterowania. Przedstawione zostały składowe prędkości liniowej i kątowej rakiety, składowe położenia w układzie nawigacyjnym oraz kąty orientacji przestrzennej. Różnica między miejscami upadku rakiety w obu symulacjach wyniosła nieco ponad 1 metr. Na wykresach prędkości liniowych, kątowych oraz kątów orientacji przestrzennej widoczne są niewielkie oscylacje występujące w końcowej fazie lotu, które wynikają z przesunięcia fazowego wartości tych parametrów związanego z wirowaniem rakiety. Nie ma to jednak dużego wpływu na wyniki co pokazuje różnica odległości między trafieniami. Uznano, że skorzystanie z tych założeń upraszczających jest zasadne.

6.1 Określenie funkcji kosztu

Celem optymalizacji jest wyznaczenie wartości parametrów sterujących w każdej chwili czasu w taki sposób, aby uzyskać trafienie w cel oraz zużyć minimalną ilość energii. Funkcję kosztu sformułowano więc w następującej postaci:

$$\min_{\mathbf{u}(t)} J(\mathbf{u}(t)) = \left|\mathbf{X}^{\mathbf{u}(t)} - \mathbf{X}_{cmd}\right|^2 + \int_0^{T_f} \mathbf{u}(t)^2 dt \tag{6.1}$$

gdzie $\mathbf{u}(t)$ jest wektorem parametrów sterujących zależnych od czasu, $\mathbf{X}^{\mathbf{u}(t)} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ opisuje współrzędne miejsca upadku rakiety zależne od sterowania, a współrzędne celu oznaczone zostały jako $\mathbf{X}_{cmd} = \begin{bmatrix} X_{cmd} & Y_{cmd} & Z_{cmd} \end{bmatrix}^T$. Zdecydowano się jako koszt końcowy zadać kwadrat z modułu różnicy tych wektorów zamiast zwykłej normy euklidesowej, ponieważ przy liczeniu gradientu pojawiający się w mianowniku pierwiastek powodowałby niezdefiniowanie problemu przy dążeniu tego członu do zera. Kwadrat normy euklidesowej nie powoduje tych problemów. Koszt ciągły jest to całka z kwadratu ciągu układu sterującego po czasie. Zagadnienie w takiej postaci jest bardzo trudne do obliczenia, gdyż przy locie trwającym około 50 sekund, mając dwa parametry sterujące oraz krok czasowy integracji wynoszący 0.001 sekundy liczba parametrów optymalizacyjnych wyniosłaby 100 tysięcy. Aby zmniejszyć wymiar problemu zdecydowano się po pierwsze sterować tylko w fazie opadania rakiety, co również na starcie miało ograniczyć ilość zużytej energii, a także zastosowano kawałkami stałą dyskretyzację parametrów sterujących, przedstawioną na Rysunku 6.3.



Rysunek 6.3: Kawałkami stała dyskretyzacja parametrów sterujących

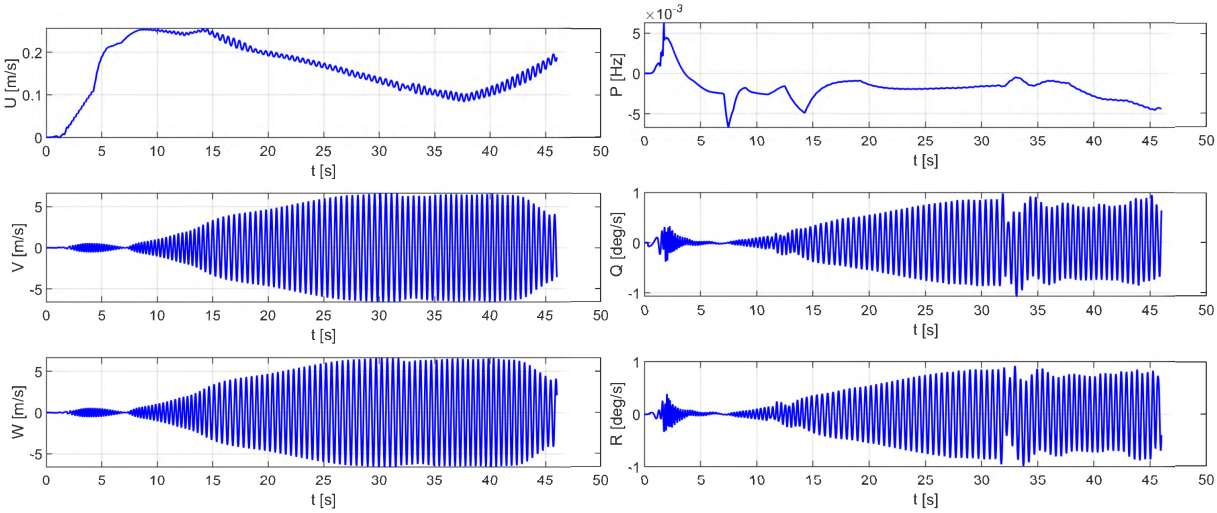
Polega ona na podzieleniu okresu opadania rakiety na pewną ilość przedziałów i założeniu, że w tych przedziałach parametry sterujące mają wartość stałą. Optymalizacji podlegały wtedy tylko wartości tych parametrów w każdym z przedziałów. Po kilku testach zdecydowano się na użycie 50 przedziałów co dało łącznie 100 zmiennych optymalizacyjnych. Dodatkowo rozwiązywane zagadnienie posiada nieustalony czas końcowy. Symulacja kończy się w momencie uderzenia rakiety w ziemię, więc czas końcowy zmienia się z iteracji na iterację. Aby uniknąć problemów mogących wynikać z tego zjawiska, podążając za procedurą opisaną w [144], zamieniono granicę całkowania z przedziału $T \in (T_0; T_f)$, gdzie T_0 jest początkowym czasem symulacji, na przedział jednostkowy $T \in (0; 1)$ dodając czas końcowy do parametrów optymalizacyjnych. Powoduje to, że liczba iteracji w procedurze całkowania za każdym razem jest taka sama, zmienia się natomiast krok całkowania. Po zastosowanych przekształceniach funkcja kosztu przybrała postać:

$$\min_{\mathbf{u}_k, T_f} J(\mathbf{u}_k, T_f) = \left|\mathbf{X}^{\mathbf{u}_k, T_f} - \mathbf{X}_{cmd}\right|^2 + T_f \int_0^1 \mathbf{u}_k^2 dt \tag{6.2}$$

gdzie $\mathbf{u}_k = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_{50}\end{bmatrix}^T$, $\mathbf{u}_i = \begin{bmatrix}u_V = U_1 & u_H = U_2\end{bmatrix}^T$ jest wektorem parametrów sterujących, którego każdy element składa się z parametru w płaszczyznach poziomej i pionowej.

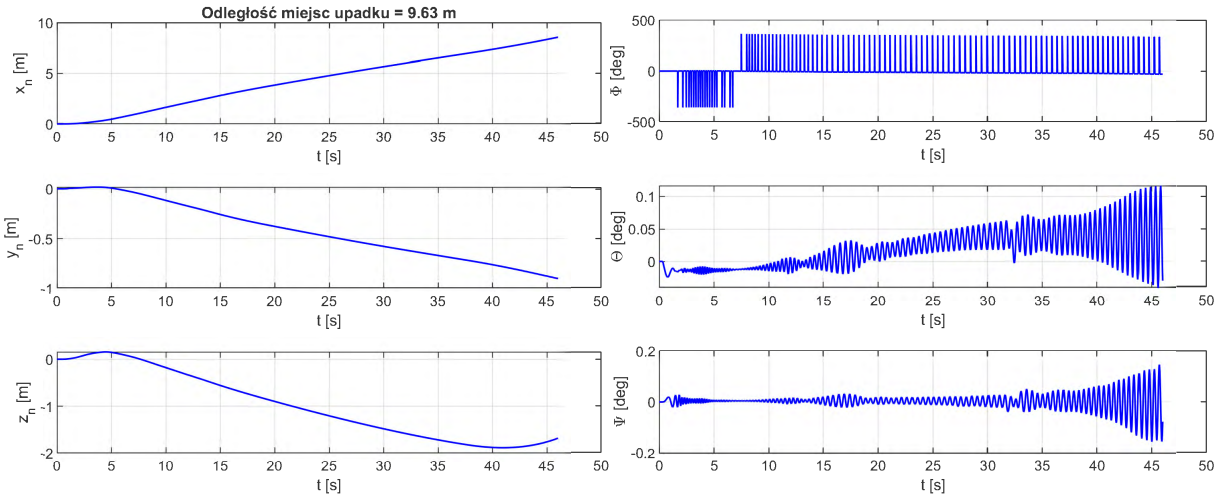
6.2 Przygotowanie środowiska

Obliczenia optymalizacyjne zdecydowano się przeprowadzić z użyciem otwartego oprogramowania IPOPT (Interior Point OPTimizer) [145, 146]. Jest ono w całości napisane w języku C++ i umożliwia rozwiązywanie nieliniowych problemów optymalizacyjnych z więzami o dużej liczbie zmiennych. Metoda obliczeniowa opiera się na algorytmie punktu wewnętrznego (ang. Interior Point), który przekształca zadanie z ograniczeniami w szereg problemów bez ograniczeń, minimalizowanych iteracyjnie. Krok optymalizacji dobierany jest korzystając z metody filtra. Dodatkowo wykorzystano oprogramowanie ADOL-C służące do automatycznego różniczkowania funkcji w języku C++. Korzysta ono z wbudowanych w ten język możliwości przeciążania operatorów i funkcji matematycznych w celu wyznaczania gradientów, jakobianów i hesjanów [147, 148]. Korzystanie z metody automatycznego różniczkowania znacząco zwiększa dokładność obliczeń optymalizacyjnych, gdyż zapewnia analityczną, poza błędem maszynowym, dokładność wyznaczania pochodnych [149, 150]. Aby móc rozpocząć obliczenia wymagane było przeniesienie modelu symulacyjnego rakiety z Simulinka do języka C++. Aby zachować kontrolę nad kodem i upewnić się, że będzie on zwracał takie same wyniki po tłumaczeniu zdecydowano się na jego ręczne przepisanie. Porównanie wyników działania symulacji z Simulinka i kodu C++ przedstawia Rysunek 6.4.



(a) Składowe prędkości liniowej

(b) Składowe prędkości kątowej



(c) Składowe położenia

(d) Kąty orientacji przestrzennej

Rysunek 6.4: Wyniki porównania działania modeli symulacyjnych w środowisku Simulink i języku C++

Porównywano ponownie składowe prędkości liniowych, kątowych, składowe położenia w układzie nawigacyjnym oraz kąty orientacji przestrzennej. Różnica między upadkiem rakiety w obu symulacjach (Simulink i C++) wyniosła niecałe 10 metrów. Na wykresach prędkości liniowych, kątowych oraz kątów orientacji przestrzennej widoczne są niewielkie oscylacje zwiększające amplitudę w miarę lotu rakiety, które wynikają z przesunięcia fazowego wartości tych parametrów związanego z wirowaniem rakiety. Różnica w punkcie upadku wynikać może z innego działania niektórych funkcji, na przykład trygonometrycznych [151], a także procedury całkowania i rozwiązywania równań liniowych, które mogą różnić się między dwoma językami. Stwierdzono, że tak małe różnice nie powinny mieć wpływu na dokładność obliczeń, szczególnie biorąc pod uwagę dokładność samego modelu względem danych telemetrycznych.

6.3 Procedura optymalizacji

Optymalizacja przygotowanej funkcji kosztu realizowana będzie w dwóch wariantach nazwanych tutaj metodami bezpośrednią i pośrednią. Obie metody zostaną porównane pod względem czasu obliczeń, dokładności rozwiązania i stosowalności do opracowanego zagadnienia. W metodzie bezpośredniej problem ten zostanie sformułowany jako zagadnienie programowania nieliniowego i rozwiązany bezpośrednio wykorzystując gradient funkcji celu. W metodzie pośredniej wykorzystana zostanie teoria sterowania optymalnego pod postacią Zasady Maksimum Pontryagina [152] w celu otrzymania wzoru na gradient funkcji celu. Jest to jedno z fundamentalnych narzędzi w teorii sterowania optymalnego, sformułowane w latach 50 ubiegłego wieku jako warunek konieczny optymalności dla problemów z więzami dynamicznymi. Dla układu dynamicznego opisanego równaniami różniczkowymi:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{6.3}$$

gdzie $\mathbf{x}(t)$ jest wektorem stanu zależnym od czasu t, a $\mathbf{u}(t)$ wektorem sterowań, gdzie celem jest minimalizacja kosztu danego jako:

$$J(\mathbf{u}) = \Phi\left(\mathbf{x}(T_f)\right) + L\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \tag{6.4}$$

składającego się z kosztu końcowego $\Phi(\dots)$ oraz kosztu ciągłego $L(\dots)$, gdzie T_f jest czasem końcowym, można wprowadzić funkcję Hamiltona opisaną równaniem:

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = L\left(\mathbf{x}(t), \mathbf{u}(t), t\right) + \boldsymbol{\lambda}^{\top}\mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \tag{6.5}$$

gdzie wektor $\boldsymbol{\lambda}(t)$ oznacza zmienne sprzężone zwane współczynnikami Lagrange'a. Zasada maksimum mówi, że optymalne sterowanie w każdej chwili czasu maksymalizuje funkcję Hamiltona. Warunki konieczne optymalności wynikające z tej zasady można przedstawić jako:

$$\dot{\mathbf{x}}(t) = \frac{\partial H}{\partial \boldsymbol{\lambda}} \tag{6.6}$$

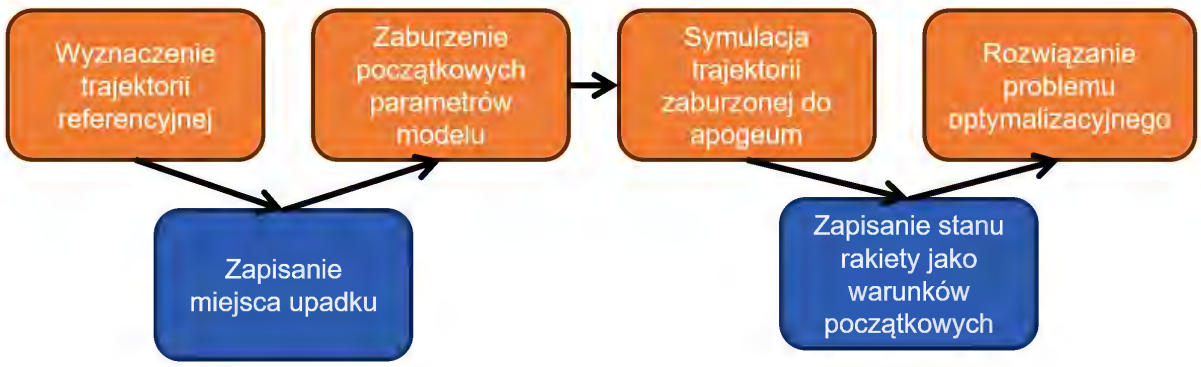
$$\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial H}{\partial \mathbf{x}} \tag{6.7}$$

$$\mathbf{u}^*(t) = \arg\max_{\mathbf{u}\in U} H(\mathbf{x}^*(t), \mathbf{u}, \boldsymbol{\lambda}(t), t) \tag{6.8}$$

gdzie $\mathbf{u}^*(t)$ jest sterowaniem optymalnym, a $\mathbf{x}^*(t)$ odpowiadającą mu trajektorią optymalną.

6.3.1 Metoda bezpośrednia

Oprogramowanie IPOPT wymaga zdefiniowania problemu optymalizacyjnego w postaci klasy języka C++. Wymagane jest sprecyzowanie wymiaru problemu, ilości zmiennych optymalizacyjnych, ilości i rodzaju więzów, a także metod wyznaczających wartość funkcji kosztu oraz jej gradientu i hesjanu. W metodzie bezpośredniej wartość funkcji kosztu wyznaczana jest z równania (6.2), a jej gradient obliczany jest z wykorzystaniem oprogramowania ADOL-C, które zapamiętując operacje matematyczne wykonywane w trakcie wyliczania wartości funkcji od razu zwraca jej gradient. Hesjan został aproksymowany przez środowisko IPOPT. Wartości parametrów sterujących były normalizowane przez maksymalną wartość ciągu do wartości z przedziału $u_i \in (-1; 1)$ i takie więzy zostały na nie nałożone. Czas końcowy również ograniczany był do przedziału $T_f \in (0, 50)$, aby uniknąć nierzeczywistych rozwiązań. Procedura obliczeń w celu otrzymania wyniku optymalizacji przedstawiona jest schematycznie na Rysunku 6.5.



Rysunek 6.5: Schemat procedury optymalizacji

Pierwszym krokiem było wyznaczenie celu trafienia. Realizowano to poprzez symulowanie tak zwanej trajektorii referencyjnej. Ustawiano wyrzutnie na ustalone kąty strzału, 0^o azymutu i 45^o elewacji, oraz używano danych odnoszących się do generycznego modelu rakiety opisanych w rozdziale 5. Rakieta nie była sterowana. W trakcie symulacji wektor stanu rakiety $\mathbf{x}$, składający się z trzech składowych prędkości liniowych, kątowych, położenia w układzie nawigacyjnym oraz kątów orientacji przestrzennej, propagowany był w czasie z użyciem metody całkowania Rungego-Kutty czwartego rzędu, zwanej potocznie metodą RK4, z krokiem całkowania wynoszącym 0,001 sekundy. Równanie, które podlegało całkowaniu ma postać:

$$\dot{\mathbf{x}} = \mathbf{f}\left(t, \mathbf{x}(t), \mathbf{u}_k(t)\right) \tag{6.9}$$

gdzie $\mathbf{f}(\dots)$ jest funkcją prawych stron opisującą model rakiety zależną w ogólności od czasu t, stanu rakiety $\mathbf{x}(t)$ oraz sterowania $\mathbf{u}_k(t)$. Symulacja kończyła się w momencie uderzenia rakiety w ziemie i ten punkt zapisywany był jako cel $\mathbf{X}_{cmd}$. Następnie początkowe kąty strzału, a także parametry modelu rakiety takie jak charakterystyki bezwład-

nościowe, współczynniki aerodynamiczne, ciąg silnika głównego oraz prędkość i kierunek wiatru były zaburzane wykorzystując rozkład normalny. Jako wartość średnią przyjmowano wartość dla danych rakiety generycznej, a odchylenia standardowe dane jako wartości bezwzględne lub procent wartości średniej określano zgodnie z Tabelą 6.1.

Tabela 6.1: Wartości zaburzeń parametrów modelu rakiety

Parametr	Odchylenie standardowe
Kąt azymutu [deg]	1
Kąt elewacji [deg]	$0,66$
Prędkość wiatru [m/s]	3
Kierunek wiatru [deg]	15
Współczynnik $C_{X_{0_{off}}}$ [-]	$0,5\%$
Współczynnik $C_{X_{0_{on}}}$ [-]	1%
Współczynnik $C_{X_{\alpha^2}}$ [-]	$0,5\%$
Współczynnik C_{N_α} [-]	5%
Współczynnik C_{N_Q} [-]	5%
Współczynnik C_{m_α} [-]	5%
Współczynnik C_{m_Q} [-]	5%
Współczynnik C_{l_P} [-]	$0,5\%$
Współczynnik C_{l_δ} [-]	$0,5\%$
Współczynnik $C_{Y_{\gamma\alpha}}$ [-]	$0,5\%$
Współczynnik $C_{l_{\gamma\alpha}}$ [-]	$0,5\%$
Współczynnik $C_{n_{\gamma\alpha}}$ [-]	$0,5\%$
Ciąg maksymalny [N]	1%
Czas działania silnika [s]	1%
Odchylenie ciągu od osi Θ_T [deg]	$0,01$
Kierunek odchylenia ciągu Φ_T [deg]	$0 - 360$ (jednorodny)
Masa startowa m_0 [kg]	$0,2753$
Masa paliwa m_p [kg]	$0,116$
Współrzędna CG x_{cg_0} [m]	$0,00565$
Współrzędna CG y_{cg_0} [m]	2%
Współrzędna CG z_{cg_0} [m]	2%
Współrzędna CG x_{cg_k} [m]	$0,0038$
Współrzędna CG y_{cg_k} [m]	2%
Współrzędna CG z_{cg_k} [m]	2%
Moment główny I_{x_0} [kgm^2]	2%
Moment główny I_{y_0} [kgm^2]	$0,0544$
Moment główny I_{z_0} [kgm^2]	$0,1006$
Moment dewiacyjny I_{xy_0} [kgm^2]	2%
Moment dewiacyjny I_{yz_0} [kgm^2]	2%
Moment dewiacyjny I_{xz_0} [kgm^2]	2%
Moment główny I_{x_k} [kgm^2]	$0,00367$
Moment główny I_{y_k} [kgm^2]	$0,07211$
Moment główny I_{z_k} [kgm^2]	$0,04267$

Parametr	Odchylenie standardowe
Moment dewiacyjny I_{xy_k} [kgm^2]	2%
Moment dewiacyjny I_{yz_k} [kgm^2]	2%
Moment dewiacyjny I_{xz_k} [kgm^2]	2%

Korzystając z nowych parametrów modelu ponownie przeprowadzano symulację lotu rakiety wykorzystując metodę RK4 z krokiem całkowania wynoszącym 0,001 sekundy. Symulację kończono w momencie osiągnięcia przez rakietę apogeum, gdyż jak zaznaczono wcześniej sterowanie miało mieć miejsce tylko w fazie opadania rakiety. Stan rakiety w apogeum $(T_0, \mathbf{x}_0)$ był zapisywany i wykorzystywany jako warunki początkowe dla symulacji w czasie obliczeń optymalizacyjnych. Oszczędzano w ten sposób czas obliczeń nie symulując niepotrzebnie pierwszej połowy lotu, która i tak w każdej iteracji nie podlegałaby żadnym zmianom. Mając warunki początkowe dla symulacji przystępowano do rozwiązania problemu optymalizacyjnego. Początkowe wartości parametrów sterujących u_i miały wartość zerową, czyli optymalizacja rozpoczynała się od lotu niesterowanego. Początkową wartość czasu końcowego T_f ustawiano jako czas lotu dla trajektorii referencyjnej. Jak opisano wcześniej, ponieważ czas końcowy symulacji nie był wartością stałą i również podlegał optymalizacji, należało przekształcić procedurę całkowania. Wprowadzono więc bezwymiarowy czas symulacji $\tau \in (0; 1)$ w miejsce czasu $t \in (T_0; T_f)$ co powodowało zmianę w równaniu (6.9). Pochodną po czasie zastąpiono pochodną po parametrze τ uzyskując:

$$\frac{d\mathbf{x}}{d\tau} = T_f \mathbf{f}\left(\tau, \mathbf{x}(\tau), \mathbf{u}_k(\tau)\right) \tag{6.10}$$

Ze względu na to, że granice całkowania były ustalone, liczba iteracji całkowania w każdej iteracji optymalizacyjnej również była taka sama i miała wartość 3000. Zakładając, że czas lotu opadającego wynosił średnio 25 sekund dawało to krok czasowy w okolicach $0,008$ sekundy. W każdej iteracji wyznaczana była wartość funkcji kosztu, obliczany był jej gradient i wypracowywane były wartości parametrów sterujących oraz czasu końcowego do momentu osiągnięcia zbieżności w obliczeniach lub przekroczenia maksymalnej liczby iteracji optymalizacyjnych. Funkcja kosztu przedstawiona w równaniu (6.2) składa się z dwóch części: kosztu końcowego $\Phi\left(\mathbf{x}(T_f)\right)$ i kosztu ciągłego $L\left(\mathbf{u}, T_f\right)$ danych wzorami:

$$\Phi\left(\mathbf{x}(T_f)\right) = \left|\mathbf{X}^{\mathbf{u}_k, T_f} - \mathbf{X}_{cmd}\right|^2 \tag{6.11}$$

$$L\left(\mathbf{u}_k, T_f\right) = T_f \sum_{k=1}^{N=100} u_k^2 dt \tag{6.12}$$

Oba człony stoją ze sobą niejako w sprzeczności, gdyż minimalizacja odległości od celu wymaga użycia sterowania, a dążenie do minimalizacji energii może powodować większy błąd trafienia. Jedną z metod obejścia możliwego problemu, który może z tego wynikać była wspomniana wcześniej normalizacja parametrów sterujących przez podzielenie ich

przez maksymalną wartość ciągu silnika korekcyjnego. Miało to spowodować, że obie części funkcji kosztu będą miały podobny rząd wielkości, więc żadna z części nie byłaby bardziej preferowana do minimalizacji. Kolejnym sposobem było rozbicie optymalizacji na dwie części. Utworzono w tym celu dwie funkcje kosztu dane wzorami:

$$\min_{\mathbf{u}_k, T_f} J_1(\mathbf{u}_k, T_f) = \left|\mathbf{X}^{\mathbf{u}_k, T_f} - \mathbf{X}_{cmd}\right|^2 \tag{6.13}$$

$$\min_{\mathbf{u}_k, T_f} J_2(\mathbf{u}_k, T_f) = \left|\mathbf{X}^{\mathbf{u}_k, T_f} - \mathbf{X}_{cmd}\right|^2 + T_f \sum_{k=1}^{N=100} u_k^2 dt \tag{6.14}$$

Pierwsza z nich zawiera tylko koszt końcowy, a druga jest analogiczna jak w równaniu (6.2). Miało to na celu w pierwszej części doprowadzić do trafienia w cel z bardzo dużą dokładnością, a następnie w części drugiej, rozpoczynając obliczenia od wartości parametrów sterujących i czasu końcowego będących wynikami z części pierwszej, doprowadzić do zmniejszenia zużycia energii jak najmniej oddalając się od ustalonego punktu trafienia. Stosując to podejście obliczono pięć przypadków testowych w celu sprawdzenia dokładności i poprawności obliczeń, które przedstawiono w Tabelach 6.2-6.3. W kolejnych kolumnach, dla obu części optymalizacji, znajdują się:

- Numer przypadku N
- Liczba iteracji $Iter$
- Liczba wywołań funkcji celu $\#F$
- Końcowa wartość funkcji kosztu $J(\mathbf{u}_k, T_f)$
- Końcowa norma gradientu funkcji celu $|\nabla J|$
- Czas obliczeń $t[s]$

Tabela 6.2: Wyniki optymalizacji bezpośredniej dla funkcji kosztu J_1

N	$Iter$	$\#F$	$J(\mathbf{u}_k, T_f)$	$\lvert\nabla J\rvert$	$t[s]$
1	12	27	$9,451e^{-16}$	$7,5e^{-9}$	32
2	14	27	$7,319e^{-16}$	$3,14e^{-9}$	32
3	13	34	$3,817e^{-18}$	$4,47e^{-10}$	40
4	11	19	$1,288e^{-16}$	$1,48e^{-10}$	23
5	11	24	$5,863e^{-16}$	$2,89e^{-9}$	28
Średnia	12,2	26,2	$4,792e^{-16}$	$2,82e^{-9}$	31

Pierwsza część optymalizacji kończyła się średnio po 12 iteracjach, wywołując funkcję celu około 26 razy. Średnia wartość kosztu na koniec obliczeń była rzędu 10^{-16}, a jako że była to odległość upadku od celu oznacza to bezpośrednie trafienie. Norma gradientu

rzędu 10^{-9} oznacza, że optymalizator znalazł minimum z dobrą dokładnością. Średni czas obliczeń pierwszej części wyniósł 31 sekund. Zapisując rezultaty w postaci parametrów sterujących i czasu końcowego rozpoczynano obliczenia części drugiej podsumowane w Tabeli 6.3.

Tabela 6.3: Wyniki optymalizacji bezpośredniej dla funkcji kosztu J_2

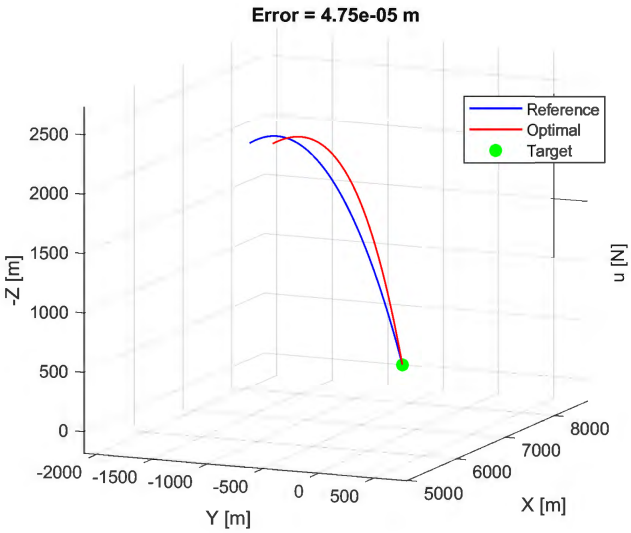
N	$Iter$	$\#F$	$J(\mathbf{u}_k, T_f)$	$\lvert\nabla J\rvert$	$t[s]$
1	300	6042	$1,472e^{-2}$	$2,62e^{-4}$	6615
2	300	5164	$3,789e^{-2}$	$1,64e^{-3}$	5739
3	300	4348	$1,735e^{-2}$	$7,21e^{-3}$	5090
4	300	7393	$6,667e^{-2}$	$5,85e^{-5}$	8129
5	300	4859	$4,861e^{-2}$	$1,77e^{-2}$	5329
Średnia	300	5561,2	$3,7e^{-2}$	$5,4e^{-3}$	6180,4

Aby uniknąć bardzo długich czasów obliczeń nałożono na optymalizator maksymalną dopuszczalną liczbę iteracji wynoszącą 300. W drugiej części optymalizacji żaden przypadek nie spełnił kryteriów zbieżności oprogramowania IPOPT stąd wszystkie skończyły się z powodu osiągnięcia limitu iteracji. Przyczyną tego jest norma gradientu wynosząca średnio $5e^{-3}$ co oznacza, że optymalizator nie znalazł się wystarczająco blisko minimum. Analizując jednak przebieg iteracji zauważono, że w każdym przypadku przez część końcowych iteracji, od około 200 do 300, nie uzyskiwano już zmian w wartości funkcji kosztu, która średnio wynosiła $3e^{-2}$, mimo bardzo małej długości kroku obliczeniowego. Powodowało to bardzo dużą ilość wywołań funkcji kosztu, ponad 5500, oraz dość długi czas obliczeń sięgający ponad 1,5 godziny. Niemniej jednak uzyskany wynik uważa się za satysfakcjonujący. Ze względu na to, że obie części optymalizacyjne traktuje się jako jeden przypadek symulacyjny wyznaczono średnie wartości parametrów końcowych optymalizacji, które przedstawiono w Tabeli 6.4.

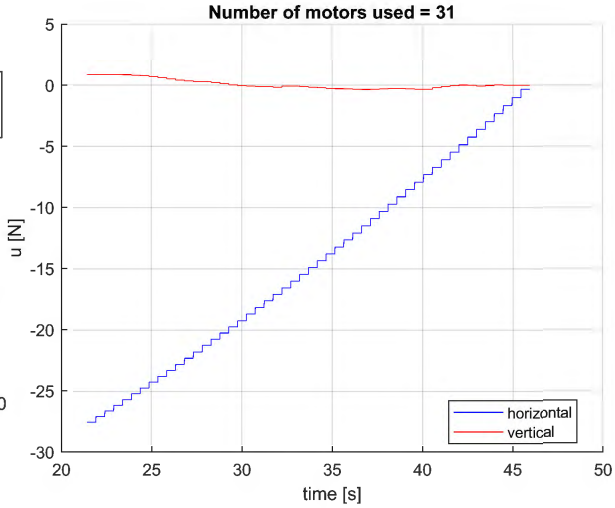
Tabela 6.4: Sumaryczne wyniki optymalizacji bezpośredniej $J_1 + J_2$

$Iter$	$\#F$	$J(\mathbf{u}_k, T_f)$	$\lvert\nabla J\rvert$	$t[min]$
312,2	5587,4	$3,7e^{-2}$	$5,4e^{-3}$	103,5

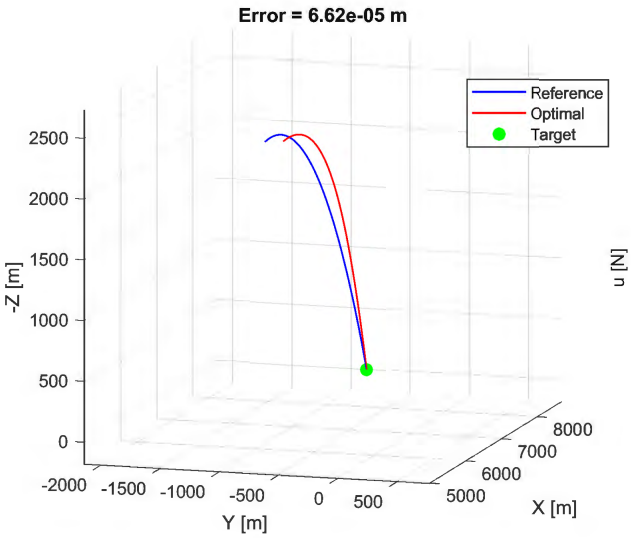
Wyniki optymalizacji bezpośredniej dla przypadków 1 oraz 3 przedstawiono na wykresach na Rysunku 6.6.



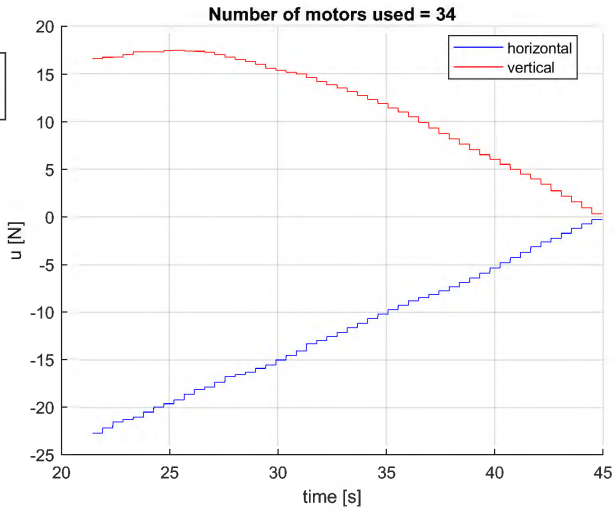
(a) Porównanie trajektorii N = 1



(b) Przebieg parametrów sterujących N = 1



(c) Porównanie trajektorii N = 3



(d) Przebieg parametrów sterujących N = 3

Rysunek 6.6: Wyniki optymalizacji bezpośredniej dla przypadków 1 oraz 3

Po lewej stronie widoczne są trajektorie lotu rakiety oznaczone niebieską linią dla trajektorii referencyjnej oraz linią czerwoną dla trajektorii optymalnej. Zielony punkt oznaczał cel trafienia. Trajektoria optymalna ma gładki przebieg, nie ma widocznych oscylacji, które mogłyby świadczyć o nieefektywnym użyciu sterowania. Po prawej stronie widoczne są przebiegi czasowe parametrów sterujących: poziomego (niebieska linia) oraz pionowego (czerwona linia). Parametry te zostały przemnożone przez maksymalną wartość ciągu pojedynczego silnika korekcyjnego. Widoczne są charakterystyczne schodki wynikające z kawałkami stałej dyskretyzacji. Największy wydatek energii występuje zaraz po apogeum i stopniowo zmniejsza się wraz ze zbliżaniem się rakiety do celu. Znając impuls całkowity pojedynczego silnika korekcyjnego $I_{c_{sk}}$, liczbę wykorzystanych silników

korekcyjnych N_{sk} można wyliczyć z równania:

$$N_{sk} = \frac{F_{T_{sk}}}{I_{c_{sk}}} \int_0^{T_f} |u_V(t)| + |u_H(t)| dt \tag{6.15}$$

Zarówno wydatek energii jak i dokładność trafienia są wartościami satysfakcjonującymi.

6.3.2 Metoda pośrednia

W metodzie pośredniej schemat obliczeń wyglądał analogicznie jak pokazano na Rysunku 6.5 oraz wyjaśniono w poprzednim rozdziale. Jedyną różnicą było użycie gradientu funkcji celu wyprowadzonego korzystając z teorii sterowania optymalnego. Ponownie optymalizację rozbito na dwie części zgodnie z równaniami (6.13), (6.14). W tej metodzie natomiast problem optymalizacyjny rozpatruje się jako zagadnienie nieliniowe z więzami dynamicznymi w postaci:

$$\begin{aligned} \min_{\mathbf{u}(t), T_f} J(\mathbf{u}(t), T_f) &= \Phi\left(\mathbf{x}(T_f)\right) + L\left(\mathbf{u}(t), T_f\right) \\ \dot{\mathbf{x}} &= \mathbf{f}\left(t, \mathbf{x}(t), \mathbf{u}(t)\right), \mathbf{x}(0) = \mathbf{x}_0 \end{aligned} \tag{6.16}$$

Przy wyprowadzaniu równań stanowiących warunki optymalności przedstawionego zagadnienia korzystano z metodologii przedstawionej w [153, 154]. W pierwszej kolejności dla równania dynamicznego opisującego więzy zastosowano zamianę granic całkowania otrzymując równanie (6.10) zapisane w postaci:

$$\frac{d\mathbf{x}}{d\tau} = T_f \mathbf{f}\left(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau)\right) = \hat{\mathbf{f}}\left(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau), T_f\right) \tag{6.17}$$

W dalszej kolejności równanie to zdyskretyzowano korzystając z metody całkowania RK4. Metoda ta jest opisana zestawem równań:

$$\begin{aligned} \mathbf{k}_1 &= d\tau \cdot \hat{\mathbf{f}}\left(\tau_k, \mathbf{x}_k, \mathbf{u}_k, T_f\right) \\ \mathbf{k}_2 &= d\tau \cdot \hat{\mathbf{f}}\left(\tau_k + \frac{1}{2} d\tau, \mathbf{x}_k + \frac{1}{2}\mathbf{k}_1, \mathbf{u}_k, T_f\right) \\ \mathbf{k}_3 &= d\tau \cdot \hat{\mathbf{f}}\left(\tau_k + \frac{1}{2} d\tau, \mathbf{x}_k + \frac{1}{2}\mathbf{k}_2, \mathbf{u}_k, T_f\right) \\ \mathbf{k}_4 &= d\tau \cdot \hat{\mathbf{f}}\left(\tau_k + d\tau, \mathbf{x}_k + \mathbf{k}_3, \mathbf{u}_k, T_f\right) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \frac{1}{6}\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right) \end{aligned} \tag{6.18}$$

gdzie subskrypt k oznacza wartość w danym kroku czasowym. Zestaw równań (6.18) można zapisać w skrócie jako:

$$\mathbf{x}_{k+1} = \hat{\mathbf{F}}\left(\mathbf{x}_k, \mathbf{u}_k, T_f\right) \tag{6.19}$$

pomijając zależność od τ, gdyż funkcja prawych stron w opadającej fazie ruchu rakiety nie zależy jawnie od czasu. Równanie (6.16) można teraz zapisać w dyskretnej postaci jako:

$$\begin{aligned} \min_{\mathbf{u},T_f} J(\mathbf{u},T_f) &= \Phi\left(\mathbf{x}(N)\right) + L\left(\mathbf{u},T_f\right) \\ \mathbf{x}(k+1) &= \hat{\mathbf{F}}\left(\mathbf{x}(k),\mathbf{u}(k),T_f\right), \mathbf{x}(0) = \mathbf{x}_0 \end{aligned} \tag{6.20}$$

gdzie N jest ostatnim punktem czasu. Dla podanej funkcji kosztu z więzami wprowadzić należy Lagrangian dany zależnością:

$$\begin{aligned} \mathcal{L}\left(\mathbf{x},\mathbf{u},T_f,\boldsymbol{\lambda}\right) = \Phi\left(\mathbf{x}(N)\right) + L\left(\mathbf{u},T_f\right) + \boldsymbol{\lambda}(0)\left(\mathbf{x}(0) - \mathbf{x}_0\right) + \\ \sum_{k=0}^{N-1}\left[\boldsymbol{\lambda}(k+1)\left(\mathbf{x}(k+1) - \hat{\mathbf{F}}\left(\mathbf{x}(k),\mathbf{u}(k),T_f\right)\right)\right] \end{aligned} \tag{6.21}$$

gdzie $\boldsymbol{\lambda}$ jest wektorem zmiennych sprzężonych, $k = 0 : N$ kolejnymi krokami czasowymi całkowania. Warunki optymalności pierwszego rzędu mówią, że gradient Lagrangianu powinien wynosić zero. W pierwszej kolejności licząc pochodne po wektorze stanu otrzymujemy równania:

$$\frac{\partial\mathcal{L}\left(\mathbf{x},\mathbf{u},T_f,\boldsymbol{\lambda}\right)}{\partial\mathbf{x}(k)} = \mathbf{0} \quad \Leftrightarrow \quad \boldsymbol{\lambda}(k) = \frac{\partial\hat{\mathbf{F}}\left(\mathbf{x}(k),\mathbf{u}(k),T_f\right)}{\partial\mathbf{x}(k)}^T \boldsymbol{\lambda}(k+1) \tag{6.22}$$

$$\frac{\partial\mathcal{L}\left(\mathbf{x},\mathbf{u},T_f,\boldsymbol{\lambda}\right)}{\partial\mathbf{x}(N)} = \mathbf{0} \quad \Leftrightarrow \quad \boldsymbol{\lambda}(N) = -\frac{\partial\Phi\left(\mathbf{x}(N)\right)}{\partial\mathbf{x}(N)}^T \tag{6.23}$$

opisujące warunek końcowy dla zmiennej sprzężonej oraz warunki jej propagacji wstecznej. Pochodne po zmiennych sprzężonych dają warunki:

$$\frac{\partial\mathcal{L}\left(\mathbf{x},\mathbf{u},T_f,\boldsymbol{\lambda}\right)}{\partial\boldsymbol{\lambda}(k+1)} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{x}(k+1) = \hat{\mathbf{F}}\left(\mathbf{x}(k),\mathbf{u}(k),T_f\right) \tag{6.24}$$

$$\frac{\partial\mathcal{L}\left(\mathbf{x},\mathbf{u},T_f,\boldsymbol{\lambda}\right)}{\partial\boldsymbol{\lambda}(0)} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{6.25}$$

które opisują propagację w czasie wektora stanu z zadanym warunkiem początkowym. Kolejnym krokiem było wyznaczenie wartości gradientów po wektorze sterowań i czasie końcowym, które dane są zależnościami:

$$\frac{\partial\mathcal{L}\left(\mathbf{x},\mathbf{u},T_f,\boldsymbol{\lambda}\right)}{\partial\mathbf{u}} = \frac{\partial L\left(\mathbf{u},T_f\right)}{\partial\mathbf{u}} - \sum_{k=0}^{N-1}\frac{\partial\hat{\mathbf{F}}\left(\mathbf{x}(k),\mathbf{u}(k),T_f\right)}{\partial\mathbf{u}(k)}^T \boldsymbol{\lambda}(k+1) \tag{6.26}$$

$$\frac{\partial\mathcal{L}\left(\mathbf{x},\mathbf{u},T_f,\boldsymbol{\lambda}\right)}{\partial T_f} = \frac{\partial L\left(\mathbf{u},T_f\right)}{\partial T_f} - \sum_{k=0}^{N-1}\frac{\partial\hat{\mathbf{F}}\left(\mathbf{x}(k),\mathbf{u}(k),T_f\right)}{\partial T_f}^T \boldsymbol{\lambda}(k+1) \tag{6.27}$$

Z równań (6.26) oraz (6.27) można otrzymać gradient funkcji celu (6.16). W tym celu należy wziąć pod uwagę dyskretyzację w czasie parametrów sterowania. W powyższych wyprowadzeniach funkcja $\hat{\mathbf{F}}\left(\mathbf{x}(k), \mathbf{u}(k), T_f\right)$ zależała w danej chwili czasu k tylko od dwóch parametrów sterujących $\mathbf{u}(k) = \begin{bmatrix} u_V & u_H \end{bmatrix}^T$. Przedział całkowania $k = 0 : N$ podzielony został również zgodnie z dyskretyzacją sterowania na przedział $i = 0 : M - 1$, gdzie M jest liczbą parametrów sterujących, w tym przypadku $M = 50$. Każda para parametrów sterujących $\mathbf{u}_i$ działała na funkcję stanu rakiety przez $P = N/M$ kroków czasowych k. Gradient funkcji celu po parametrach sterujących można wtedy wyrazić zależnością:

$$\frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial \mathbf{u}_i} = \sum_{k=Pi}^{P(i+1)} \left(2T_f \mathbf{u}_i(k) d\tau - \frac{\partial \hat{\mathbf{F}}\left(\mathbf{x}(k), \mathbf{u}_i(k), T_f\right)}{\partial \mathbf{u}_i(k)}^T \boldsymbol{\lambda}(k+1) \right) \tag{6.28}$$

Gradient funkcji celu po czasie końcowym wyraża zależność:

$$\frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial T_f} = \sum_{k=0}^{N-1} \left(\mathbf{u}^2(k) d\tau - \frac{\partial \hat{\mathbf{F}}\left(\mathbf{x}(k), \mathbf{u}(k), T_f\right)}{\partial T_f}^T \boldsymbol{\lambda}(k+1) \right) \tag{6.29}$$

Łącząc wyznaczone wartości w jeden wektor otrzymano:

$$\nabla J = \begin{bmatrix} \frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial \mathbf{u}_1} & \frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial \mathbf{u}_2} & \cdots & \frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial \mathbf{u}_{50}} & \frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial T_f} \end{bmatrix}^T \tag{6.30}$$

gdzie $\frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial \mathbf{u}_i} = \begin{bmatrix} \frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial u_{V_i}} & \frac{\partial J\left(\mathbf{u}, T_f\right)}{\partial u_{H_i}} \end{bmatrix}^T$. Zdefiniowany w ten sposób gradient funkcji celu przekazywany był do odpowiedniej metody oprogramowania IPOPT. Wszystkie jakobiany funkcji $\hat{\mathbf{F}}$ obliczane były metodą automatycznego różniczkowania z wykorzystaniem ADOL-C. Gradient kosztu końcowego ze względu na prostą zależność matematyczną można zapisać w postaci analitycznej. Jest to wektor 12-elementowy, którego składowe związane z położeniem mają postać:

$$\frac{\partial \Phi\left(\mathbf{x}^{[X,Y,Z]}(N)\right)}{\partial \mathbf{x}(N)} = 2\left(\mathbf{x}^{[X,Y,Z]}(N) - \mathbf{x}_{cmd}^{[X,Y,Z]}\right) \tag{6.31}$$

Pozostałe składowe wektora końcowego zmiennych sprzężonych wynoszą zero. Przy metodzie pośredniej również obliczenia rozbito na dwie część w taki sam sposób jak przy metodzie bezpośredniej, nieuwzględniając w części pierwszej członów funkcji celu i jej gradientu odpowiadających za koszt ciągły L. Ponownie obliczono pięć przypadków testowych z identycznymi jak przy metodzie bezpośredniej warunkami początkowymi, ich wyniki przedstawiono w Tabelach 6.5-6.6.

Tabela 6.5: Wyniki optymalizacji pośredniej dla funkcji kosztu J_1

N	$Iter$	$\#F$	$J(\mathbf{u}_k, T_f)$	$\lvert\nabla J\rvert$	$t[s]$
1	12	27	$7,582e^{-16}$	$8,04e^{-9}$	8,1
2	15	28	$3,505e^{-17}$	$8,62e^{-10}$	8,7
3	13	34	$3,909e^{-18}$	$5,09e^{-10}$	9,0
4	11	19	$2,515e^{-15}$	$2,21e^{-9}$	5,4
5	11	24	$3,690e^{-16}$	$1,20e^{-9}$	7,4
Średnia	12,4	26,4	$7,362e^{-16}$	$2,56e^{-9}$	7,72

W porównaniu do metody bezpośredniej średnia liczba iteracji wynosząca 12, średnia liczba wywołań funkcji kosztu równa 26, także wartości końcowe funkcji kosztu i moduł jej gradientu mające rząd wielkości odpowiednio 10^{-16} oraz 10^{-9} są niemal identyczne. Natomiast średni czas obliczeń pierwszej części optymalizacji wynoszący 7,7 sekundy jest cztery razy niższy. Wynika to zapewne z faktu, że przy liczeniu jakobianów z użyciem automatycznego różniczkowania dla metody pośredniej jest łącznie $12+2+1 = 15$ zmiennych niezależnych (zmienne stanu, dwa sterowania i czas końcowy), a dla metody pośredniej dla gradientu jest ich 101. Oprogramowania ADOL-C jest dość kosztowne obliczeniowo im więcej zmiennych niezależnych musi przechowywać w pamięci. Zaletą metody pośredniej jest więc także to, że dyskretyzacja sterowania mogłoby być dużo gęstsza, wynosząca 1000 czy 10000 zmiennych, bez znacznego wpływu na szybkość obliczeń, gdzie przy metodzie bezpośredniej wpływ ten byłby znaczący. Zapisując rezultaty w postaci parametrów sterujących i czasu końcowego przechodzono do części drugiej optymalizacji. Wyniki podsumowane są w Tabeli 6.3.

Tabela 6.6: Wyniki optymalizacji pośredniej dla funkcji kosztu J_2

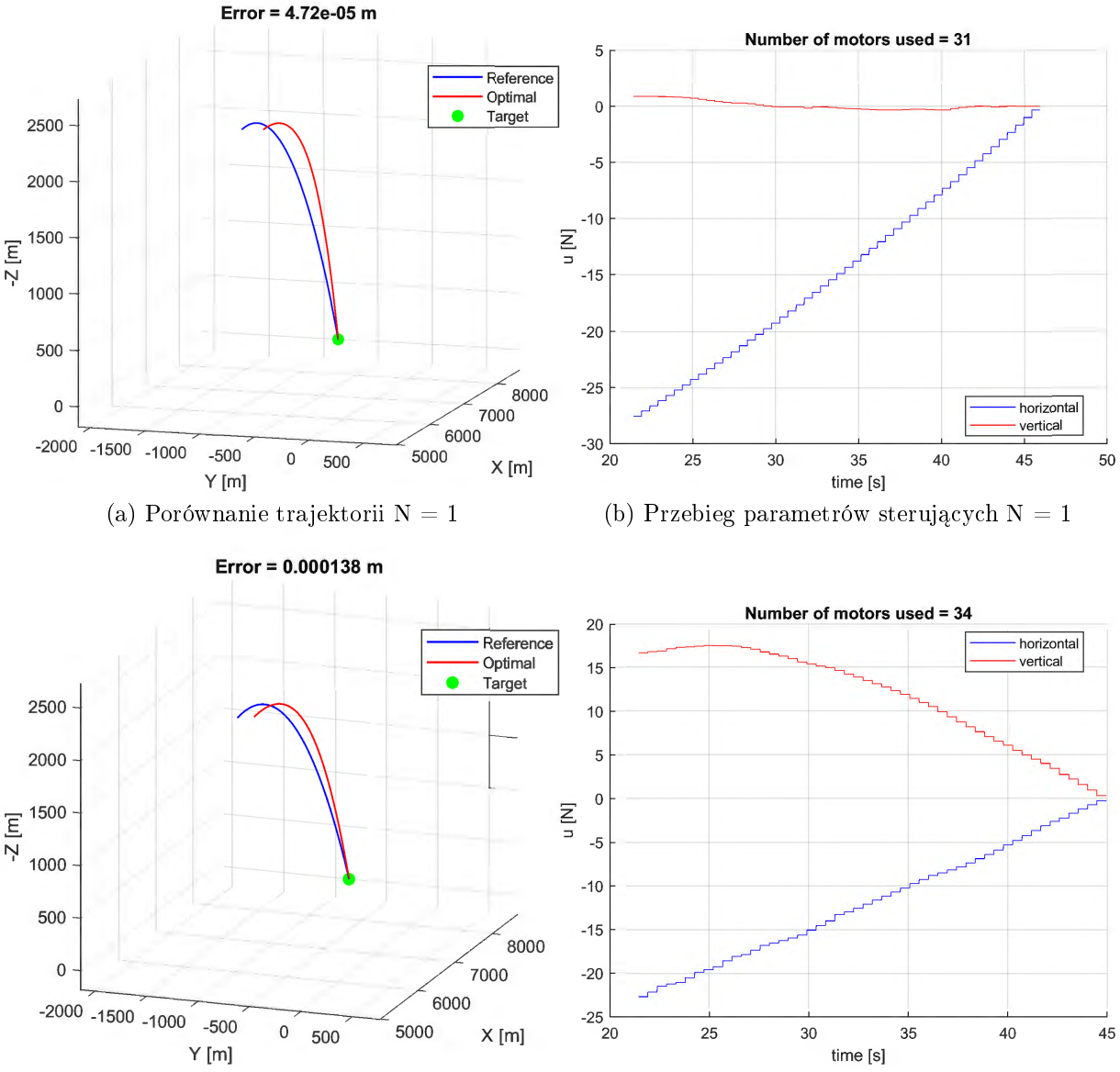
N	$Iter$	$\#F$	$J(\mathbf{u}_k, T_f)$	$\lvert\nabla J\rvert$	$t[s]$
1	177	3251	$1,472e^{-2}$	$5,99e^{-6}$	860
2	267	4146	$3,789e^{-2}$	$3,20e^{-6}$	1035
3	300	4660	$1,753e^{-2}$	$6,42e^{-2}$	1084
4	300	8371	$6,667e^{-2}$	$1,66e^{-5}$	2093
5	300	4971	$4,861e^{-2}$	$4,91e^{-3}$	1171
Średnia	268,8	5079,8	$3,71e^{-2}$	$1,38e^{-2}$	1248,6

Ponownie limit iteracji ustawiono na 300, natomiast dla metody pośredniej część przypadków znalazła rozwiązanie wcześniej. Wartość końcowa funkcji kosztu oraz norma jej gradientu ponownie przyjmują taki sam rząd wielkości jak dla metody bezpośredniej. Liczba wywołań funkcji kosztu jest nieznacznie mniejsza wynosząc około 5000, natomiast ponownie uzyskano około sześciokrotny zysk jeśli chodzi o czas obliczeń wynoszący średnio 20 minut. Traktując ponownie obie części optymalizacyjne jako jeden przypadek symulacyjny wyznaczono średnie wartości parametrów końcowych optymalizacji, które przedstawiono w Tabeli 6.4.

Tabela 6.7: Sumaryczne wyniki optymalizacji pośredniej $J_1 + J_2$

$Iter$	$\#F$	$J(\mathbf{u}_k, T_f)$	$\lvert\nabla J\rvert$	$t[min]$
$281,2$	$5106,2$	$3,71e^{-2}$	$1,38e^{-2}$	$20,9$

Wyniki optymalizacji pośredniej dla przypadków 1 oraz 3 przedstawiono na wykresach na Rysunku 6.7.



(a) Porównanie trajektorii N = 1

(b) Przebieg parametrów sterujących N = 1

(c) Porównanie trajektorii N = 3

(d) Przebieg parametrów sterujących N = 3

Rysunek 6.7: Wyniki optymalizacji pośredniej dla przypadków 1 oraz 3

Wyniki te są niemal identyczne jak dla przypadku użycia metody bezpośredniej. Przebieg parametrów sterujących oraz liczba użytych silników korekcyjnych jest taka sama, podobnie odległość od celu dla przypadku 1. Dla przypadku 3 odległość ta jest rząd wielkości większa, natomiast przy tak dokładnych trafieniach taka różnica nie ma znaczenia.

Analizując wyniki wyraźnie widać, że pod względem dokładności obliczeń obie metody są równie dobre, natomiast metoda pośrednia jest wyraźnie szybsza. Z tego względu do dalszych obliczeń wykorzystywana była tylko metoda pośrednia.

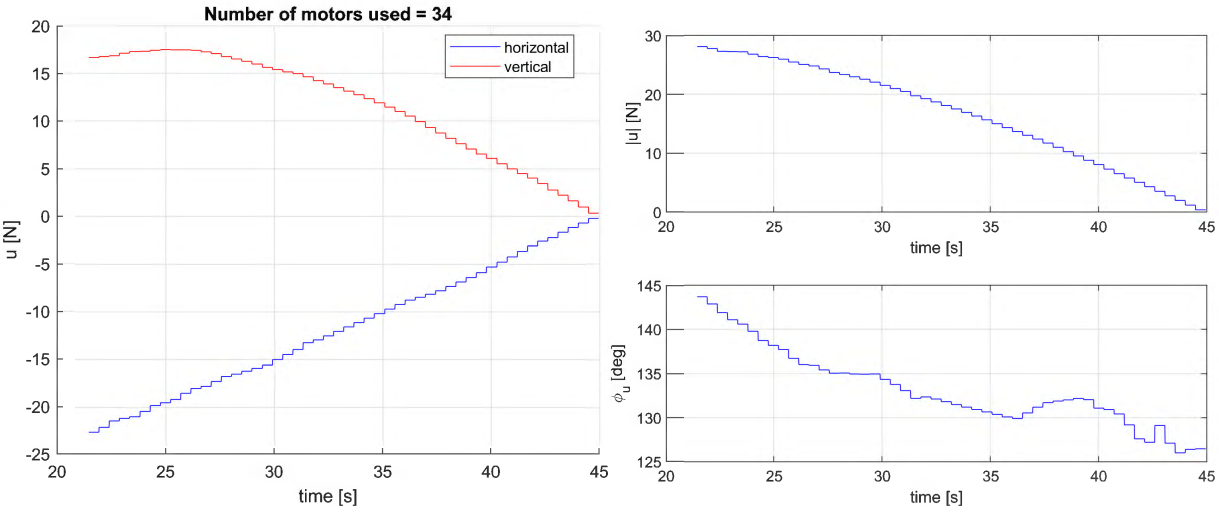
7 Dyskretyzacja

Rezultatem obliczeń optymalizacyjnych były przebiegi czasowe dwóch parametrów sterujących, reprezentujące niezbędny ciąg w dwóch prostopadłych płaszczyznach rakiety, umożliwiające trafienie w cel z dużą dokładnością. Kolejnym krokiem była transformacja uzyskanego sygnału sterującego ciągłego na dyskretne impulsy wynikające z działania silników korekcyjnych. W pierwszej kolejności dokonano transformacji parametrów sterujących z układu kartezjańskiego na biegunowy. Stosując równania (7.1) oraz (7.2) otrzymano moduł generowanego ciągu oraz kierunek jego działania.

$$|u| = F_{T_{sk}} \sqrt{u_H^2 + u_V^2} \tag{7.1}$$

$$\phi_u = \text{atan2}\left(u_V, u_H\right) \tag{7.2}$$

Wynik transformacji układu pokazano na Rysunku 7.1.



(a) Układ kartezjański

(b) Układ biegunowy

Rysunek 7.1: Transformacja parametrów sterujących z układu kartezjańskiego (a) do biegunowego (b)

Następnie wymagane było, aby zamienić wartości modułu i kierunku działania wypadkowej siły ciągu na czasy i kierunki odpalenia kolejnych silników korekcyjnych.

7.1 Wyznaczenie czasu odpalenia

Aby wyznaczyć czasy odpalenia kolejnych silników korekcyjnych wykorzystano znajomość impulsu całkowitego pojedynczego silnika korekcyjnego $I_{c_{sk}}$. Mając przebieg ciągu w czasie, impuls całkowity można wyznaczyć ze wzoru:

$$I_c = \int_{t_0}^{t_f} |u| dt \tag{7.3}$$

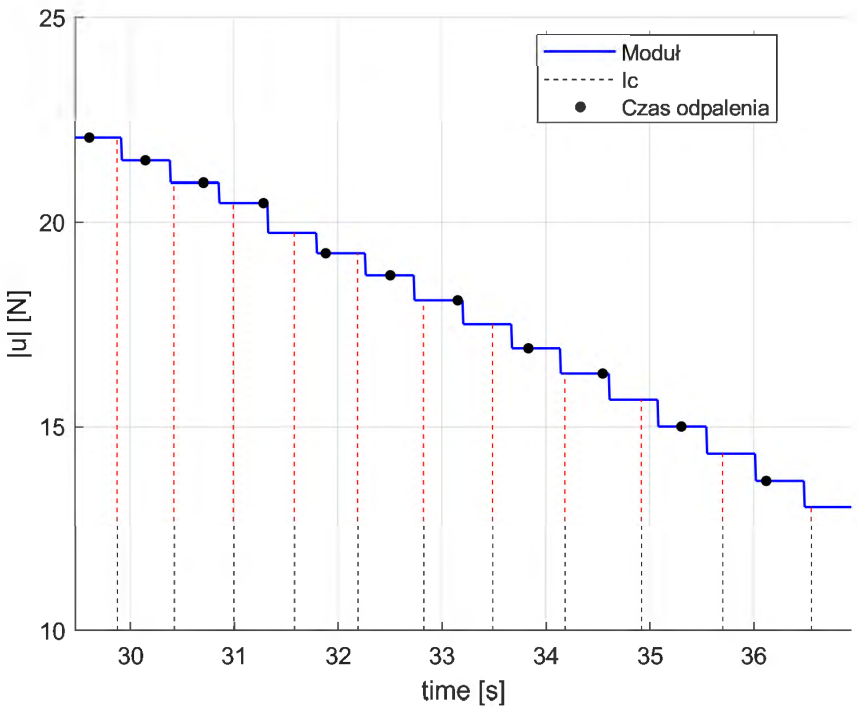
gdzie t_0 i t_f oznaczają czas początkowy i końcowy całkowania. Przedział całkowania podzielono na N podprzedziałów $(t_0; t_1), (t_1; t_2), \ldots, (t_{N-1}; t_f)$, gdzie N jest liczbą silników korekcyjnych, w taki sposób, że wartość impulsu całkowitego wyznaczona w każdym z nich odpowiada wartości impulsu całkowitego pojedynczego silnika korekcyjnego $I_{c_{sk}}$ zgodnie z:

$$I_{c_{sk}} = \int_{t_0}^{t_1} |u|dt = \int_{t_1}^{t_2} |u|dt = \cdots = \int_{t_{N-1}}^{t_f} |u|dt \tag{7.4}$$

W ten sposób w każdym z tych przedziałów znajduje się taka sama ilość energii odpowiadająca energii zgromadzonej w pojedynczym silniku korekcyjnym. Przyjęto, że czas odpalenia każdego silnika korekcyjnego znajduje się w punkcie przedziału, w którym osiągane jest 50% energii. Wyznaczyć go można dzieląc moment impulsu całkowitego przez wartość tego impulsu zgodnie z równaniem:

$$t_{sk_i} = \frac{\int_{t_i}^{t_{i+1}} t|u|dt}{\int_{t_i}^{t_{i+1}} |u|dt} = \frac{1}{I_{c_{sk}}} \int_{t_i}^{t_{i+1}} t|u|dt \tag{7.5}$$

Wynik procedury wyznaczania czasu odpaleń kolejnych silników korekcyjnych pokazano poglądowo na Rysunku 7.2.



Rysunek 7.2: Wykres przedstawiający wyznaczanie czasów odpaleń kolejnych silników korekcyjnych

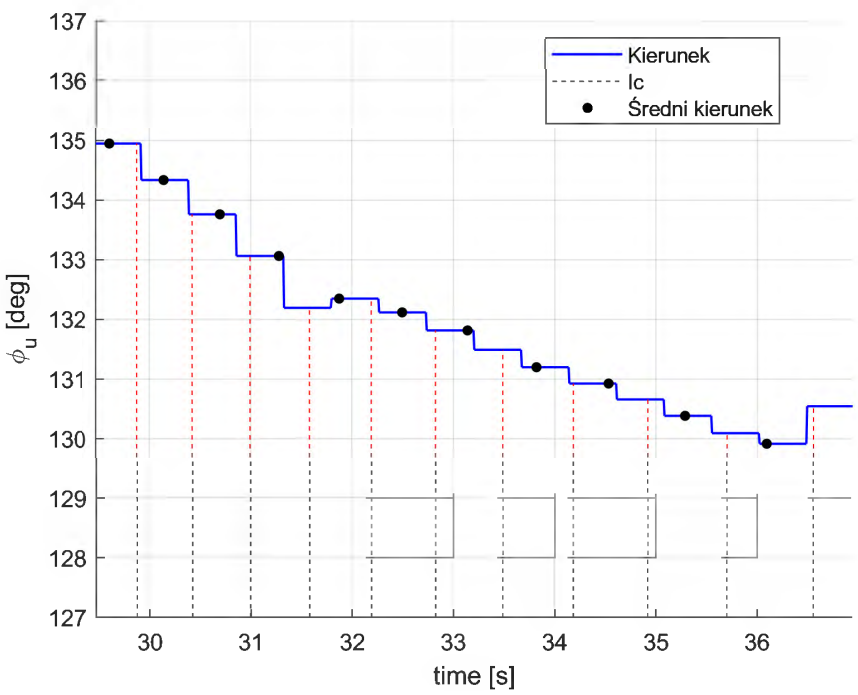
Na wykresie z Rysunku 7.2 linią niebieską zaznaczono wartość modułu ciągu w czasie, linia czerwona przerywana oddziela przedziały, w których znajduje się taka sama ilość energii równa impulsowi całkowitemu pojedynczego silnika korekcyjnego, a czarne punkty oznaczają wyznaczone z równania (7.5) czasy odpaleń kolejnych silników korekcyjnych.

7.2 Wyznaczenie kierunku odpalenia

Wyznaczenie kierunków odpaleń silników korekcyjnych odbywało się w podobny sposób. Przebieg czasowy wartości kąta ϕ_u również podzielono na takie same przedziały $(t_0; t_1), (t_1; t_2), \ldots, (t_{N-1}; t_f)$. W tym przypadku zdecydowano, że kierunek, w którym powinien zadziałać silnik równy będzie średniej wartości kąta ϕ_u w każdym z podprzedziałów zgodnie z równaniem:

$$\phi_{sk_i} = \frac{1}{t_{i+1} - t_i} \int_{t_i}^{t_{i+1}} \phi_u dt \tag{7.6}$$

Wynik tej operacji przedstawiono poglądowo na Rysunku 7.3.



Rysunek 7.3: Wykres przedstawiający wyznaczanie kierunków odpaleń kolejnych silników korekcyjnych

Na wykresie z Rysunku 7.3 linią niebieską zaznaczono wartość kierunku działania ciągu w czasie, linia czerwona przerywana oddziela przedziały, w których znajduje się taka sama ilość energii równa impulsowi całkowitemu pojedynczego silnika korekcyjnego, a czarne punkty oznaczają wyznaczone z równania (7.6) kierunki, w których powinny zostać odpalone kolejne silniki korekcyjne.

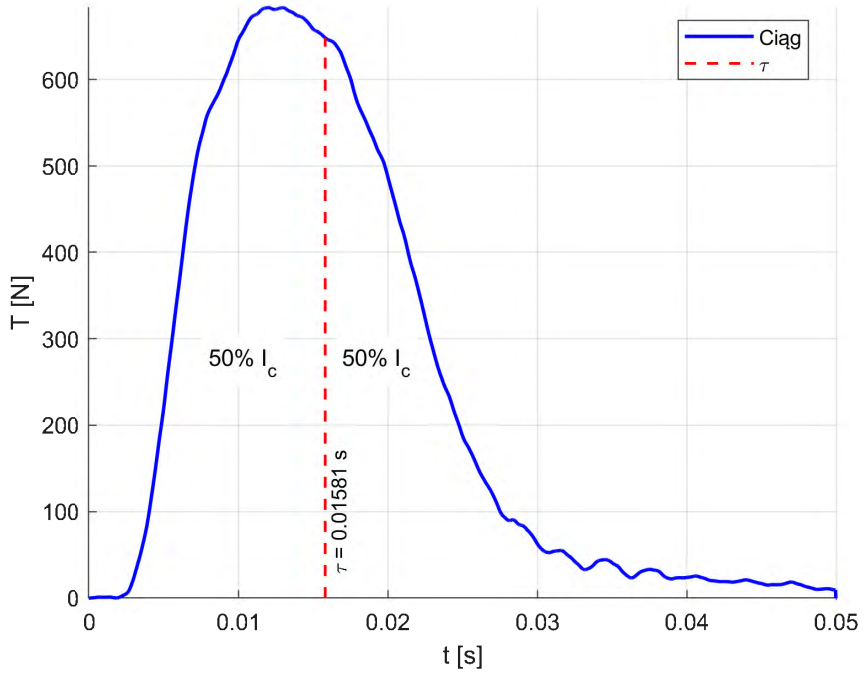
7.3 Algorytm sterowania silnikami

Mając dane czasy i kierunki odpaleń silników korekcyjnych opracowano prosty algorytm sterujący, który pozwoli na sprawdzenie jak zmieniła się dokładność trafienia w cel po zastosowaniu algorytmu dyskretyzacji. W tym momencie założono, że silniki korekcyjne zostaną odpalone zawsze w zadanym czasie i kierunku, nie uwzględniano niepewności

wynikających z działania spłonek silników, czy danych nawigacyjnych dostarczanych do algorytmu sterującego. Czas odpalenia silnika korekcyjnego brano jako moment, w którym znajduje się 50% energii w danym przedziale całkowania. Na etapie wczytywania danych dla algorytmu sterującego do czasów odpalenia dodawana była korekta, w taki sposób aby punkt, w którym znajduje się 50% energii silnika korekcyjnego pokrywał się z wyznaczonym wcześniej czasem. Korekta czasu odpalenia τ_{sk} wyznaczana była z równania:

$$\tau_{sk} = \frac{\int_0^{t_{f_{sk}}} t|F_{T_{sk}}|dt}{\int_0^{t_{f_{sk}}} |F_{T_{sk}}|dt} \tag{7.7}$$

gdzie $t_{f_{sk}}$ jest czasem palenia silnika korekcyjnego. Widoczne jest to na Rysunku 7.4. Niebieską krzywą zaznaczono przebieg ciągu silnika korekcyjnego, a czerwona linia przerywana dzieli wykres w taki sposób, że po obu jej stronach znajduje się 50% energii silnika.



Rysunek 7.4: Korekta czasu odpalenia silnika korekcyjnego

Czas odpalenia silnika korekcyjnego korygowany był zgodnie z równaniem:

$$t_{sk_i} = t_{sk_i} - \tau_{sk} \tag{7.8}$$

W trakcie trwania symulacji algorytm sprawdzał czy minął czas odpalenia danego silnika korekcyjnego:

$$t >= t_{sk_i} \tag{7.9}$$

a następnie dodawana była korekta kierunku działania silnika. Niezbędne było aby średnia siła generowana przez silnik korekcyjny miała kierunek zgodny z wyliczonym kątem

ϕ_{sk_i}. Do tego kąta dodawana więc była poprawka związana z lokalną prędkością kątową przechylania P:

$$\phi_{sk_i} = \phi_{sk_i} - P\tau_{sk} \tag{7.10}$$

Procedurę wyznaczenia czasu i kierunku odpalenia silnika w symulacji przedstawia Algorytm 7.1.

Algorytm 7.1 Dyskretny algorytm sterowania

```
for all Liczba silników korekcyjnych i do
  if t >= t_{sk_i} then
    if Silnik i nie był jeszcze odpalony then
      Zaznacz silnik i jako odpalony
      φ_{sk_i} = φ_{sk_i} − Pτ_{sk}
      Zapisz orientację rakiety w momencie odpalenia silnika φ_{start_i}
      Zapisz czas odpalenia silnika t_{start_i}
    end if
  end if
end for
```

Wewnątrz symulacji fizyczny model silnika korekcyjnego uwzględniany był z godnie z Algorytmem 7.2.

Algorytm 7.2 Fizyczny model silnika korekcyjnego

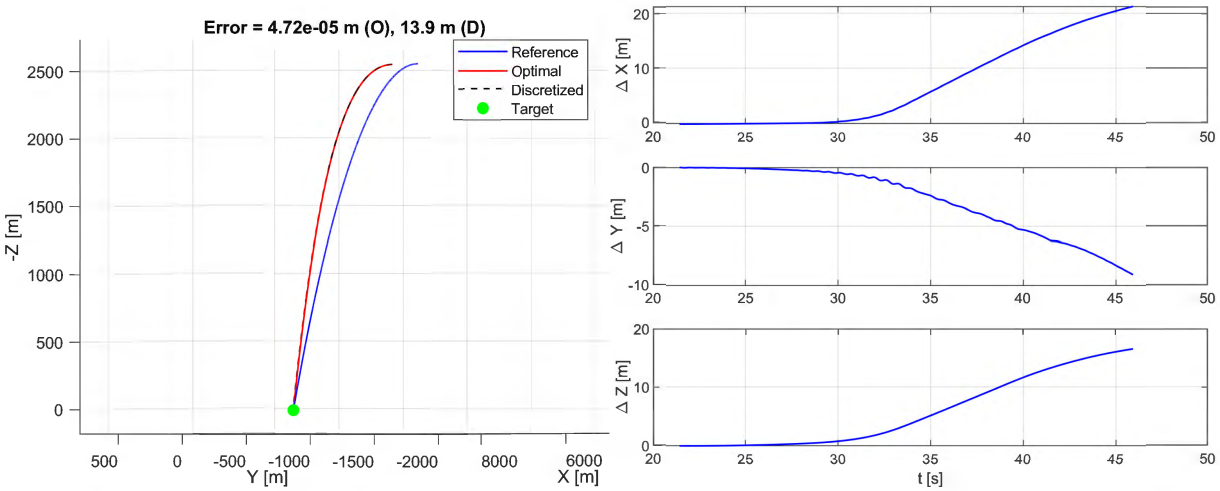
```
for all Liczba silników korekcyjnych i do
  if Silnik i został odpalony then
    Wyznacz czas palenia t_p = t − t_{start_i}
    Znajdź ciąg silnika korzystając z interpolacji liniowej T = f(t_p)
    Wyznacz składowe ciągu wzdłuż osi y_{NB} → F_H i z_{NB} → F_V
    F_H = T · cos(φ_{sk_i} − (φ_{start_i} − φ))
    F_V = T · sin(φ_{sk_i} − (φ_{start_i} − φ))
  end if
end for
```

Przeprowadzono 10 symulacji testowych w celu sprawdzenia jak zmienił się punkt upadku rakiety po transformacji optymalnego sygnału sterującego z ciągłego na dyskretny. Wyniki pokazujące błąd trafienia dla każdego przypadku, używając sterowania ciągłego i dyskretnego przedstawia Tabela 7.1.

Tabela 7.1: Wyniki działania algorytmu dyskretyzacji

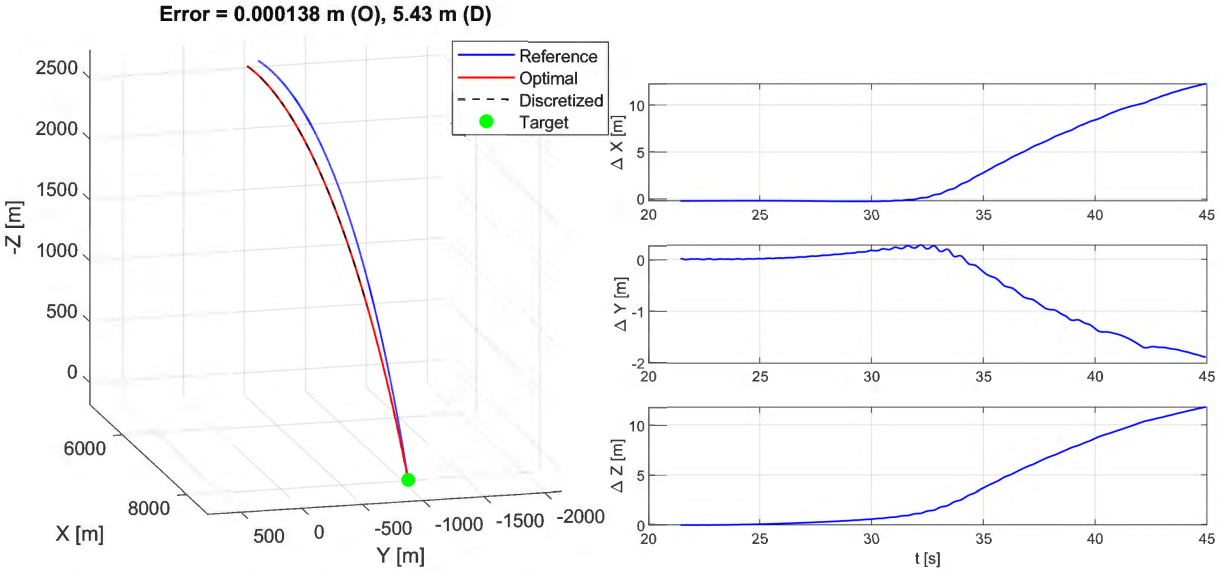
Przypadek	Błąd ciągły [m]	Błąd dyskretny [m]
1	$4,72e^{-5}$	$13,9$
2	$7,42e^{-5}$	$6,59$
3	$1,38e^{-4}$	$5,43$
4	$1,04e^{-4}$	$9,08$
5	$8,65e^{-5}$	$7,83$

Przypadek	Błąd ciągły [m]	Błąd dyskretny [m]
6	$9,75e^{-5}$	3,31
7	$3,40e^{-5}$	7,23
8	$9,09e^{-5}$	4,18
9	$2,61e^{-5}$	2,00
10	$9,21e^{-5}$	9,30
Średnia	$7,91e^{-5}$	6,88



(a) Trajektorie dla przypadku 1

(b) Błąd trajektorii dla przypadku 1



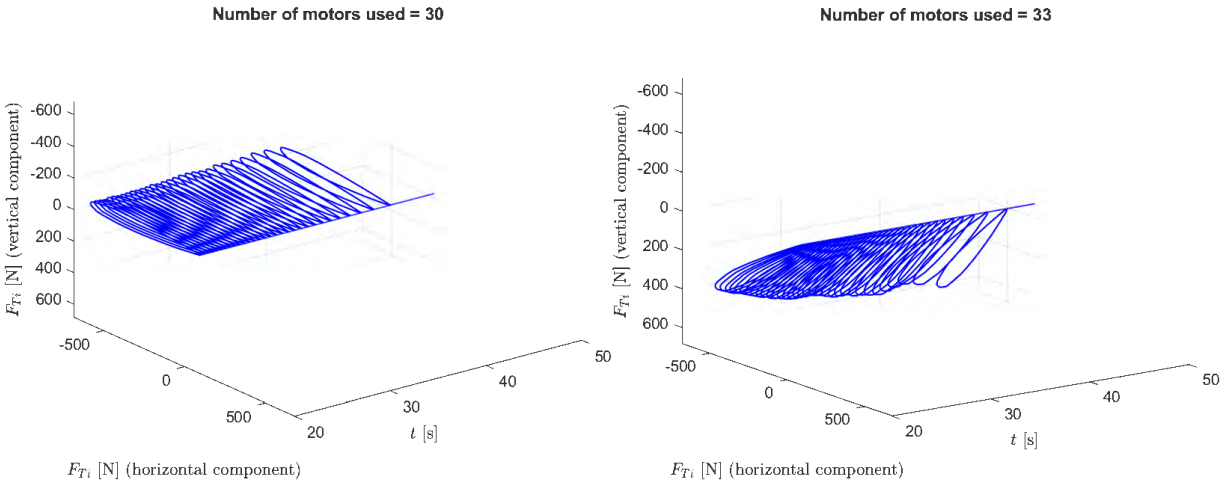
(c) Trajektorie dla przypadku 3

(d) Błąd trajektorii dla przypadku 3

Rysunek 7.5: Trajektorie referencyjna, optymalna i dyskretna oraz błąd wynikający z dyskretyzacji

Po procedurze dyskretyzacji błąd trafienia znacząco wzrósł, natomiast w większości przypadków testowych znajduje się poniżej wartości 10 metrów, co jest wynikiem bardzo dobrym. Dodatkowo na Rysunku 7.5 pokazano, dla przypadków 1 i 3, porównanie trajektorii wynikającej z optymalizacji i tej przy działaniu dyskretnego sterowania, a także

jak zmieniał się błąd między nimi w czasie. Na wykresach z lewej strony kolorem niebieskim oznaczono trajektorię referencyjną, kolorem czerwonym trajektorię będącą wynikiem działania sterowania ciągłego, a kolorem czarnym tę wynikającą z działania sterowania dyskretnego. Zielony punkt oznaczał cel. Na wykresach po prawej stronie widoczny jest błąd pomiędzy trajektoriami czerwoną i czarną dla każdej współrzędnej z osobna. Można zauważyć, że błąd ten przez około połowę okresu sterowania jest bardzo mały, a narasta w dalszej fazie lotu. Można to skorelować z wykresami z Rysunku 7.6 gdzie przedstawiono ciąg generowany przez odpalenia kolejnych silników korekcyjnych dla przypadków 1 oraz 3.



(a) Przypadek 1 (b) Przypadek 3

Rysunek 7.6: Siła generowana przez dyskretny układ sterowania w czasie lotu

Jak wynikało z Rysunków 6.7 oraz 7.1 większość energii zużywana jest w początkowej fazie sterowania. Widoczne na Rysunku 7.6 pętle zakreślane przez ciąg generowany przez silniczki są gęstsze na początku okresu sterowania, a odstępy między nimi wzrastają w miarę lotu. Ze względu na dosyć duży ciąg pojedynczego silniczka, ciężko jest dobrze odwzorować mały niezbędny wydatek energii pod koniec trajektorii z wykorzystaniem opracowanej metody konwersji. Ze względu na bardzo dynamiczny charakter ruchu rakiety, zarówno wybór czasu jak i kierunku odpalenia wprowadzają pewien błąd powodujący, że konwersja sterowania z ciągłego na dyskretne nie jest idealna. W każdym z przedziałów odpowiadających jednemu silniczkowi jeden duży impuls ciągu w danym kierunku zadziała trochę inaczej niż mniejsza wartość rozłożona na więcej kierunków. Niemniej jednak uzyskana dokładność trafienia jest bardzo dobra.

7.4 Optymalizacja wyników dyskretyzacji

Wynikiem optymalizacji poza ciągłym sygnałem sterującym była też niezbędna liczba silników korekcyjnych. Wykorzystując ten fakt, aby poprawić jeszcze bardziej do-

kładność trafienia po dyskretyzacji zdecydowano się na przeprowadzenie kolejnej optymalizacji, w której jako zmienne optymalizacyjne wybrano wyznaczone wcześniej czasy i kierunki odpaleń silników korekcyjnych. Ponieważ ilość energii, więc liczba silników, jest ustalona, jako funkcję kosztu wybrano tylko błąd trafienia zgodnie z równaniem:

$$\min_{\mathbf{t}_{sk},\boldsymbol{\phi}_{sk}} J_d(\mathbf{t}_{sk},\boldsymbol{\phi}_{sk}) = \left|\mathbf{X}^{\mathbf{t}_{sk},\boldsymbol{\phi}_{sk}} - \mathbf{X}_{cmd}\right|^2 \tag{7.11}$$

gdzie $\mathbf{t}_{sk}$ jest wektorem czasów odpaleń, $\boldsymbol{\phi}_{sk}$ wektorem kierunków odpaleń, a $\mathbf{X}^{\mathbf{t}_{sk},\boldsymbol{\phi}_{sk}}$ zależnym od nich punktem trafienia. Gradient funkcji celu ponownie wyznaczano z użyciem automatycznego różniczkowania. Maksymalną liczbę iteracji ustalono na 10, gdyż celem była tylko poprawa celności, a same obliczenia ze względu na dużą możliwą ilość zmiennych mogłyby trwać dosyć długo. Przeprowadzono obliczenia dla tych samych 10 przypadków co przy testowaniu procedury dyskretyzacji, a wyniki w postaci porównania błędów trafienia przedstawia Tabela 7.2.

Tabela 7.2: Wynikowe błędy trafienia po optymalizacji wyników dyskretyzacji

Przypadek	Błąd trafienia przed optymalizacją [m]	Błąd trafienia po optymalizacji[m]
1	13,9	11,3
2	6,59	6,59
3	5,43	5,27
4	9,08	9,08
5	7,83	7,59
6	3,31	2,80
7	7,23	6,29
8	4,18	3,92
9	2,00	1,89
10	9,30	9,29
Średnia	6,88	6,40

Średni błąd trafienia zmalał o około 0.5 metra. Widoczne jest także, że dla części przypadków optymalizator nie znalazł lepszego rozwiązania. Tak niewielkie różnice potwierdzają, że mimo pewnych niedoskonałości procedury dyskretyzacji, uzyskane za jej pomocą czasy i kierunki odpaleń silników korekcyjnych są dokładne.

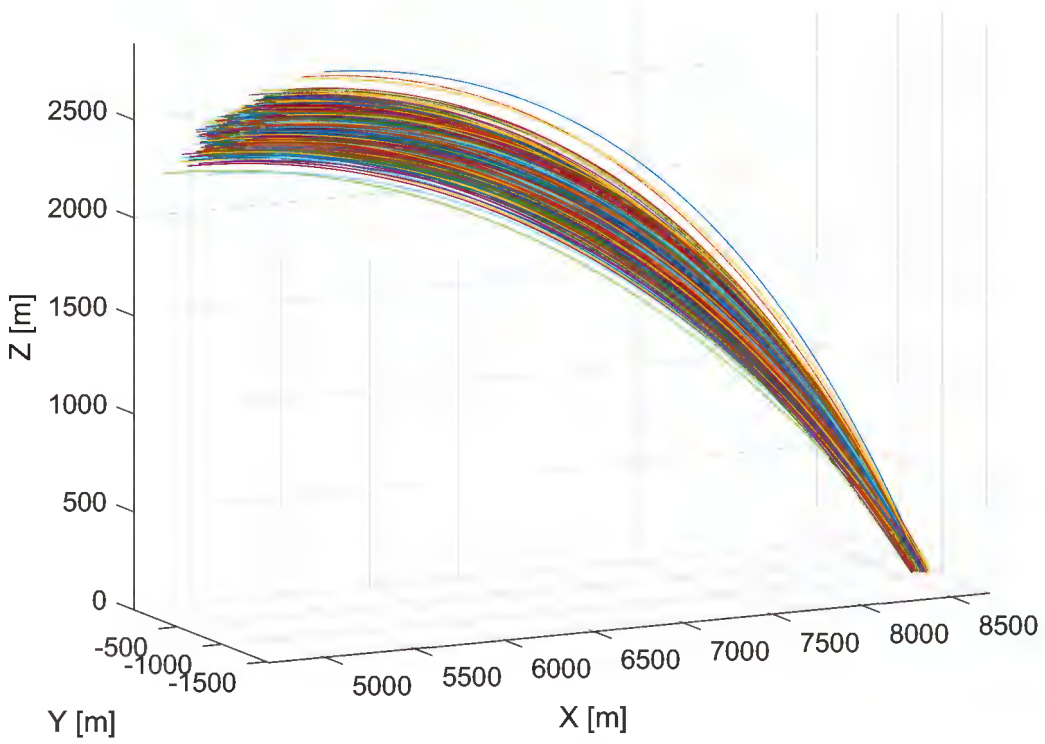
8 Algorytm sterowania

Do tej pory sterowanie odbywało się w pętli otwartej. Wyznaczane były czasy i kierunki odpaleń silników, które spełnią warunek optymalizacji, mając do dyspozycji pełną informację o modelu rakiety oraz warunkach środowiska. W rzeczywistości taka sytuacja jest mało prawdopodobna, z tego względu należało opracować metodę zamknięcia pętli sprzężenia zwrotnego w czasie lotu, aby uwzględnić niepewności wynikające z rzeczywistości. Zdecydowano się to zrobić opracowując algorytm bazujący na bazie danych rozwiązań uzyskanych w procesie optymalizacji. Rakieta dysponując taką bazą oraz informacjami o swoim stanie z czujników pokładowych i algorytmów nawigacyjnych, wybierać będzie to rozwiązanie, które aktualnie wyda się najlepsze. Dzięki temu podejściu powinno się otrzymać zarówno korzyści wynikające z nieliniowej optymalizacji, która daje dokładne rozwiązania, ale kosztem dużego czasu obliczeń oraz działania w pętli otwartej, z korzyściami sterowania w pętli zamkniętej, które bierze pod uwagę aktualnie panujące warunki lotu.

8.1 Generacja bazy danych

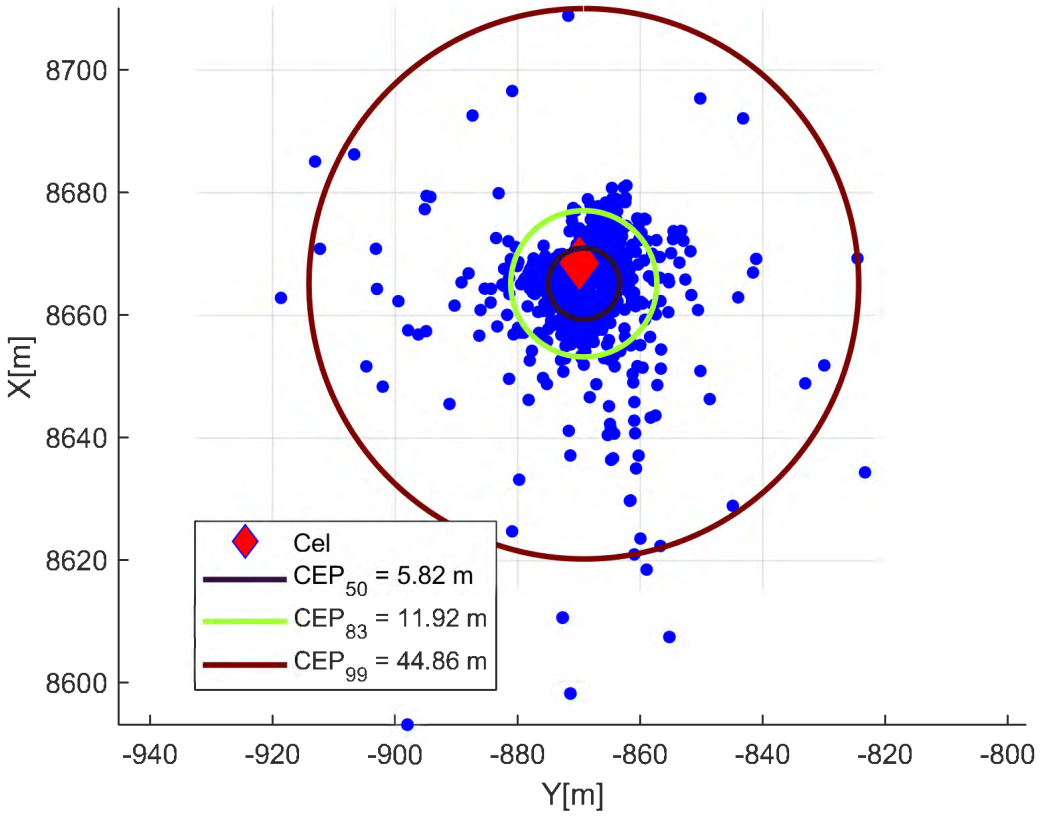
Procedura generacji bazy danych sprowadzała się do przeprowadzenia szeregu symulacji optymalizacyjnych. Proces ten przedstawiono schematycznie na Rysunku 8.1. Każda z symulacji wyglądała w następujący sposób. Najpierw ustawiano wyrzutnie na odpowiednie kąty strzału azymutu i elewacji, a następnie generowana była trajektoria referencyjna, której ostatni punkt stanowił współrzędne celu. Następnie zaburzano parametry modelu zgodnie z Tabelą 6.1, puszczano symulację do momentu osiągnięcia przez rakietę apogeum, jednocześnie zapisując całą trajektorie wznoszącą do pliku tekstowego. Trajektoria ta wchodziła później w skład trajektorii referencyjnych w bazie danych. Stan rakiety w apogeum wykorzystywany był następnie jako punkt początkowy dla optymalizacji. Wartości parametrów sterujących rakiety wyznaczane były następnie korzystając z metody optymalizacji pośredniej. Jej wyniki zamieniane były na czasy i kierunki odpaleń silników korekcyjnych korzystając z procedury dyskretyzacji, a następnie były dodatkowo optymalizowane. Na koniec, wynik w postaci ilości niezbędnych silników, a także sekwencji czasów i kierunków odpaleń również zapisywany był do pliku tekstowego oraz wiązany był z wcześniej zapisaną trajektorią wznoszącą. W ten sposób przeprowadzono obliczenia dla 944 przypadków. Rysunek 8.2 przedstawia trajektorie lotu rakiet pod wpływem działania dyskretnego układu sterowania będącego wynikiem całej procedury optymalizacji i dyskretyzacji. Zaczynają się one w momencie apogeum i wszystkie zbiegają się mniej więcej w jednym punkcie będącym celem trafienia.

Rysunek 8.1: Procedura generacji bazy danych



Rysunek 8.2: Trajektorie lotu uzyskane pod wpływem działania dyskretnego układu sterowania

Rysunek 8.3 przedstawia rozrzut charakteryzowany parametrem CEP (kolorowe okręgi), w postaci naniesionych na płaszczyznę ostatnich punktów trajektorii zaznaczonych niebieskimi markerami. Wartość CEP_{50} wyniosła niecałe 6 metrów, CEP_{83} niecałe 12 metrów, a CEP_{99} około 45 metrów. Są to wartości bardzo dobre biorąc pod uwagę impulsowy charakter sterowania rakietą. Cel trafienia oznaczony czerwonym rombem znajduję się wewnątrz najmniejszego okręgu co oznacza dużą precyzję trafień, a małe wartości parametrów CEP oznaczają także dużą dokładność algorytmu.



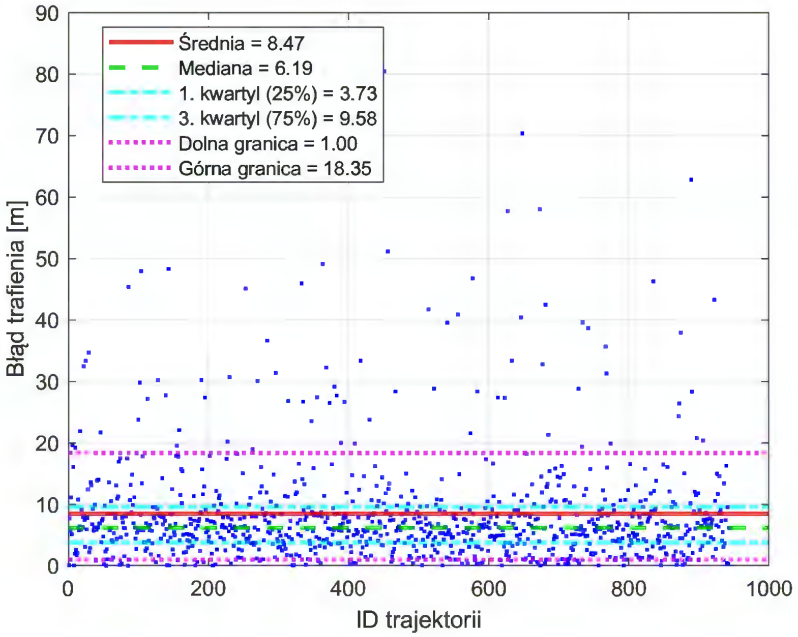
Rysunek 8.3: Rozrzut punktów trafienia trajektorii z bazy danych

Rysunek 8.4 przedstawia statystyczne dane dotyczące błędu trafienia dla wszystkich trajektorii. Średnia wartość błędu wyniosła około $8,5$ metra, a mediana nieco ponad 6 metrów. Pierwszy Q_1 oraz trzeci Q_3 kwartyl oznaczające odpowiednio 25% oraz 75% danych wynoszą $Q_1 = 3,73$ metra oraz $Q_3 = 9,58$ metra. Dolna L oraz górna U granica danych zostały wyznaczone z równań:

$$L = Q_1 - 1,5\,(Q_3 - Q_1) \tag{8.1}$$

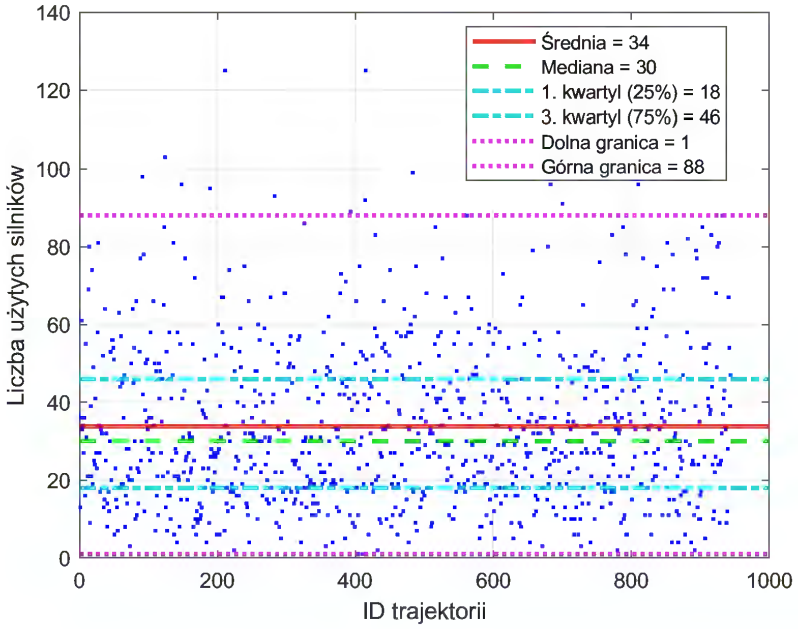
$$U = Q_3 + 1,5\,(Q_3 - Q_1) \tag{8.2}$$

i wynoszą odpowiednio $L = 1,0$ metr oraz $U = 18,35$ metra. Pozostałe wartości traktowano jako odstające. Dodatkowo minimalny błąd trafienia wyniósł poniżej 1 centymetra, a maksymalny ponad 80 metrów.



Rysunek 8.4: Błąd trafienia trajektorii z bazy danych

Podobną analizę statystyczną przeprowadzono dla ilości wykorzystanych silników korekcyjnych i przedstawiono ją na Rysunku 8.5. Średnia wartość użytych silników korekcyjnych wyniosła 34, a mediana 30. Pierwszy Q_1 oraz trzeci Q_3 kwartyl wynoszą $Q_1 = 18$ oraz $Q_3 = 46$. Dolna oraz górna granica wynoszą odpowiednio $L = 1$ oraz $U = 88$. Pozostałe wartości również traktowano jako odstające. Dodatkowo minimalna ilość użytych silników wyniosła 1, a maksymalna 125.



Rysunek 8.5: Liczba użytych silników korekcyjnych dla trajektorii z bazy danych

Opracowana w ten sposób baza danych wykorzystywana była w docelowym algorytmie sterowania.

8.2 Opracowanie algorytmu

W skład bazy danych wchodził zapis trajektorii wznoszącej oraz ilość, czasy i kierunki odpaleń silników korekcyjnych. Przed startem rakiety do pamięci algorytmu sterującego wczytywane były te dane dla każdego z obliczonych przypadków symulacyjnych. Aby ograniczyć trochę wykorzystanie pamięci punkty z trajektorii wznoszących wczytywane były z mniejszą częstotliwości niż były zapisywane. Ustalono, że pojedyncza trajektoria składać się może z 3000 punktów. Sam algorytm sterujący składał się z czterech części:

- Zapisanie w pamięci przebiegu trajektorii wznoszącej
- Wyznaczenie odpowiedniej trajektorii referencyjnej z bazy danych
- Sterowanie w pętli otwartej zgodnie z rozwiązaniem z bazy danych
- Korekta lotu w ostatniej fazie z wykorzystaniem algorytmu nawigacji proporcjonalnej

Lot wznoszący trwał średnio nieco ponad 21 sekund. W czasie jego trwania, po końcu pracy silnika głównego i ustabilizowaniu się trajektorii, od 3 sekundy lotu do 18 sekundy zapisywano do pamięci przebieg trajektorii rakiety z częstotliwością 10 Hz, zgodnie z Algorytmem 8.1.

Algorytm 8.1 Zapis trajektorii wznoszącej do pamięci

1: **if** $t > 3.0$ **and** $t < 18.0$ **then**
2: **if** Czas ostatniego zapisu ≥ 0.1 sekundy (częstotliwość 10Hz) **then**
3: $t_{mem}(i) = t$
4: $X_{mem}(i) = X$
5: $Y_{mem}(i) = Y$
6: $Z_{mem}(i) = Z$
7: i_{++}
8: **end if**
9: **end if**

Następnie po 18 sekundzie lotu wyznaczana była trajektoria referencyjna z bazy danych, która była najbardziej zbliżona, w sensie średnio-kwadratowym, do tej zapisanej w pamięci, zgodnie z Algorytmem 8.2.

Algorytm 8.2 Wybór najlepiej pasującej trajektorii referencyjnej

1: **if** $t > 18.0$ **then**
2: **for all** Trajektorie referencyjne i **do**
3: $RMS_i = 0.0$
4: **for all** Zarejestrowane punkty czasowe $t_{mem}(j)$ **do**
5: Oblicz pozycję (X, Y, Z) trajektorii referencyjnej i dla czasu $t_{mem}(j)$ stosując interpolację liniową
6: $RMS_i += \sqrt{(X_{mem}(j) - X)^2 + (Y_{mem}(j) - Y)^2 + (Z_{mem}(j) - Z)^2}$
7: **end for**
8: $RMS_i = \sqrt{RMS_i}$
9: **if** $RMS_i <$ minimalny błąd **then**
10: Zaktualizuj minimalny błąd
11: Zapamiętaj indeks i trajektorii referencyjnej
12: **end if**
13: **end for**
14: Oznacz trajektorię referencyjną jako znalezioną
15: **end if**

Po znalezieniu trajektorii referencyjnej zapisywano do pamięci odpowiadające jej dane o czasach i kierunkach odpaleń silników korekcyjnych, a także ich liczbie. Następnie po osiągnięciu przez rakietę apogeum rozpoczynano sterowanie w pętli otwartej. Odpalano kolejne silniki korekcyjne zgodnie z Algorytmem 7.1 opisanym w rozdziale 7.3. Ze względu na to, że spodziewano się niedokładności trafienia wynikających, po pierwsze z algorytmu dyskretyzacji co pokazano wcześniej, ale także z tego, że istnieje błąd między aktualną trajektorią wznoszącą, a tą wyznaczoną z bazy danych, postanowiono w ostatniej fazie lotu przełączyć się na algorytm nawigacji proporcjonalnej opisany w rozdziale 5.2. Pominięto przy tym konwersje układów odniesienia niezbędne przy korzystaniu z danych z czujników inercjalnych, gdyż pomięto ich model, a wykorzystano tylko część odpowiadającą za naprowadzanie się na cel. W tym celu do danych wejściowych dołożono znane współrzędne celu $(X_{cmd}, Y_{cmd}, Z_{cmd})$, a przełączanie między algorytmami realizowano na podstawie aktualnej wysokości lotu zgodnie z Algorytmem 8.3.

Algorytm 8.3 Przełączanie między algorytmami sterowania

1: **if** Apogeum **then**
2: **if** $H > H_{thr}$ **then**
3: Sterowanie w pętli otwartej
4: **else**
5: Sterowanie z wykorzystaniem nawigacji proporcjonalnej
6: **end if**
7: **end if**

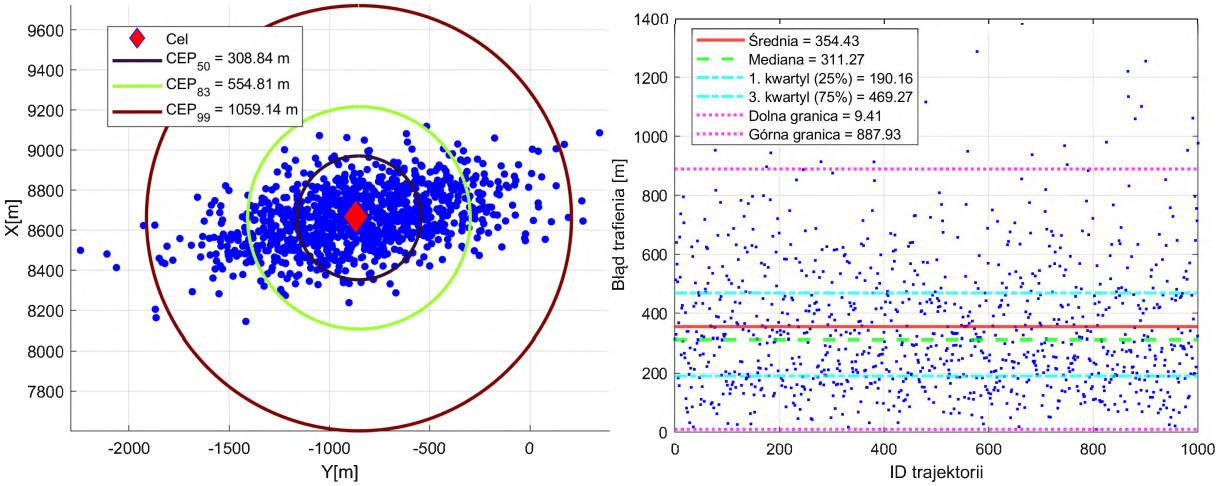
Wykorzystaną procedurę nawigacji proporcjonalnej przedstawia Algorytm 8.4. Wyznaczone przez nią czas i kierunek odpalenia kolejnych silników korekcyjnych trafiały do fizycznego modelu silnika korekcyjnego opisanego Algorytmem 7.2.

Algorytm 8.4 Naprowadzanie metodą nawigacji proporcjonalnej z wykorzystaniem silników korekcyjnych

Wyznacz wektor odległości do celu $\mathbf{R}_{TM}$
Wyznacz wektor prędkości zbliżania $\mathbf{V}_c$ (równania (5.7)-(5.9))
Wyznacz wartość kątów linii wizowania oraz ich pochodnych (równania (5.10)-(5.15))
Wyznacz wartość przyspieszenia normalnego do linii wizowania (równanie (5.6))
Wyznacz wartość zadanego przyspieszenia bocznego (równanie (5.16))
Wyznacz kierunek odpalenia $\phi_{sk_i} = \text{atan2}\left(a_{CMD_z}, a_{CMD_y}\right) - P\tau_{sk}$
if $|\mathbf{a}_{CMD}| > a_{th}$ oraz $t - t_{last} > \tau$ oraz $t > t_g$ **then**
  if Liczba odpalonych silników korekcyjnych nie przekroczyła ich zadanej ilości **then**
    Zapisz czas odpalenia t_{start_i}
    Zapisz kierunek odpalenia ϕ_{sk_i}
    Zapisz kąt przechylenia rakiety ϕ_{start_i}
    Oznacz silnik jako odpalony
  end if
end if

8.3 Wyniki działania algorytmu

Jako rezultaty odniesienia traktowano wartości parametru CEP dla lotu niesterowanego oraz sterowanego w całości z użyciem algorytmu nawigacji proporcjonalnej PNG. Rakietę wyposażono w 32 silniki korekcyjne, bazując na opracowanej konstrukcji RPT ORION, a sterowanie uruchamiano w momencie osiągnięcia apogeum.

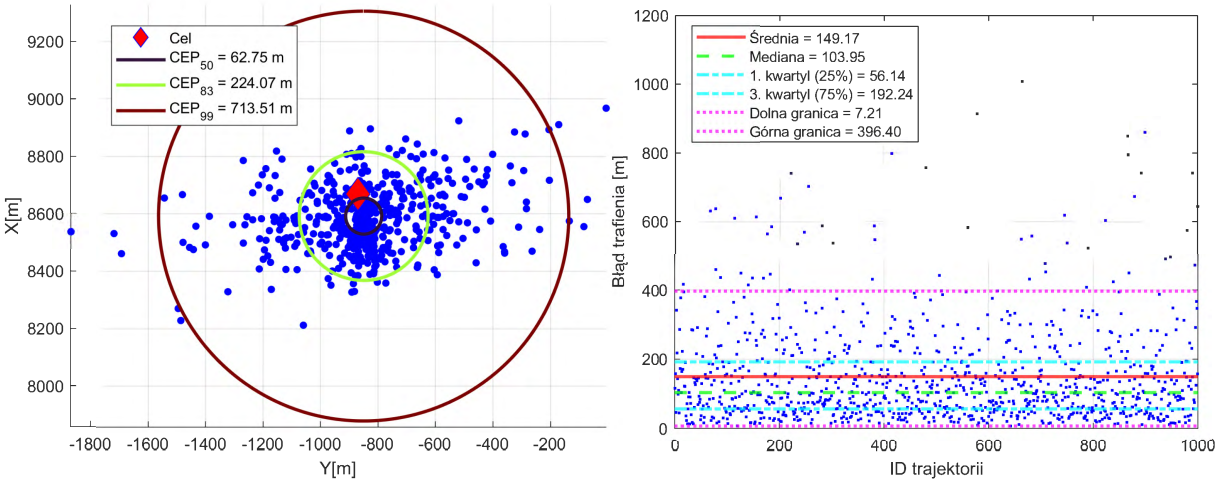


(a) Rozrzut (b) Statystyka

Rysunek 8.6: Wyniki rozrzutu dla przypadku niesterowanego

Rysunek 8.6 przedstawia rozrzut miejsc upadku dla przypadku lotu niesterowanego. Po lewej stronie przedstawiono okręgi symbolizujące parametr CEP, a po prawej stronie jego analizę statystyczną. Parametr CEP_{50} wyniósł ponad 300 metrów, CEP_{83} ponad 550 metrów, a CEP_{99} ponad 1000 metrów. Średni błąd upadku wyniósł ponad 350 metrów.

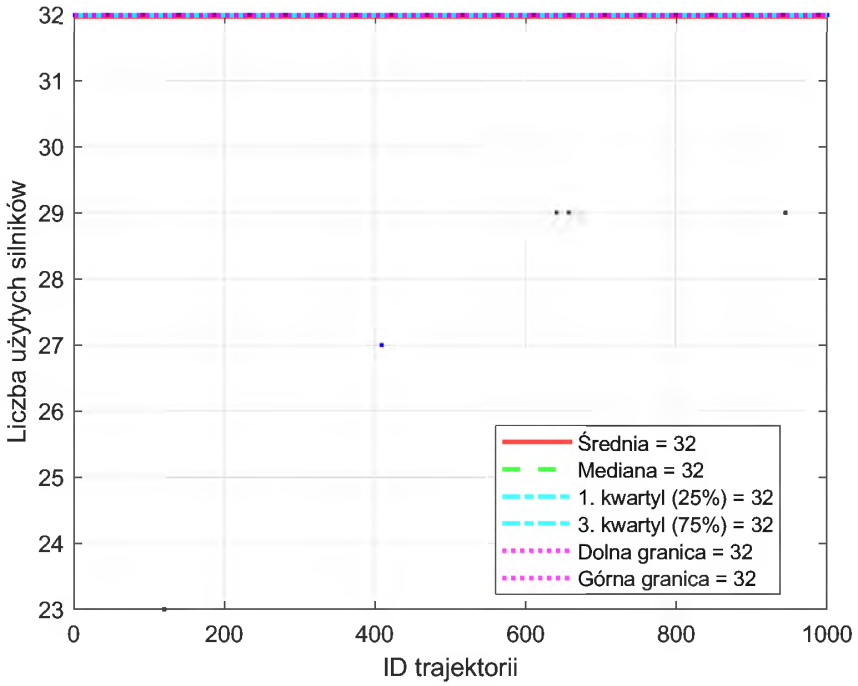
Są to wartości porównywalne z wartościami osiągniętymi przy symulacji wykorzystującej model w Simulinku z Rysunku 5.13.



(a) Rozrzut (b) Statystyka

Rysunek 8.7: Wyniki rozrzutu dla przypadku sterowanego 32 silnikami z algorytmem PNG

Rysunek 8.7 przedstawia rozrzut miejsc upadku dla przypadku lotu sterowanego algorytmem PNG z wykorzystaniem 32 silników korekcyjnych. Parametr CEP_{50} wyniósł nieco ponad 60 metrów dając zysk około 80%, CEP_{83} ponad 220 metrów, czyli około 50% lepszy rezultat, a CEP_{99} ponad 700 metrów, czyli około 30% lepiej. Średni błąd upadku wyniósł ponad 150 metrów. Wartości te są również porównywalne z wartościami osiągniętymi przy symulacji wykorzystującej model w Simulinku z Rysunku 5.13. Stanowić one będą wartość odniesienia dla algorytmu wykorzystującego bazę danych. Dodatkowo Rysunek 8.8 pokazuje ilość wykorzystanych silników korekcyjnych w trakcie lotu. W znaczącej większości przypadku algorytm wykorzystał wszystkie dostępne silniki. Rozrzuty miejsc upadku z symulacji wykorzystujących algorytm opierający się o bazę danych zostaną wyznaczone dla kilku przypadków. Sprawdzony zostanie wpływ czasu przełączenia między algorytmami, wpływ wielkości bazy danych oraz ilość dostępnych silników korekcyjnych na celność trafienia.



Rysunek 8.8: Zużycie silników korekcyjnych dla przypadku sterowanego 32 silnikami z algorytmem PNG

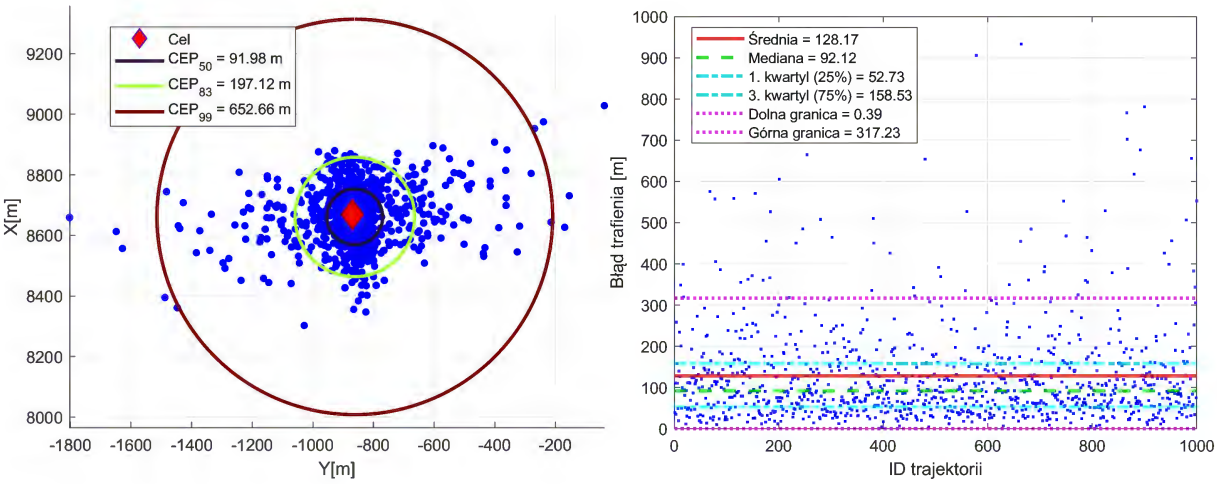
8.3.1 Wpływ czasu przełączenia między algorytmami

W pierwszej kolejności zbadano wpływ czasu przełączania między algorytmami. Tabela 8.1 przedstawia moment zmiany algorytmu w postaci szacowanego czasu do uderzenia, wysokości lotu, oraz szacowanego procentowego udziału trajektorii wykorzystującej algorytm PNG. Wartości te bazowano na opadającej części trajektorii referencyjnej (niesterowanej). Przypadek pierwszy sprawdzał działanie algorytmu wykorzystującego tylko bazę danych, bez korekcji z algorytmem PNG.

Tabela 8.1: Momenty przełączania między algorytmami

Przypadek	Czas do trafienia [s]	Wysokość lotu [m]	% lotu PNG
1	0	0	0
2	2	500	8
3	5	1000	20
4	10	1500	40

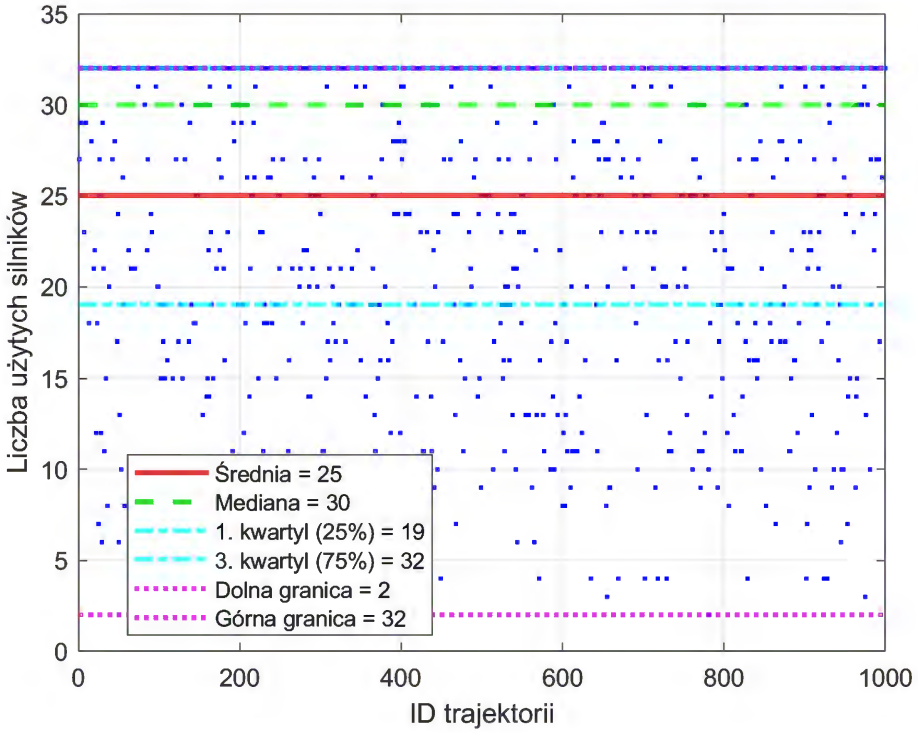
Dla każdego z przypadków puszczono 1000 symulacji Monte Carlo z takimi samymi wartościami niepewności jak dla przypadków niesterowanego i sterowanego w całości z użyciem algorytmu PNG.



(a) Rozrzut (b) Statystyka

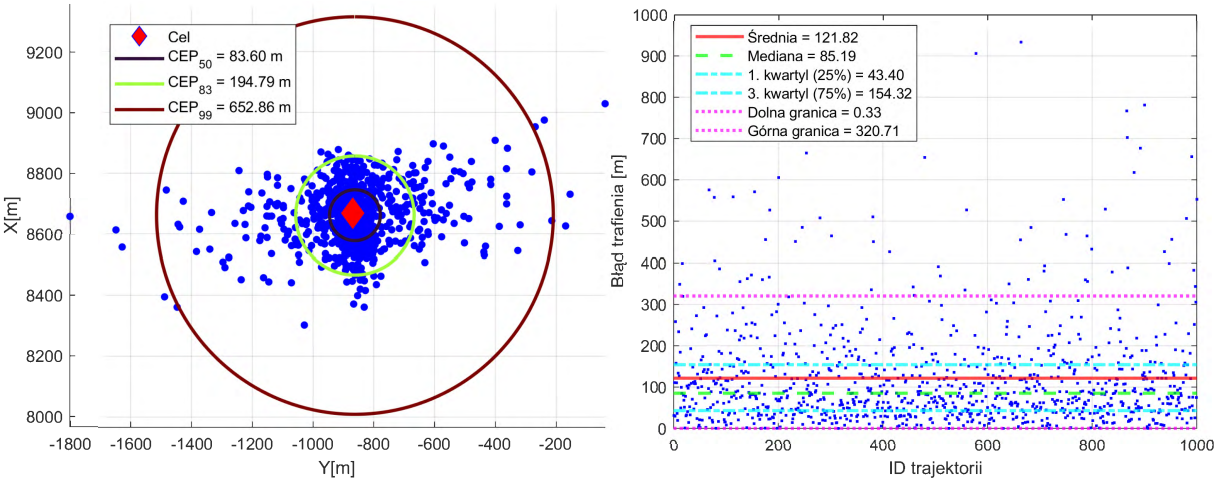
Rysunek 8.9: Wyniki rozrzutu dla przypadku bez przełączania algorytmów

Rysunek 8.9 przedstawia rozrzuty i ich analizę statystyczną dla przypadku wykorzystywania tylko algorytmu bazującego na bazie danych. Parametr CEP_{50} wyniósł nieco ponad 90 metrów, dając wynik gorszy niż czyste PNG o około 50%, CEP_{83} niecałe 200 metrów, czyli około 10% lepszy rezultat, a CEP_{99} ponad 650 metrów, czyli około 8% lepiej. Średni błąd upadku wyniósł ponad 128 metrów. Wykorzystanie algorytmu z bazą danych pogorszyło wyniki jeśli chodzi o wartość CEP_{50}, natomiast poprawiło nieznacznie dokładność wszystkich trafień zmniejszając wartości pozostałych parametrów.



Rysunek 8.10: Zużycie silników korekcyjnych dla przypadku bez przełączania algorytmów

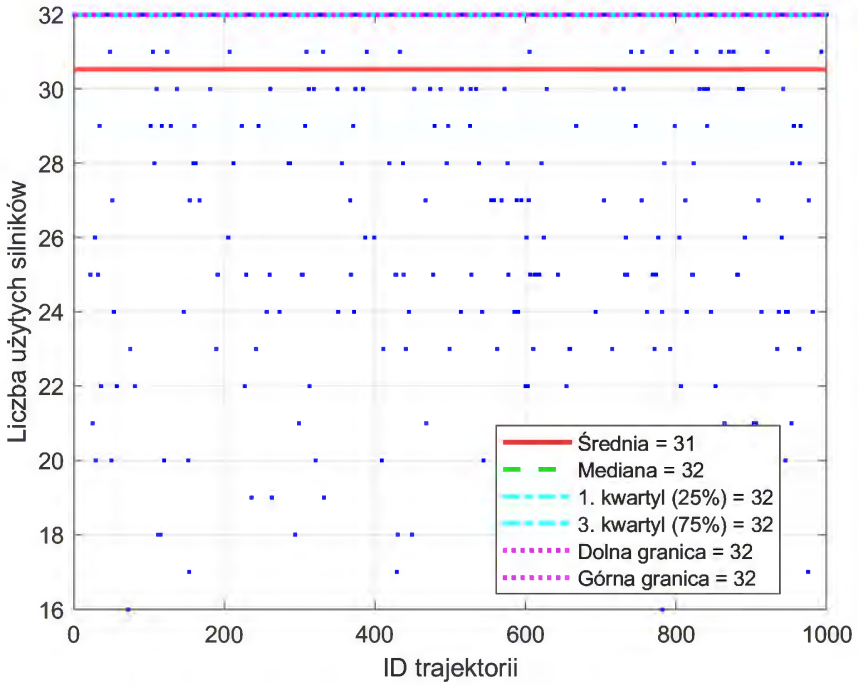
Rysunek 8.10 przedstawia wykorzystanie silników korekcyjnych. Można zauważyć, że średnio na lot wykorzystywano tylko 25 silników, co wynika z rozłożenia energii w trakcie lotu i było omawiane przy wynikach algorytmu dyskretyzacji. Ponieważ, w wielu przypadkach można by było wykorzystać więcej silników, które i tak są dostępne, można się spodziewać poprawy tych wyników dokładając jeszcze sterowanie z algorytmem PNG w końcowej fazie lotu.



(a) Rozrzut (b) Statystyka

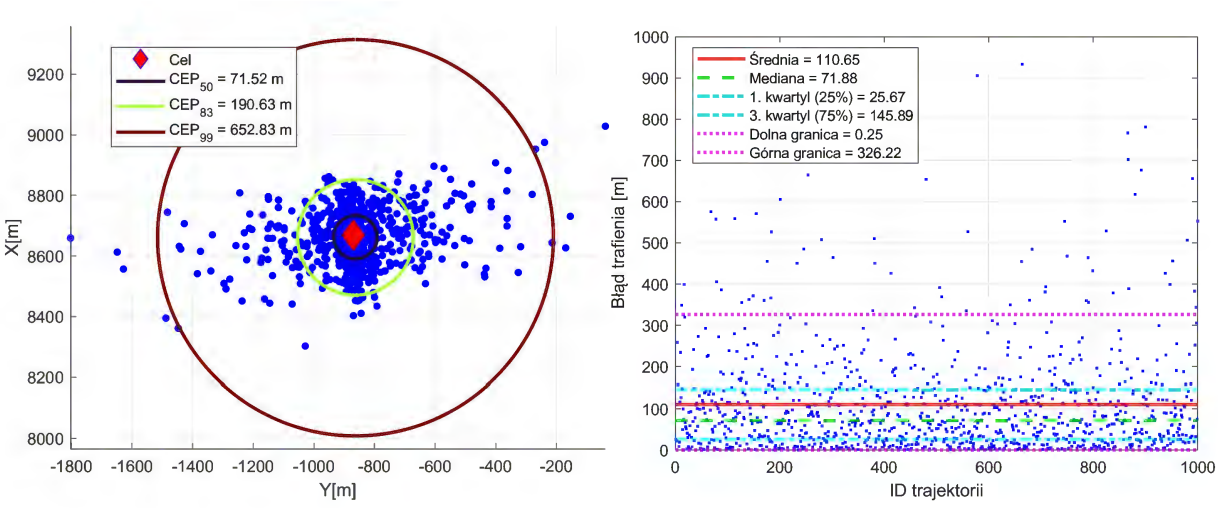
Rysunek 8.11: Wyniki rozrzutu dla przypadku przełączenia na wysokości 500 metrów

Rysunek 8.11 przedstawia rozrzuty i ich analizę statystyczną dla przypadku gdy przełączenie między algorytmami nastąpiło przy wysokości lotu wynoszącej 500 metrów. Parametr CEP_{50} wyniósł nieco ponad 80 metrów, będąc około 33% gorszy niż PNG, a parametry CEP_{83} oraz CEP_{99} dały bardzo zbliżone rezultaty do przypadku bez przełączania. Średni błąd upadku wyniósł ponad 121 metrów. Zachowując dokładność wszystkich trafień udało się zmniejszyć parametr CEP_{50}, lecz wciąż uzyskując wynik gorszy niż czyste PNG.



Rysunek 8.12: Zużycie silników korekcyjnych dla przypadku przełączenia na wysokości 500 metrów

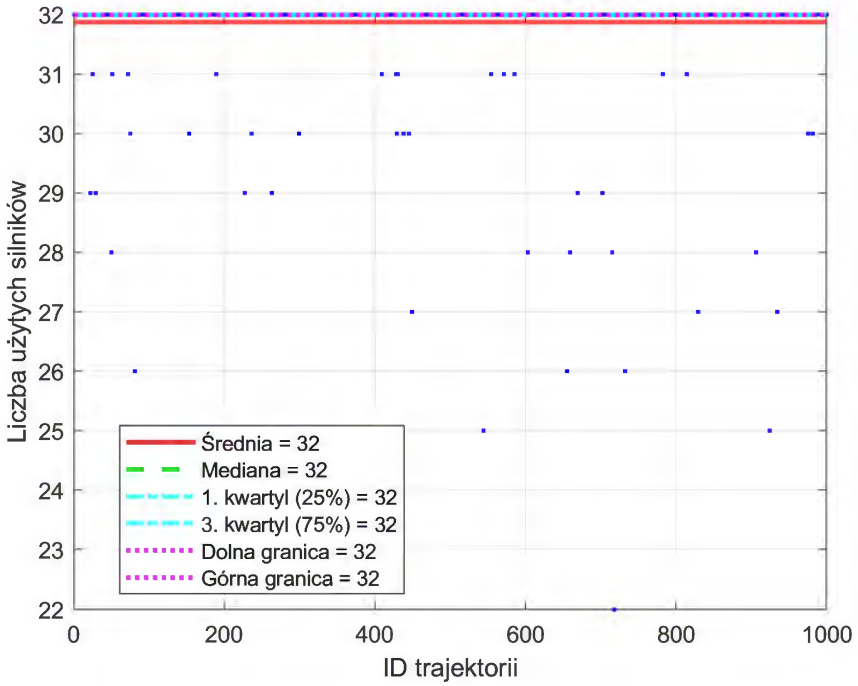
Rysunek 8.12 przedstawia wykorzystanie silników korekcyjnych. Średnie zużycie wzrosło do 31 silników na lot. Z tego względu dalej można przesuwać granicę zmiany algorytmów.



(a) Rozrzut (b) Statystyka

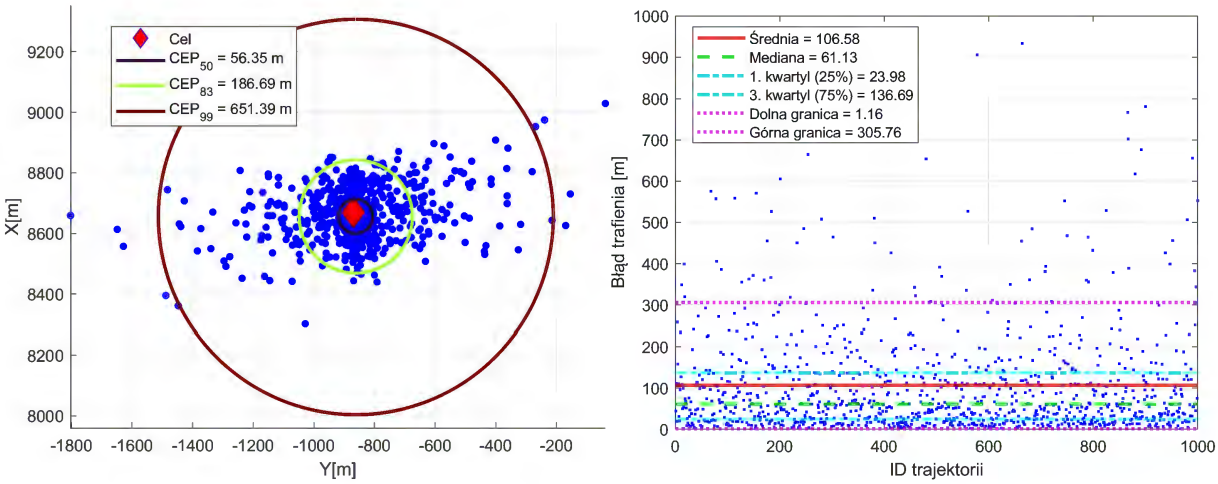
Rysunek 8.13: Wyniki rozrzutu dla przypadku przełączenia na wysokości 1000 metrów

Rysunek 8.13 przedstawia kolejny przypadek gdzie przełączenie między algorytmami wystąpiło na wysokości lotu wynoszącej 1000 metrów. Parametr CEP_{50} wyniósł nieco ponad 70 metrów, powodując że różnica względem PNG zmalała do 14%, zachowując mniej więcej wartości parametrów CEP_{83} oraz CEP_{99} lotu bez przełączania. Średni błąd upadku zmalał do 110 metrów.



Rysunek 8.14: Zużycie silników korekcyjnych dla przypadku przełączenia na wysokości 1000 metrów

Analizując wykorzystanie silników korekcyjnych z Rysunku 8.14, gdzie w większości przypadków użyto wszystkich dostępnych silników zdecydowano się na jeszcze jedno podwyższenie wysokości przełączenia algorytmów.

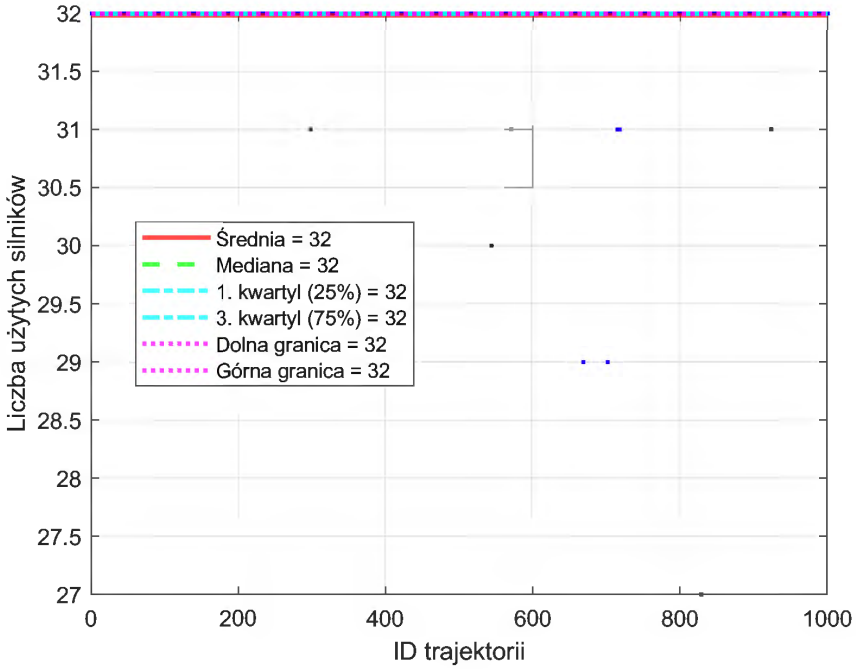


(a) Rozrzut (b) Statystyka

Rysunek 8.15: Wyniki rozrzutu dla przypadku przełączenia na wysokości 1500 metrów

Rysunek 8.15 pokazuje wyniki dla momentu przełączenia na wysokości 1500 metrów. Udało się uzyskać parametr CEP_{50} o wartości nieco ponad 55 metrów, co dało wynik lepszy niż PNG o około 11%, zachowując dokładność pozostałych parametrów. Średni błąd upadku ponownie zmalał do 106 metrów. Ostatni rezultat pokazuje, że opracowany algo-

rytm daje lepsze wyniki względem zastosowania klasycznych metod sterowania rakietami wyposażonymi w układy wykorzystujące silniki korekcyjne.



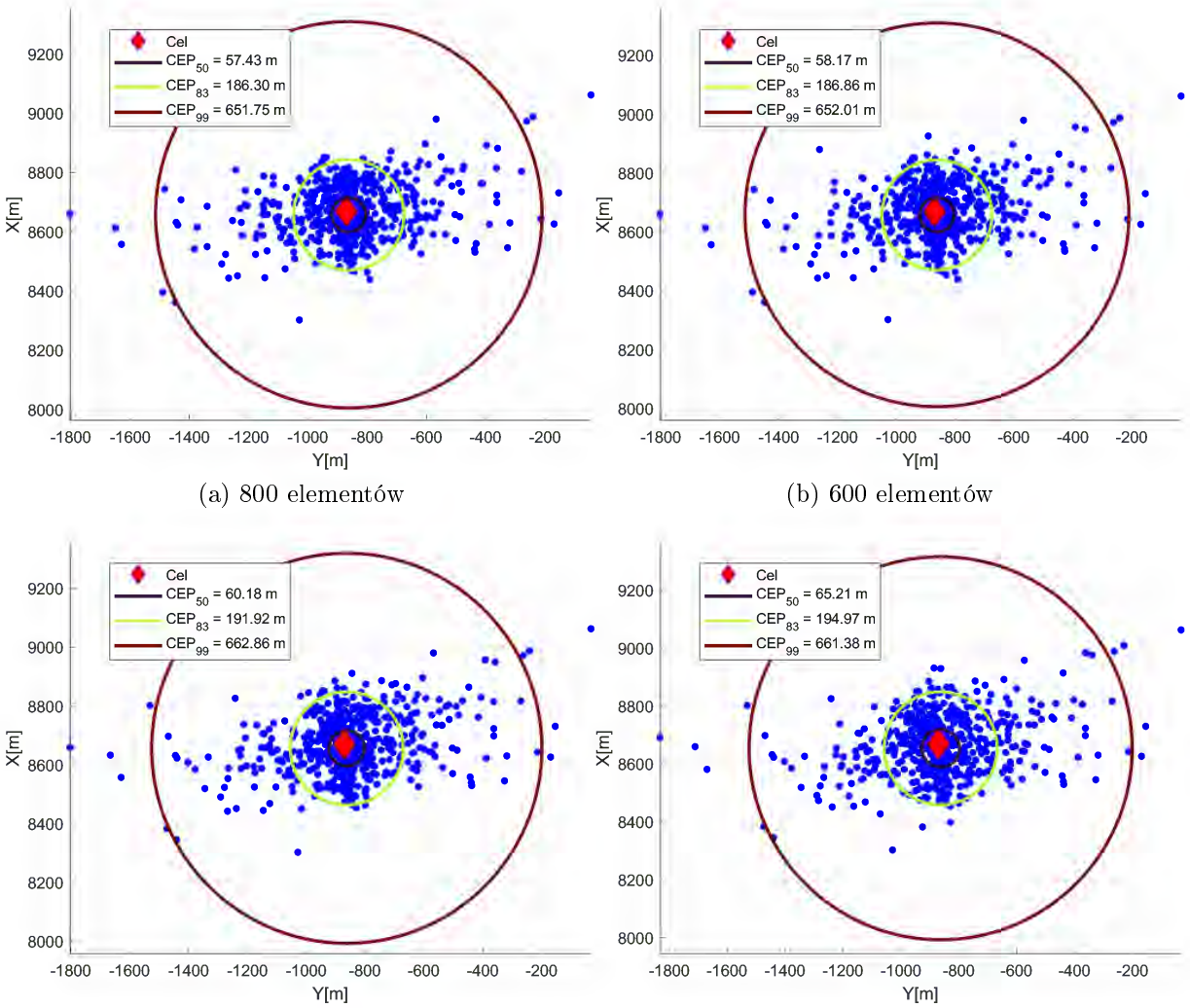
Rysunek 8.16: Zużycie silników korekcyjnych dla przypadku przełączenia na wysokości 1500 metrów

Rysunek 8.16 pokazuje, że w praktycznie każdym przypadku wykorzystana została pełna dostępna ilość silników korekcyjnych, więc dalsze zwiększanie wysokości przełączania nie spowodowałoby zapewne znacząco lepszych rezultatów. Otrzymany wynik wskazuje, że opracowana metoda sterowania lepiej radzi sobie w pierwszej fazie sterowania, zaraz po osiągnięciu przez rakietę apogeum, a algorytm PNG ma większe możliwości w końcowej fazie lotu, korygując błąd wynikający ze sterowania w pętli otwartej.

8.3.2 Wpływ wielkości bazy danych

W dalszej kolejności zbadano wpływ ilości danych znajdujących się w bazie na jakość uzyskiwanych wyników. Wykonując obliczenia optymalizacyjne wygenerowano 944 rozwiązania, które weszły w skład bazy danych algorytmu sterującego. Stosując wysokość przełączania algorytmu wynoszącą 1500 metrów przeprowadzono symulację Monte Carlo, ponownie po 1000 strzałów, dla różnych wielkości wczytanej bazy danych, wynoszącej 800, 600, 400 oraz 200 elementów. W trakcie generowania bazy danych parametry modelu losowane były z użyciem odchyleń standardowych z Tabeli 6.1. Wyniki zapisywane były w odpowiednich folderach, które następnie wczytywane były do algorytmu. Z tego względu, niezależnie od ilości wczytanych elementów, statystyka rozkładu parametrów zapisanej trajektorii wznoszącej powinna zostać zachowana. Zwiększone będą zapewne odległości między trajektoriami, co może spowodować większą różnicę między aktualną trajektorią

wznoszącą podczas działania docelowego algorytmu, więc i większy skumulowany błąd przy locie ze sterowaniem w pętli otwartej.



(a) 800 elementów

(b) 600 elementów

(c) 400 elementów

(d) 200 elementów

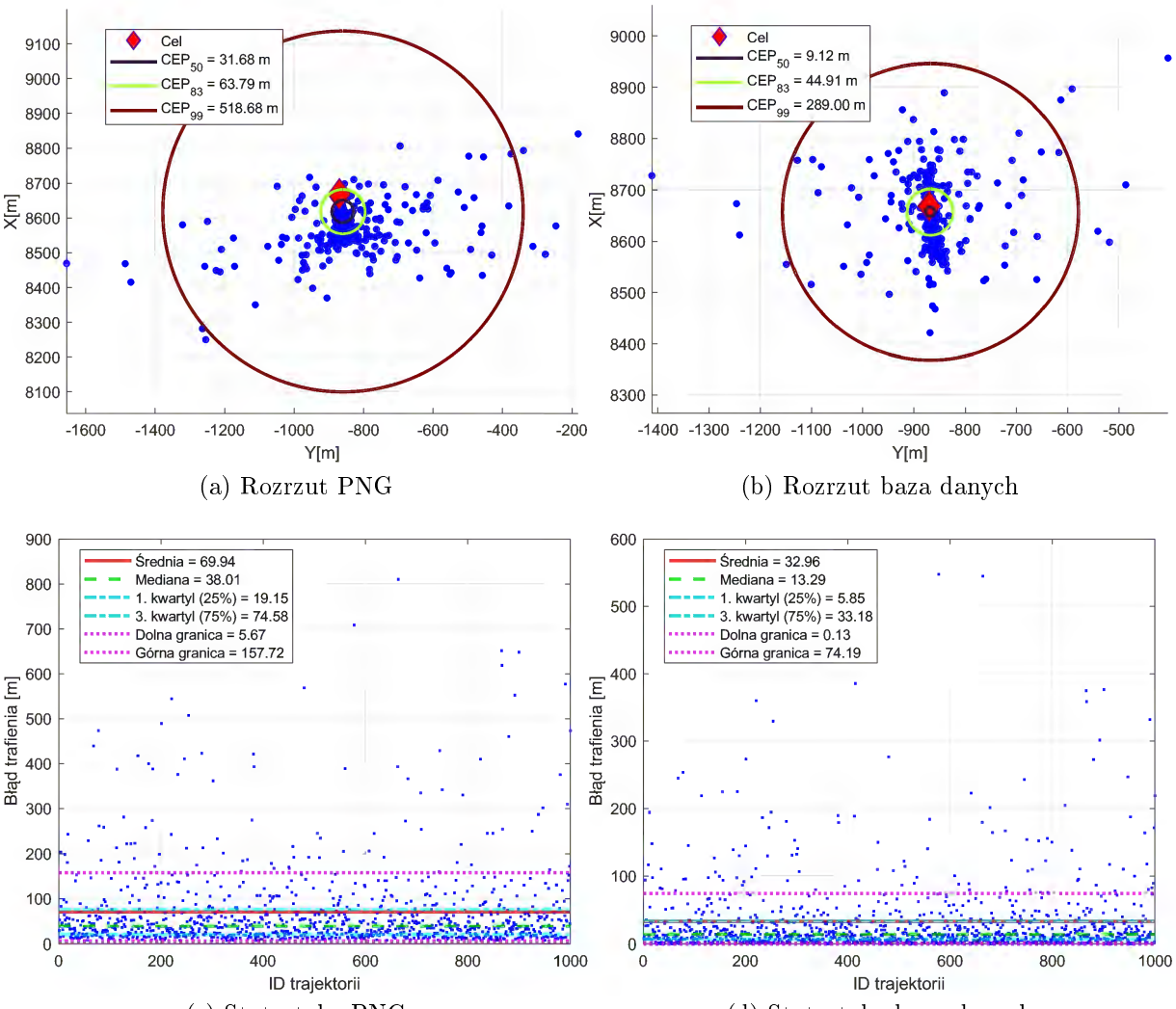
Rysunek 8.17: Wyniki rozrzutu dla różnych przypadków wielkości bazy danych

Rysunek 8.17 przedstawia rozrzuty dla przypadków gdzie baza danych liczyła 800 (a), 600 (b), 400 (c) oraz 200 (d) elementów. Różnice w otrzymanych parametrach CEP nie przekraczają 10 metrów. Wiadomo, że im więcej elementów znajduje się w bazie danych, tym większe jest prawdopodobieństwo znalezienia trajektorii najbardziej zbliżonej do aktualnej trajektorii wznoszącej. Liczba elementów powinna być więc jak największa, ale jej dokładna ilość powinna być podyktowana dostępną pamięcią na docelowym komputerze pokładowym rakiety.

8.3.3 Wpływ dostępnej liczby silników korekcyjnych

Ostatnim przypadkiem testowym było porównanie wpływu dostępnej liczby silników korekcyjnych na zmniejszenie rozrzutu miejsc upadku rakiety. Opracowana konstruk-

cja RPT ORION wyposażona była w moduł sterujący z 32 silnikami korekcyjnymi. Jeszcze w trakcie trwania projektu okazało się, że byłaby możliwość, zachowując długość modułu, gęściej upakować silniki korekcyjne i zwiększyć ich ilość dwukrotnie, do 64 egzemplarzy. Przeprowadzono więc dodatkowe symulacje Monte Carlo, ponownie po 1000 strzałów, zakładając, że rakieta może zostać wyposażona w 64, 96 lub 128 silników korekcyjnych. Wyniki działania algorytmu wykorzystującego pełną bazę danych oraz przełączenie na algorytm PNG w momencie osiągnięcia wysokości 1500 metrów, porównano z działaniem czystego algorytmu nawigacji proporcjonalnej.



(a) Rozrzut PNG

(b) Rozrzut baza danych

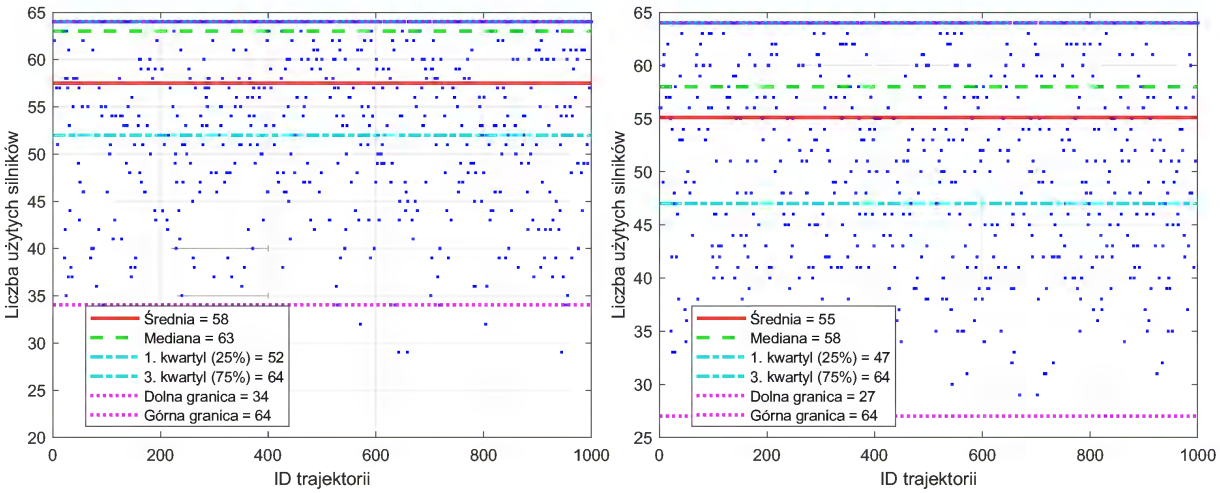
(c) Statystyka PNG

(d) Statystyka baza danych

Rysunek 8.18: Wyniki rozrzutu przy wykorzystaniu 64 silników korekcyjnych

Rysunek 8.18 przedstawia wyniki rozrzutu i analizę statystyczną dla przypadku 64 dostępnych silników korekcyjnych. Wartość CEP_{50} ma wartość około 9 metrów dla bazy danych względem prawie 32 metrów dla PNG, dając zysk ponad 70%. W przypadku parametrów CEP_{83} oraz CEP_{99} zysk wyniósł odpowiednio 30% oraz 45%. Średni błąd trafienia wynoszący 33 metry dla bazy danych jest ponad dwukrotnie mniejszy niż dla

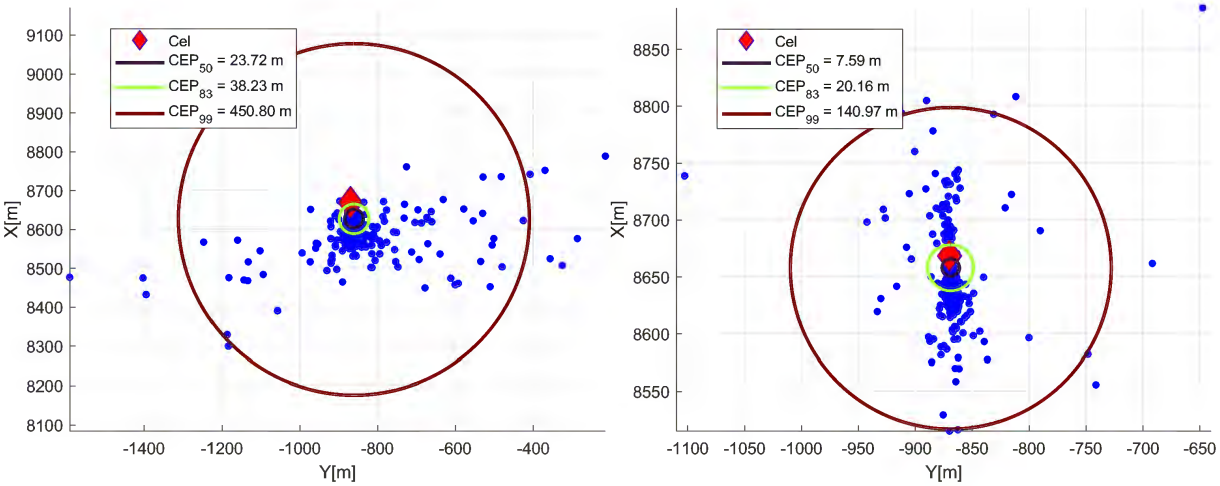
algorytmu PNG.



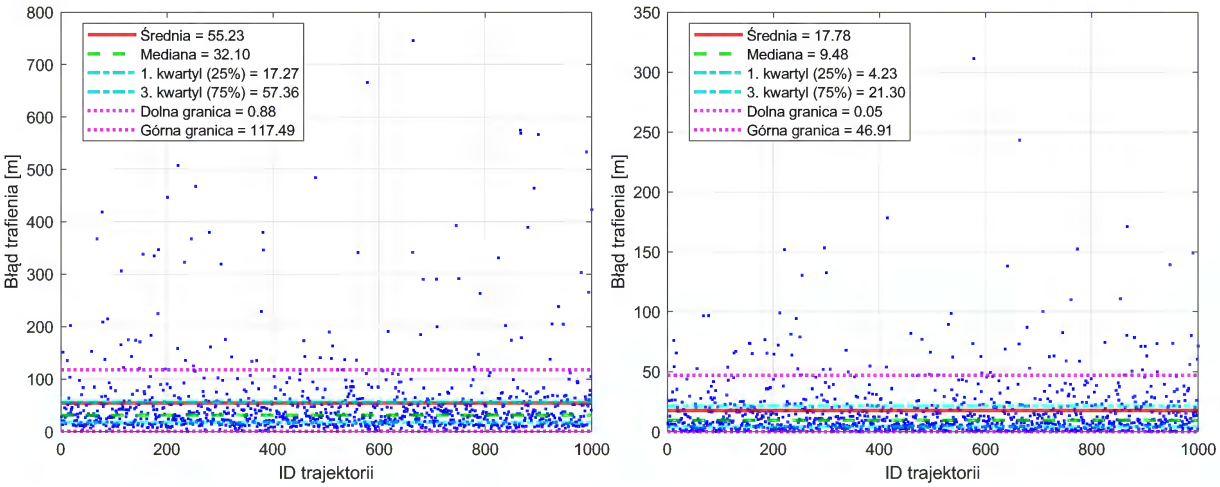
(a) PNG (b) Baza danych

Rysunek 8.19: Ilość wykorzystanych silników korekcyjnych przy 64 dostępnych

Rysunek 8.19 przedstawia liczbę wykorzystanych silników korekcyjnych, która w obu przypadkach wyniosła średnio około 55 silników. Pozostałe statystyki pokazują, że algorytm PNG generalnie używa ich więcej. Ponieważ w bardzo wielu przypadkach nie wszystkie dostępne silniki korekcyjne zostały zużyte można rozważyć ponownie zwiększenie wysokości przełączenia się pomiędzy algorytmami sterowania, co mogłoby spowodować uzyskanie jeszcze lepszych wyników.



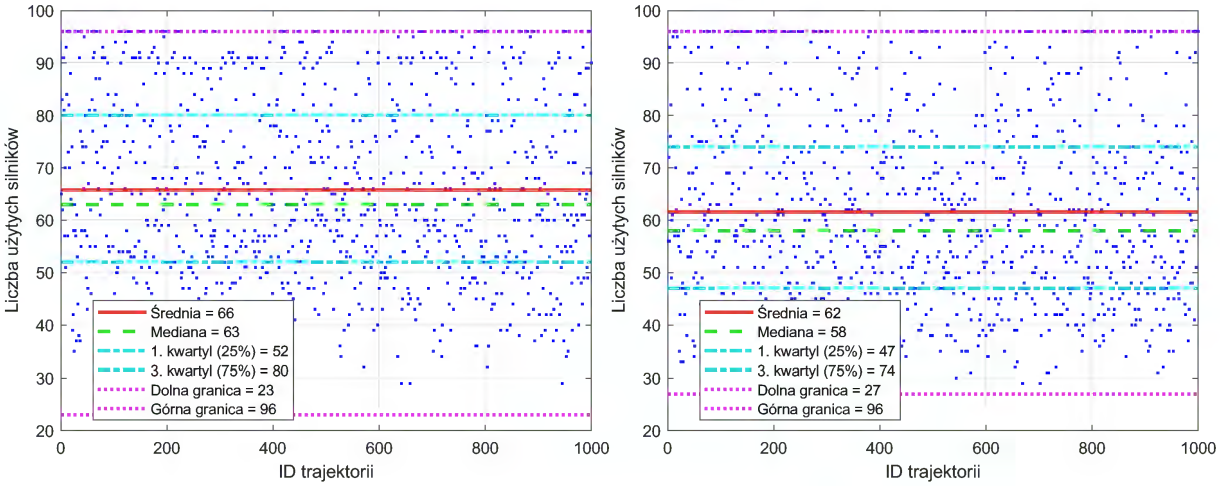
(a) Rozrzut PNG (b) Rozrzut baza danych



(c) Statystyka PNG (d) Statystyka baza danych

Rysunek 8.20: Wyniki rozrzutu przy wykorzystaniu 96 silników korekcyjnych

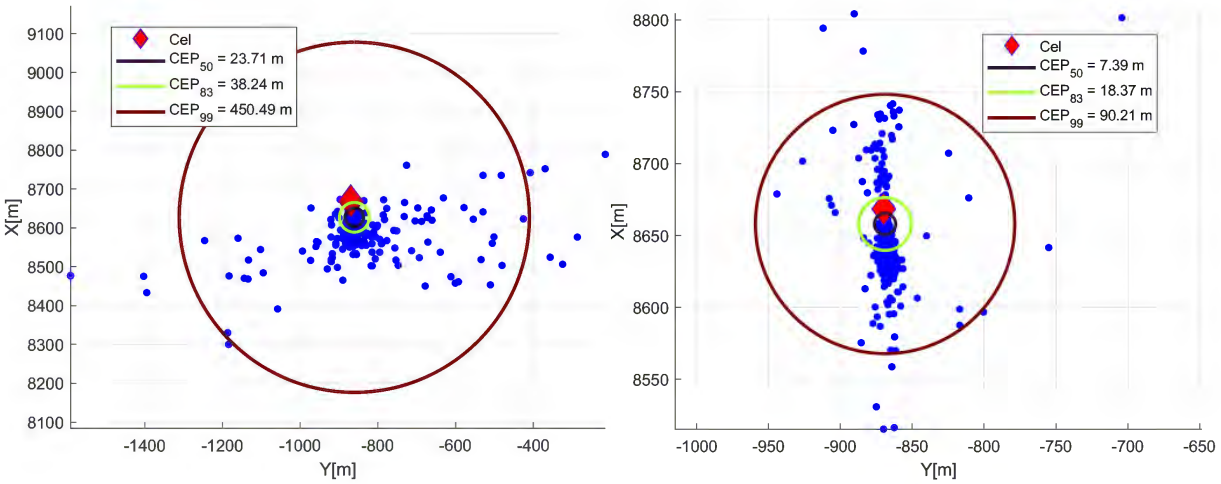
Rysunek 8.20 przedstawia wyniki rozrzutu i analizę statystyczną dla przypadku 96 dostępnych silników korekcyjnych. Wartość CEP_{50} osiągnęła wartość około 7,5 metra dla bazy danych względem prawie 24 metrów dla PNG, co ponownie daje wynik lepszy o około 70%. Dla parametrów CEP_{83} oraz CEP_{99} zysk wyniósł odpowiednio 50% oraz 70%. Średni błąd trafienia wynoszący 18 metrów dla bazy danych jest ponad trzykrotnie mniejszy niż dla algorytmu PNG. O ile wartość CEP_{50} zmniejszyła się nieznacznie, mocnej redukcji uległ parametr CEP_{99} co oznacza, że algorytm drastycznie zwiększył swoją dokładność.



(e) PNG (f) Baza danych

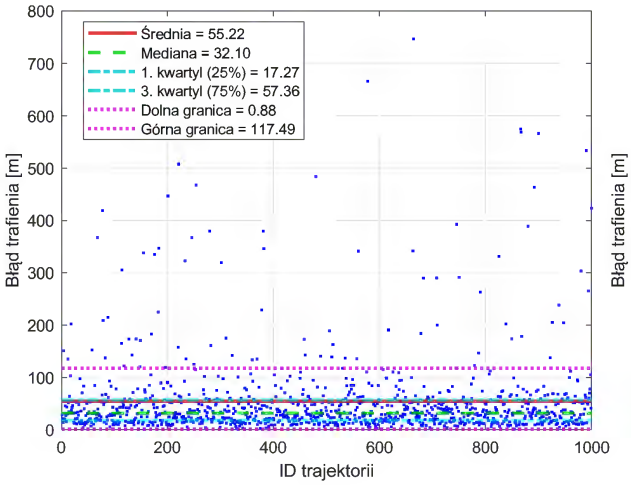
Rysunek 8.21: Ilość wykorzystanych silników korekcyjnych przy 96 dostępnych

Rysunek 8.21 przedstawia liczbę wykorzystanych silników korekcyjnych, która w obu przypadkach wyniosła średnio około 65 silników. Oba algorytmy charakteryzują się podobnym zużyciem analizując pozostałe parametry statystyczne. W większości lotów spora ilość silników pozostaje niewykorzystana.

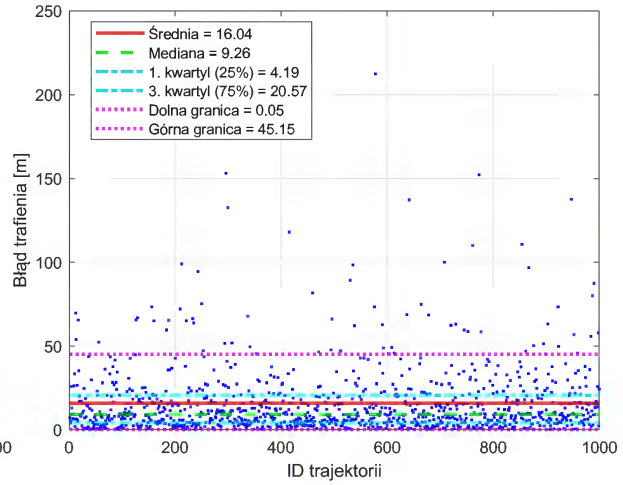


(a) Rozrzut PNG

(b) Rozrzut baza danych



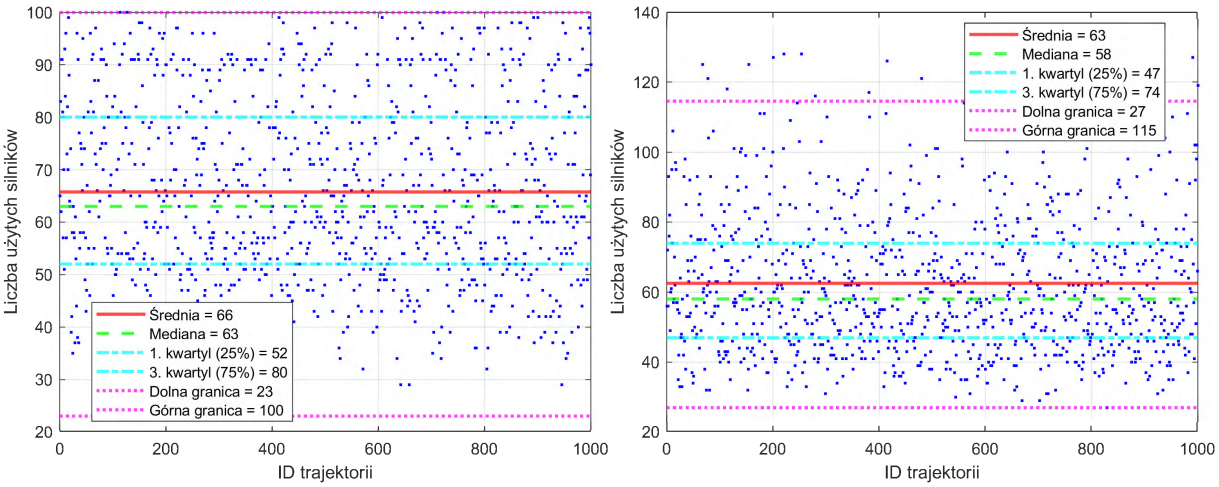
(c) Statystyka PNG



(d) Statystyka baza danych

Rysunek 8.22: Wyniki rozrzutu przy wykorzystaniu 128 silników korekcyjnych

Ostatni przypadek testowy wykorzystujący 128 silników korekcyjnych pokazano na Rysunku 8.22. W przypadku algorytmu PNG nie ma praktycznie żadnej różnicy w porównaniu do przypadku z 96 dostępnymi silnikami. Już w tamtym przypadku przy większości symulacji algorytm nie wykorzystywał wszystkich dostępnych silników. W przypadku algorytmu opartego o bazę danych jedyny wyraźnych zysk widoczny jest przy parametrze CEP_{99}, który zmalał o około 35%. Wyniki te oznaczają, że praktycznie wszystkie wystrzelone rakiety spadły w okręgu o promieniu poniżej 100 metrów co jest wynikiem bardzo dobrym.



(a) PNG (b) Baza danych

Rysunek 8.23: Ilość wykorzystanych silników korekcyjnych przy 128 dostępnych

Rysunek 8.23 przedstawia wykorzystanie silników korekcyjnych dla przypadku 128 dostępnych. Jak poprzednio większość symulacji nie wykorzystała wszystkich możliwych silników, a dla algorytmu PNG nie ma większej różnicy niż dla przypadku wcześniejszego (96 dostępnych). Pokazuje to, że wyposażenie rakiety w moduł składający się z większej liczby silników nie zapewni uzyskania lepszych rezultatów.

9 Podsumowanie

Celem pracy było opracowanie algorytmu sterowania pociskiem rakietowym typu ziemia-ziemia wyposażonym w gazodynamiczny moduł sterujący, składający się z zestawu silników korekcyjnych na stały materiał pędny, który zapewniłby trafienie cel naziemny o znanych współrzędnych z wymaganą dokładnością i wykorzystał w tym celu minimalną ilość energii, czyli minimalną ilość silników korekcyjnych. Ze względu na impulsowy charakter układu sterującego i możliwość manipulowania tylko kierunkiem i czasem działania siły sterującej, bez wpływu na jej wartość, nie liczyło się uzyskanie bezpośredniego trafienia w cel, lecz zmniejszenie dyspersji miejsc upadku. W celu rozwiązania postawionego zagadnienia skorzystano z metod optymalizacji gradientowej. Aby spełnić drugi z warunków optymalizacji, minimalizację energii, nie możliwe było zadanie jako zmiennych optymalizacyjnych bezpośrednio czasów i kierunków odpaleń poszczególnych silników korekcyjnych, gdyż ich niezbędna ilość nie była z góry znana. Dodatkowo, założeniem było wykorzystanie metod optymalizacji dedykowanych dla sygnałów ciągłych. Z tego powodu opracowano specjalną metodykę uzyskania rozwiązania, na którą składało się opracowanie modelu symulacyjnego rakiety, procedury optymalizacji wykorzystującej uproszczony, ciągły, układ sterujący, a następnie dyskretyzacji jej wyników oraz opracowanie algorytmu sterowania bazującego na bazie danych składających się z optymalnych rozwiązań.

Model matematyczny rakiety pozwalał na symulację lotu obiektu o sześciu stopniach swobody i zmiennych parametrach masowych. Opisano dokładnie wykorzystywane równania ruchu rakiety uwzględniające również więzy występujące podczas ruchu po wyrzutni prowadnicowej. Uwzględniono model środowiska składający się z zależności opisujących atmosferę standardową, dodatkowo uwzględniając lokalne warunki panujące w miejscu startu, takie jak temperatura, ciśnienie oraz wilgotność powietrza. Charakterystyki bezwładnościowe rakiety podlegały zmianom w funkcji działania silnika głównego oraz silników korekcyjnych. Uwzględniano obciążenia działające na obiekt od zespołu napędowego, grawitacji, układu sterowania oraz aerodynamiki. Wykorzystano także model czujników inercjalnych, akcelerometrów oraz giroskopów, a także uproszczony model układu nawigacji inercjalnej INS. Model numeryczny rakiety opracowano najpierw w środowisku MATLAB/Simulink, a następnie przepisano go do języka C++ w celu przeprowadzenia obliczeń optymalizacyjnych.

Opracowany model wymagał weryfikacji oraz walidacji działania. Niezbędne w procesie weryfikacji dane wymagane w modelu uzupełniono korzystając z opracowanej na PW konstrukcji RPT ORION. Zmierzono w części charakterystyki bezwładnościowe, uzupełniając brakujące dane wartościami z przygotowanego modelu CAD, zmierzono ciąg silnika głównego oraz silników korekcyjnych, a także oszacowano niezbędne współczynniki aerodynamiczne korzystając z pół-empirycznych zależności bazujących na geometrii rakiety. Po udanej weryfikacji modelu matematycznego korzystając z danych z testów lotnych

kilku egzemplarzy rakiety ORION przeprowadzono jego walidację. W ramach projektu badawczego opracowano także algorytm sterujący bazujący na klasycznych metodach naprowadzania, takich jak nawigacja proporcjonalna. Dysponując zapisaną telemetrią z testów poligonowych dokonano identyfikacji niektórych współczynników aerodynamicznych dostrajając model do wyników eksperymentu. Ze względu na ograniczone środki finansowe dostępne w projekcie możliwe było przeprowadzenie tylko kilku strzałów docelowej konstrukcji rakiety, gdzie skupiono się głównie na sprawdzeniu samej stateczności i sterowności opracowanej rakiety oraz poprawności współdziałania wszystkich podsystemów. Uzyskane wyniki potwierdzały, że opracowany model symulacyjny z dobrą dokładnością potrafi odwzorować rzeczywisty lot rakiety. Niska ilość testów poligonowych wymagała, aby stwierdzić efektywność układu sterowania, przeprowadzenia obliczeń symulacyjnych. Rozrzuty miejsc upadku przy działaniu układu sterowania wykorzystującego algorytm oparty o nawigację proporcjonalną wyznaczono wykorzystując metodę Monte Carlo. Otrzymane wyniki służyły za punkt odniesienia dla algorytmu sterowania będącego wynikiem niniejszej pracy.

Procedurę optymalizacji rozpoczęto od uproszczenia układu sterowania. Dyskretne impulsy odpaleń silników korekcyjnych zamieniono na dwa ciągłe parametry sterujące działające w dwóch prostopadłych płaszczyznach rakiety. Do optymalizacji wykorzystano dwie metody: bezpośrednią, która traktowała problem jak zwykłe zagadnienie optymalizacji nieliniowej oraz pośrednią, która zamieniała zagadnienie na problem sterowania optymalnego. Funkcję kosztu zadano zgodnie z celem minimalizacji, a jej gradient wyznaczano w zależności od metody, bezpośrednio korzystając z automatycznego różniczkowania funkcji kosztu lub w metodzie pośredniej wyprowadzając odpowiednią zależność matematyczną korzystając z Zasady Maksimum Pontryagina. Do obliczeń wykorzystywano otwarte oprogramowanie IPOPT oraz pakiet ADOL-C. Opisano dokładnie całą procedurę optymalizacji oraz przedstawiono wyniki działania obu metod. Pod względem dokładności obliczeń obie metody potrafiły znaleźć rozwiązanie z bardzo dobrą dokładnością. Wynikowy błąd upadku był rzędu pojedynczych centymetrów. Metoda pośrednia okazała się kilkukrotnie szybsza, z tego względu w dalszych obliczenia tylko ona była wykorzystywana.

Dysponując wynikami optymalizacji w postaci przebiegów czasowych dwóch parametrów sterujących, należało wrócić do oryginalnego układu sterującego, to znaczy zamienić je na czasy i kierunki odpaleń poszczególnych silników korekcyjnych. Przedstawiono opracowaną metodę konwersji, a także jej wyniki. Błąd upadku po dyskretyzacji wzrastał do wartości kilku metrów. Ze względu na to, że optymalizacja dostarczała informację o niezbędnej ilości energii, a więc wymaganej liczbie silników korekcyjnych, dodatkowo przeprowadzano optymalizację uzyskanych w procedurze dyskretyzacji czasów i kierunków odpaleń. W ten sposób udało się zmniejszyć błąd upadku rakiety korzystającej z dyskretnego układu sterowania o średnio pół metra. Oznacza to, że opracowana proce-

dura dyskretyzacji jest dokładna.

Aby zrealizować cel pracy opracowano algorytm sterowania bazujący na optymalnych wynikach. Przeprowadzając obliczenia optymalizacyjne oraz dyskretyzację dla różnych warunków początkowych opracowano bazę danych rozwiązań optymalnych składającą się z ilości, czasów oraz kierunków odpaleń silników korekcyjnych oraz odpowiadającej im trajektorii referencyjnej służącej do wyboru odpowiedniego rozwiązania w trakcie lotu. Lot rakiety z docelowym algorytmem sterowania podzielono na kilka faz. W pierwszej fazie, lotu wznoszącego, algorytm miał na celu znalezienie w bazie danych trajektorii najbardziej zbliżonej do aktualnej, którą rakieta podąża, oraz wczytaniu odpowiadającej jej sekwencji odpaleń silników korekcyjnych. Następnie, po osiągnięciu apogeum rakieta sterowana była na jej podstawie w pętli otwartej. Aby zwiększyć dokładność sterowania w końcowej fazie lotu przełączano się na algorytm nawigacji proporcjonalnej. Sprawdzono zależność uzyskanych rozrzutów miejsc upadku od momentu przełączenia algorytmu, ilości wczytanych trajektorii referencyjnych oraz dostępnej ilości silników korekcyjnych. Przy przełączeniu algorytmu w okolicach połowy lotu sterowanego, wykorzystaniu wszystkich obliczonych trajektorii referencyjnych oraz 32 dostępnych silników korekcyjnych uzyskano około 10% redukcję rozrzutu charakteryzowaną przez parametry CEP_{50}, CEP_{83} oraz CEP_{99}. Wpływ wielkości bazy danych na wartość tych parametrów uznano za znikomy. Badając wpływ większej możliwej do wykorzystania ilości silników korekcyjnych przedstawiono wyższość opracowanego algorytmu nad rozwiązaniami klasycznymi w każdym przypadku uzyskując zysk niemal 70% dla parametru CEP_{50} oraz od około 35% do 70% dla pozostałych.

W przyszłości opracowany algorytm należałoby uzupełnić o wpływ danych dostarczanych przez czujniki pokładowe oraz algorytm nawigacji inercjalnej, a także przeprowadzić testy SIL, PIL oraz HIL. Wymagane byłoby sprawdzenie ilości niezbędnej pamięci operacyjnej oraz prędkości wykonywania kodu na docelowym komputerze pokładowym rakiety. Jako system odniesienia wykorzystywana była konstrukcja RPT ORION. Aby opracowany algorytm mógł być stosowanych na rakietach kalibru 122 mm będących na wyposażeniu Sił Zbrojnych RP, algorytm ten wraz z całym gazodynamicznych modułem sterującym należałoby przetestować w warunkach bardziej zbliżonych do osiągów tych rakiet, które poruszają się z prędkościami dużo większymi oraz na dalszych zasięgach. Niezbędne jest także oszacowanie efektywności silników korekcyjnych przy sterowaniu w prędkościach naddźwiękowych. Mimo posiadania dokładnego modelu matematycznego rakiety niezbędne jest wykonanie wielu testów lotnych, aby potwierdzić uzyskane wyniki rozrzutów.

Bazując na uzyskanych wynikach stwierdzono, że postawiona teza badawcza: **Wykorzystanie metod optymalizacji do wyznaczania parametrów sterowania rakietą oraz sterowanie z wykorzystaniem bazy danych pozwoli skuteczniej i efektywniej naprowadzać rakietę na cel stacjonarny** została potwierdzona. Z suk-

cesem zrealizowano postawiony plan badawczy, opracowano niezbędne oprogramowania i zdaniem autora znacząco przyczyniono się do rozwoju kompetencji w zakresie techniki rakietowej na PW.

Bibliografia

[1] G. M. Siouris, *Missile Guidance and Control Systems.* Springer, 2004.

[2] E. L. Fleeman, *Tactical Missile Design.* AIAA Education Series, 2001.

[3] E. L. Fleeman, *Missile Design Guide.* AIAA Education Series, 2022.

[4] F. Mingireanu, L. Georgescu, G. Murariu, and I. Mocanu, "Trajectory modeling of grad rocket with low-cost terminal guidance upgrade coupled to range increase through step-like thrust-curves," *Romanian Journal of Physics*, vol. 59, pp. 369–381, 01 2014.

[5] R. Głębocki and M. Żugaj, "Model of Gasodynamic Control System for Guided Bombs," *Journal of Theoretical and Applied Mechanics*, vol. 48, no. 1, pp. 27–44, 2010.

[6] R. Głębocki, "Guidance impulse algorithms for air bomb control," *Bulletin of the Polish Academy of Sciences, Technical Sciences*, vol. 60, no. 4, pp. 825–833, 2012.

[7] B.-Y. Min, J.-W. Lee, Y.-H. Byun, J.-S. Hyun, and S. Kim, "Numerical investigation of the lateral jet effect on the aerodynamic characteristics of the missile: Part i. jet flow condition effect," *Journal of the Korean Society for Aeronautical & Space Sciences*, vol. 32, 10 2004.

[8] J.-W. Lee, B.-Y. Min, Y.-H. Byun, and Y. H. Yu, "Computational Investigation and Design Optimization of a Lateral-Jet-Controlled Missile," *Journal of Aircraft*, vol. 43, no. 5, pp. 1292–1300, 2006.

[9] T. Jitpraphai, B. Burchett, and M. Costello, "A Comparison of Different Guidance Schemes for a Direct Fire Rocket With a Pulse Jet Control Mechanism," tech. rep., Army Research Laboratory, USA, 2002.

[10] S. Mandić, "Dispersion reduction of artillery rockets guided by flight path steering method," *The Aeronautical Journal*, vol. 120, no. 1225, pp. 435—-456, 2016.

[11] S. Gupta, S. Saxena, A. Singhal, and A. Ghosh, "Trajectory correction flight control system using pulsejet on an artillery rocket," *Defence Science Journal*, vol. 58, 01 2008.

[12] T. Jitpraphai and M. Costello, "Dispersion Reduction of a Direct Fire Rocket Using Lateral Pulse Jets," *Journal of Spacecraft and Rockets*, vol. 38, no. 6, pp. 929–936, 2001.

[13] R. Li, D. Li, and J. Fan, "Correction Strategy of Mortars with Trajectory Correction Fuze Based on Image Sensor," *Sensors*, vol. 19, no. 5, 2019.

[14] M. Pavic, B. Pavkovic, S. Mandic, S. Zivkovic, and D. Cuk, "Pulse-frequency modulated guidance laws for a mortar missile with a pulse jet control mechanism," *Aeronautical Journal*, vol. 119, pp. 389–405, 03 2015.

[15] R. Głębocki and M. Jacewicz, "Parametric Study of Guidance of a 160-mm Projectile Steered with Lateral Thrusters," *Aerospace*, vol. 7, no. 61, pp. 1–27, 2020.

[16] R. Ożóg, M. Jacewicz, and R. Głębocki, "Modified Trajectory Tracking Guidance For Artillery Rocket," *Journal of Theoretical and Applied Mechanics*, vol. 58, no. 3, pp. 611–622, 2020.

[17] X. Sun, M. Gao, X. Zhou, J. Lv, F. Tian, and Z. Qiao, "Guidance Simulation and Experimental Verification of Trajectory Correction Mortar Projectile," *IEEE Access*, vol. 9, pp. 15609–15622, 2021.

[18] Z. Shi, W. Ma, and Y. Zhang, "A Novel Control System Design Method for Missile with Lateral Jet and Aerodynamic Surfaces," in *2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 858–861, 2013.

[19] H. J. Pesch, "A practical guide to the solution of real-life optimal control problems," *Control and Cybernetics*, vol. 23, no. 1/2, pp. 7–60, 1994.

[20] J.-B. Caillau, R. Ferretti, E. Trélat, and H. Zidani, "Chapter 15 - An algorithmic guide for finite-dimensional optimal control problems," in *Numerical Control: Part B* (E. Trélat and E. Zuazua, eds.), vol. 24 of *Handbook of Numerical Analysis*, pp. 559–626, Elsevier, 2023.

[21] A. Rao, "A Survey of Numerical Methods for Optimal Control," *Advances in the Astronautical Sciences*, vol. 135, pp. 1–32, 01 2010.

[22] J.-B. Caillau, R. Ferretti, E. Trélat, and H. Zidani, "Numerics for finite-dimensional optimal control problems," *hal-03707475*, 2022.

[23] D. Devadze and V. Beridze, "Methods of numerical solution of optimal control problems based on the pontryagin maximum principle," *Journal of Mathematical Sciences*, vol. 206, 04 2015.

[24] J. Henriques, J. Lemos, L. Eça, J. Valério, L. Gato, and A. Falcão, "A Discontinuous Galerkin Method for optimal and sub-optimal control applied to an oscillating water column wave energy converter," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15670–15677, 2017. 20th IFAC World Congress.

[25] R. Bonalli, B. Hérissé, and E. Trélat, "Optimal control of endoatmospheric launch vehicle systems: Geometric and computational issues," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2418–2433, 2020.

[26] J. F. Bonnans, "The shooting approach to optimal control problems," in *IFAC International Workshop on Adaptation and Learning in Control and Signal Processing*, pp. 281–292, 2013.

[27] A. Trent, R. Venkataraman, and D. Doman, "Trajectory Generation Using A Modified Simple Shooting Method," in *2004 IEEE Aerospace Conference Proceedings*, pp. 2723–2729, 2004.

[28] G. Fraser-Andrews, "A Multiple Shooting Technique for Optimal Control," *Journal of Optimization Theory and Applications*, vol. 102, no. 2, pp. 299–313, 1999.

[29] H. J. Oberle, "Numerical Solution of Minimax Optimal Control Problems by Multiple Shooting Technique," *Journal of Optimization Theory and Applications*, vol. 50, no. 2, pp. 331–357, 1986.

[30] U. Ali and Y. Wardi, "Multiple shooting technique for optimal control problems with application to power aware networks**research supported in part by nsf under grant cns-1239225.," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 286–290, 2015. Analysis and Design of Hybrid Systems ADHS.

[31] W. Hager, "Runge-kutta methods in optimal control and the transformed adjoint system," *Numerische Mathematik*, vol. 87, 02 2001.

[32] A. Karbowski, "Shooting Methods to Solve Optimal Control Problems with State and Mixed Control-State Constraints," in *Challenges in Automation, Robotics and Measurement Techniques*, (Cham), pp. 189–205, Springer International Publishing, 2016.

[33] W. Grimm and A. Markl, "Adjoint Estimation from a Direct Multiple Shooting Method," *Journal of Optimization Theory and Applications*, vol. 92, pp. 263–283, 1997.

[34] H. Diedam and S. Sager, "Global optimal control with the direct multiple shooting method," *Optimal Control Applications and Methods*, vol. 39, no. 2, pp. 449–470, 2018.

[35] P. J. Enright and B. A. Conway, "Discrete approximations to optimal trajectories using direct transcription and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 4, pp. 994–1002, 1992.

[36] P. Drąg and K. Styczeń, *Direct Shooting Method for Optimal Control of the Highly Nonlinear Differential-Algebraic Systems*, pp. 75–90. Springer International Publishing, 2016.

[37] M. Gerdts, "Direct Shooting Method for the Numerical Solution of Higher-Index DAE Optimal Control Problems," *Journal of Optimization Theory and Applications*, vol. 117, pp. 267–294, 2003.

[38] M. Jeon, "Parallel optimal control with multiple shooting, constraints aggregation and adjoint methods," *J Appl Math Comput*, vol. 19, pp. 215–229, 03 2005.

[39] Y. Meng, H. Zhang, and Y. Gao, "Low-thrust minimum-fuel trajectory optimization using multiple shooting augmented by analytical derivatives," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 3, pp. 662–677, 2019.

[40] G. T. Huntington and A. V. Rao, "Comparison of Global and Local Collocation Methods for Optimal Control," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 2, pp. 432–436, 2008.

[41] F. Fahroo and I. M. Ross, "A Spectral Patching Method for Direct Trajectory Optimization," *The Journal of the Astronautical Sciences*, vol. 48, no. 2, pp. 269–286, 2000.

[42] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order gauss-lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 592–599, 1996.

[43] P. Williams, "A Gauss-Lobatto quadrature method for solving optimal control problems," in *Proceedings of the 7th Biennial Engineering Mathematics and Applications Conference, EMAC-2005*, vol. 47, pp. 101–115, 2006.

[44] I. M. Ross and F. Fahroo, "Pseudospectral Knotting Methods for Solving Optimal Control Problems," *Journal of Guidance, Control and Dynamics*, vol. 27, no. 3, pp. 397–405, 2004.

[45] I. M. Ross and F. Fahroo, "Legendre Pseudospectral Approximations of Optimal Control Problems," in *New Trends in Nonlinear Dynamics and Control, LNCIS 295*, pp. 327–342, 2003.

[46] B. T. Burchett, "A Gauss Pseudospectral Collocation for Rapid Trajectory Prediction and Guidance," in *AIAA Atmospheric Flight Mechanics Conference*, pp. 1–20, 2017.

[47] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006.

[48] G. N. Elnagar and M. Kazemi, "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, vol. 11, pp. 195––217, 1998.

[49] F. Fahroo and I. Ross, "Direct trajectory optimization by a chebyshev pseudospectral method," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, vol. 6, pp. 3860–3864 vol.6, 2000.

[50] J. Vlassenbroeck and R. Van Dooren, "A Chebyshev Technique for Solving Nonlinear Optimal Control Problems," *IEEE Transactions on Automatic Control*, vol. 33, no. 4, pp. 333–340, 1988.

[51] J. Vlassenbroeck, "A Chebyshev Polynomial Method for Optimal Control with State Constraints," *Automatica*, vol. 24, no. 4, pp. 499–506, 1988.

[52] G. N. Elnagar and M. A. Kazemi, "Pseudospectral Legendre-based optimal computation of nonlinear constrained variational problems," *Journal of Computational and Applied Mathematics*, vol. 88, no. 2, pp. 363–375, 1998.

[53] G. Elnagar, M. Kazemi, and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1793–1796, 1995.

[54] P. Williams, "Jacobi Pseudospectral Method for Solving Optimal Control Problems," *Journal of Guidance*, vol. 27, no. 2, pp. 293–297, 2003.

[55] F. Fahroo and I. M. Ross, "Pseudospectral Methods for Infinite-Horizon Optimal Control Problems," *Journal of Guidance, Control and Dynamics*, vol. 31, no. 4, pp. 1–10, 2008.

[56] D. Garg, M. Patterson, C. Francolin, C. Darby, G. Huntington, W. Hager, and A. Rao, "Direct Trajectory Optimization and Costate Estimation of General Optimal Control Problems Using a Radau Pseudospectral Method," *Computational Optimization and Applications*, vol. 49, pp. 335–358, 06 2011.

[57] G. Fraser-Andrews, "Shooting Method for the Numerical Solution of Optimal Control Problems with Bounded State Variables," *Journal of Optimization Theory and Applications*, vol. 89, no. 2, pp. 351–372, 1996.

[58] M. T. Davis, M. J. Robbins, and B. J. Lunday, "Approximate dynamic programming for missile defense interceptor fire control," *European Journal of Operational Research*, vol. 259, no. 3, pp. 873–886, 2017.

[59] D. Ghose, U. Prasad, and K. Guruprasad, "Missile battery placement for air defense: A dynamic programming approach," *Applied Mathematical Modelling*, vol. 17, no. 9, pp. 450–458, 1993.

[60] D. Bertsekas, M. Homer, D. Logan, S. Patek, and N. Sandell, "Missile defense and interceptor allocation by neuro-dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 30, no. 1, pp. 42–51, 2000.

[61] J. Sun and C. Liu, "Finite-horizon differential games for missile–target interception system using adaptive dynamic programming with input constraints," *International Journal of Systems Science*, vol. 49, pp. 1–20, 11 2017.

[62] G. Xu, K. Sun, and H. Liu, "Optimal control of missile interception process based on adaptive dynamic programming," in *2019 Chinese Automation Congress (CAC)*, pp. 1256–1260, 2019.

[63] E. Cristiani and P. Martinon, "Initialization of the shooting method via the hamilton-jacobi-bellman approach," *Journal of Optimization Theory and Applications*, vol. 146, 08 2010.

[64] J. Lin, X. Li, S. C. Hoe, and Z. Yan, "A generalized finite difference method for solving hamilton–jacobi–bellman equations in optimal investment," *Mathematics*, vol. 11, no. 10, 2023.

[65] J. Ma and J. Ma, "Finite Difference Methods for the Hamilton–Jacobi–Bellman Equations Arising in Regime Switching Utility Maximization," *Journal of Scientific Computing*, vol. 85, no. 55, 2020.

[66] O. Bokanowsky, E. Bourgeois, A. Désilles, and H. Zidani, "Hjb approach for a multi-boost launcher trajectory optimization problem," *IFAC-PapersOnLine*, vol. 49, no. 17, pp. 456–461, 2016. 20th IFAC Symposium on Automatic Control in AerospaceACA 2016.

[67] L. Yan, X. Chang, N. Wang, L. Zhang, W. Liu, and X. Deng, "Aerodynamic Identification and Control Law Design of a Missile Using Machine Learning," *AIAA Journal*, vol. 61, no. 7, pp. 2998–3018, 2023.

[68] W. Li, Y. Zhu, and D. Zhao, "Missile guidance with assisted deep reinforcement learning for head-on interception of maneuvering target," *Complex & Intelligent Systems*, vol. 8, pp. 1205—-1216, 2022.

[69] W. S. Kovalik, L. Zhai, and K. G. Vamvoudakis, "Model-free perception-based control via q-learning with an application to heat-seeking missile guidance," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1142–1147, 2021.

[70] R. Xie, X. Jin, and Q. Zhao, "Rapid Bootstrapping of Deep Reinforcement Learning with Curriculum and Imitation Strategies for Missile Guidance," *International Journal of Aeronautical and Space Sciences*, 2025.

[71] Y. Wang, P. Kai, H. Chen, K. Zhao, J. Guirao, and D. Cao, "Maneuvering penetration method for missile under unknown interception strategies based on transfer deep reinforcement learning," *Fractals*, 05 2025.

[72] C. Ferro, M. Cafaro, and P. Maggiore, "Optimizing Solid Rocket Missile Trajectories: A Hybrid Approach Using an Evolutionary Algorithm and Machine Learning," *Aerospace*, vol. 11, no. 912, pp. 1–18, 2024.

[73] D. Hong, M. Kim, and S. Park, "Study on Reinforcement Learning-Based Missile Guidance Law," *Applied Sciences*, vol. 10, no. 6567, pp. 1–13, 2020.

[74] A. L. Nash and B. T. Burchett, "Euler-Lagrange Optimal Control of Indirect Fire Symmetric Projectiles," in *AIAA Atmospheric Flight Mechanics Conference*, pp. 1–18, 2016.

[75] B. T. Burchett, "Predictive Optimal Pulse-jet Control for Symmetric Projectiles," in *AIAA Atmospheric Flight Mechanics Conference*, pp. 1–12, 2014.

[76] B. Burchett and M. Costello, "Model predictive lateral pulse jet control of an atmospheric rocket," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 5, pp. 860–867, 2002.

[77] H. Fenghua, M. Kemao, Y. Baoqing, and Y. Yu, "Model predictive control for a missile with blended lateral jets and aerodynamic fins," in *2008 27th Chinese Control Conference*, pp. 430–434, 2008.

[78] Y. Zhang, B. Yang, X. Huo, and S. Chen, "Firing logic design for acs of missile with pulse jets," in *2016 Chinese Control and Decision Conference (CCDC)*, pp. 5596–5601, 2016.

[79] E. J. Ohlmeyer, "Control of Terminal Engagement Geometry Using Generalized Vector Explicit Guidance," in *Proceedings of the American Control Conference*, pp. 396–401, 2003.

[80] C.-K. Ryoo, H. Cho, and M. Tahk, "Optimal Guidance Laws with Terminal Impact Angle Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, pp. 724–732, 2005.

[81] H. Fenghua, M. Kemao, and Y. Yu, "Firing logic optimization design of lateral jets in missile attitude control systems," in *2008 IEEE International Conference on Control Applications*, pp. 936–941, 2008.

[82] M. Gao, Z. Yongwei, and S. Yang, "Firing control optimization of impulse thrusters for trajectory correction projectiles," *International Journal of Aerospace Engineering*, vol. 2015, pp. 1–11, 04 2015.

[83] C. T. Lawrence and A. L. Tits, "A computationally efficient feasible sequential quadratic programming algorithm," *SIAM Journal on Optimization*, vol. 11, no. 4, pp. 1092–1118, 2001.

[84] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 979–1006, 2002.

[85] J. Nocedal, F. Öztoprak, and R. A. Waltz, "An interior point method for nonlinear programming with infeasibility detection capabilities," *Optimization Methods Software*, vol. 29, no. 4, p. 837–854, 2014.

[86] J. Nocedal, A. Wächter, and R. A. Waltz, "Adaptive Barrier Update Strategies for Nonlinear Interior Methods," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1674–1693, 2009.

[87] C. Lemarechal, "A view of line-searches," in *Optimization and Optimal Control*, pp. 59–78, Springer Berlin Heidelberg, 1981.

[88] J. Moré and D. Thuente, "On line search algorithms with guaranteed sufficient decrease," *ACM Trans. Math. Softw.*, vol. 20, pp. 286–307, 09 1994.

[89] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*. MPS-SIAM Series on Optimization, SIAM, 2000.

[90] G. A. Schultz, R. B. Schnabel, and R. H. Byrd, "A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties," *SIAM Journal on Numerical Analysis*, vol. 22, pp. 47—67, 1985.

[91] R. Fletcher, S. Leyffer, and C. Shen, "Nonmonotone Filter Method for Nonlinear Optimization," *Computational Optimization and Applications*, vol. 52, pp. 1–24, 11 2009.

[92] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: Local convergence," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 32–48, 2005.

[93] N. I. M. Gould and P. L. Toint, "SQP Methods for Large-Scale Nonlinear Programming," in *System Modelling and Optimization* (M. J. D. Powell and S. Scholtes, eds.), (Boston, MA), pp. 149–178, Springer US, 2000.

[94] O. von Stryk, "Numerical Solution of Optimal Control Problems by Direct Collocation," *International Series of Numerical Mathematics*, vol. 111, pp. 129–143, 1993.

[95] H. Jiang, Y. Mei, J. Yu, J. Li, and X. Chen, "Optimization design of the trajectory about gun-launched missile based on sqp," in *Proceedings of the 33rd Chinese Control Conference*, pp. 7555–7560, 2014.

[96] F. Wang and C. Dong, "Fast Intercept Trajectory Optimization for Multi-stage Air Defense Missile Using Hybrid Algorithm," *Procedia Engineering*, vol. 67, pp. 447–456, 2013.

[97] S. Lee, H. Lee, Y. Kim, J. Kim, and W. Choi, "GPU-Accelerated PD-IPM for Real-Time Model Predictive Control in Integrated Missile Guidance and Control Systems," *Sensors*, vol. 22, no. 12, 2022.

[98] P. Lin, J. Hou, and S. Sun, "Velocity-Constrained Trajectory Planning of Anti-ship Missile," in *2015 34TH CHINESE CONTROL CONFERENCE (CCC)*, Chinese Control Conference, pp. 5451–5454, 2015.

[99] F. Xiao, Z. Jian, W. Zhang, T. Wang, D. Wang, and M. Chen, "An integrated data model and its application for missile design based on hyper-graph," in *2009 International Conference on Computers & Industrial Engineering*, pp. 1526–1531, 2009.

[100] J. Chang, X. Hu, G. Zhang, and J. Xiao, "Modeling of Missile Rudder Based on Modelica," in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 2490–2495, 2019.

[101] S. Tamblyn, J. Henry, and E. King, "A model-based design and testing approach for Orion GN&C flight software development," in *2010 IEEE Aerospace Conference*, pp. 1–12, 2010.

[102] M. Jackson and J. Henry, "Orion GN&C Model Based Development: Experience And Lessons Learned," in *AIAA Guidance, Navigation, and Control Conference*, 2012.

[103] X. Sun, M. Gao, X. Zhou, J. Lv, F. Tian, and Z. Qiao, "Guidance Simulation and Experimental Verification of Trajectory Correction Mortar Projectile," *IEEE Access*, vol. 9, pp. 15609–15622, 2021.

[104] W. Yang, X. Zhang, and H. He, "Unsteady numerical simulation of missile trajectories with attitude control law," in *2017 8th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, pp. 576–580, 2017.

[105] M. Sharma and A. K. Gupta, "A Study And Mathematical Modeling For Autopilot Guidance And Control In Homing Guided Missiles," *Advances and Applications in Mathematical Sciences*, vol. 19, no. 5, pp. 341–352, 2020.

[106] X. Guo, S. Zhao, P. He, and J. Xie, "Simulation Analysis and Improved Design of the Control System of a Certain Missile," in *Communications, Signal Processing, and Systems* (Q. Liang, J. Mu, W. Wang, and B. Zhang, eds.), (Singapore), pp. 899–907, Springer Singapore, 2018.

[107] Y.-c. Chen, X.-b. Gao, M. Gao, and D. Fang, "An Optimal Missile Autopilot Design Model," *International Journal of Enterprise Information Systems*, vol. 14, no. 1, pp. 104–110, 2018.

[108] J. Yang, X. Wang, S. Baldi, S. Singh, and S. Farì, "A software-in-the-loop implementation of adaptive formation control for fixed-wing UAVs," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 5, pp. 1230–1239, 2019.

[109] A. Tullu, S. Jung, S. Lee, and S. Ko, "Effects of Model-Specific Parameters on the Development of Custom Module in PX4 Autopilot Software-in-the-Loop," *International Journal of Aerospace Engineering*, vol. 2025, no. 1, pp. 1–18, 2025.

[110] E. H. Kapeel, M. A. H. Abozied, A. M. Kamel, and H. Hendy, "Evaluation of Missile Terminal Guidance Using Software in Loop (SIL)," in *2020 12th International Conference on Electrical Engineering (ICEENG)*, pp. 389–395, 2020.

[111] A. Kaviyarasu, A. Saravanakumar, and K. Elumalai, "Software in the loop simulation of formation flying of multi rotor uav," in *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 336–340, 2019.

[112] S. Jain, O. Halbe, K. Sachan, V. Thenignaledathil, S. C, and A. Nayak, "Software-In-The-Loop Simulation for Low-Speed, High-Altitude Platform," in *2024 IEEE Space, Aerospace and Defence Conference (SPACE)*, pp. 930–934, 07 2024.

[113] D. Kumar and G. Kumaresan, "Processor-in-Loop Simulation for Formation Flying of Multiple Unmanned MAVs," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 688–693, 2016. 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016.

[114] B. Elsayed, A. Ouda, Y. Zakaria, E. Gamal, and G. Elnashar, "Autopilot Implementation Synthesis via Processor-in-Loop Test," in *International Conference on Aerospace Sciences and Aviation Technology*, 2017.

[115] M. Mammarella and E. Capello, "Tube-Based Robust MPC Processor-in-the-Loop Validation for Fixed-Wing UAVs.," *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 239—-258, 2020.

[116] E. Tramacere, S. Luciani, S. Feraco, A. Bonfitto, and N. Amati, "Processor-in-the-loop architecture design and experimental validation for an autonomous racing vehicle," *Applied Sciences*, vol. 11, no. 16, 2021.

[117] K. Saulnier, D. Pérez, R. Huang, D. Gallardo, G. Tilton, and R. Bevilacqua, "A six-degree-of-freedom hardware-in-the-loop simulator for small spacecraft," *Acta Astronautica*, vol. 105, no. 2, pp. 444–462, 2014.

[118] S. B. Mobley and J. P. Gareri, "Hardware-in-the-loop simulation (HWIL) facility for development, test, and evaluation of multispectral missile systems: update," in *Technologies for Synthetic Environments: Hardware-in-the-Loop Testing V* (R. L. M. Jr., ed.), vol. 4027, pp. 11 – 21, International Society for Optics and Photonics, SPIE, 2000.

[119] J. A. Ray, G. A. Larson, and J. E. Terry, "Hardware-in-the-loop support of the Longbow/HELLFIRE modular missile systems preplanned product improvement program," in *Technologies for Synthetic Environments: Hardware-in-the-Loop Testing VI* (R. L. M. Jr., ed.), vol. 4366, pp. 82–90, 2001.

[120] J. Kiesbye, D. Messmann, M. Preisinger, G. Reina, D. Nagy, F. Schummer, M. Mostad, T. Kale, and M. Langer, "Hardware-in-the-loop and software-in-the-loop testing of the move-ii cubesat," *Aerospace*, vol. 6, no. 12, 2019.

[121] National Imagery and Mapping Agency (NIMA), "Department of Defense World Geodetic System 1984: Its Definition, and Relationship with Local Geodetic Systems, TR8350.2, Third Ed.," tech. rep., Department of Defence, Washington, DC, 1997.

[122] J. Leyko, *Mechanika Ogólna: Dynamika Tom 2*. Warszawa: Wydawnictwo Naukowe PWN, 2010.

[123] M. R. Napolitano, *Aircraft Dynamics, From Modeling to Simulation*. Hoboken: John Wiley & Sons, 2012.

[124] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*. American Institute of Aeronautics and Astronautics (AIAA), 2007.

[125] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation, 3rd Edition*. John Wiley & Sons, 2016.

[126] W. T. Thomson, "Equations of motion for the variable mass system.," *AIAA Journal*, vol. 4, no. 4, pp. 766–768, 1966.

[127] T. C. Mao and F. O. Eke, "Attitude Dynamics of a Torque-Free Variable Mass Cylindrical Body," *The Journal of the Astronautical Sciences*, vol. 48, no. 4, pp. 435–448, 2000.

[128] H. Irschik, *Dynamics of Mechanical Systems with Variable Mass*. Springer, 2014.

[129] Military Handbook, "Missile Flight Simulation, Part One, Surface-to-Air Missiles," tech. rep., Department of Defence, USA, 1995.

[130] D. Allerton, *Principles of Flight Simulation*. John Wiley & Sons, 2009.

[131] R. Gutowski, *Mechanika Analityczna*. Warszawa: Państwowe Wydawnictwo Naukowe, 1971.

[132] D. J. Diston, *Computational Modelling and Simulation of Aircraft and the Environment, Volume 1: Platform Kinematics and Synthetic Environment*. United Kingdom: John Wiley & Sons, 2009.

[133] Arrow Tech, "PRODAS V3 Technical Manual," tech. rep., Arrow Tech, Vermont, USA, 2010.

[134] M. Jacewicz, P. Lichota, D. Miedziński, and R. Głębocki, "Study of model uncertainties influence on the impact point dispersion for a gasodynamicaly controlled projectile," *Sensors*, vol. 22, no. 9, 2022.

[135] Analog Devices, "ADIS16467: Precision MEMS IMU Module Data Sheet (Rev.C)," 2017.

[136] M. Jacewicz, D. Miedziński, G. Chmaj, and R. Głębocki, "Model-Based Development of Autopilot for a Gasodynamically Controlled High-Speed Unmanned Aerial Vehicle," *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 19, p. 8–25, Jun. 2025.

[137] C.-D. Yang and C.-C. Yang, "Analytical solution of generalized 3D proportional navigation," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 4, pp. 3974–3979, 1995.

[138] F. P. Adler, "Missile Guidance by Three-Dimensional Proportional Navigation," *Journal of Applied Physics*, vol. 27, pp. 500–507, 05 1956.

[139] I. Moran and D. T. Altilar, "Three Plane Approach for 3D True Proportional Navigation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 8, 2005.

[140] R. Ożóg, M. Jacewicz, and R. Głębocki, "Modified Trajectory Tracking Guidance for Artilery Rocket," *Journal of Theoretical and Applied Mechanics*, vol. 58, no. 3, pp. 611–622, 2020.

[141] M. Jacewicz and R. Głębocki, "Parametric Study of Guidance of a 160-mm Projectile Steered with Lateral Thrusters," *aerospace*, vol. 7, no. 61, pp. 1–27, 2020.

[142] R. V. Jategaonkar, *Flight Vehicle System Identification: A Time Domain Methodology*. Institute of Flight Systems - German Aerospace Center, AIAA, 2015.

[143] D. Miedziński, K. Kaczmarek, P. Rodo, N. Sahbon, M. Sochacki, and M. Łukasiewicz, "Missile aerodynamics model identification using flight data," in *2023 IEEE Aerospace Conference*, pp. 1–12, 2023.

[144] H. Oberle and W. Grimm, "BNDSCO: A Program for the Numerical Solution of Optimal Control Problems," tech. rep., DFVLR, 01 1989.

[145] A. Wächter, "Short tutorial: Getting started with ipopt in 90 minutes," in *Combinatorial Scientific Computing*, vol. 9061, pp. 1–17, 2009.

[146] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[147] A. Walther, A. Griewank, "Adol-c: A package for the automatic differentiation of algorithms written in c/c++," 2016.

[148] A. Walther, "Getting Started with ADOL-C," in *Combinatorial Scientific Computing* (U. Naumann, O. Schenk, H. D. Simon, and S. Toledo, eds.), vol. 9061 of *Dagstuhl Seminar Proceedings (DagSemProc)*, (Dagstuhl, Germany), pp. 1–10, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2009.

[149] A. Griewank and A. Walther, "Evaluating derivatives - principles and techniques of algorithmic differentiation, Second Edition," in *Frontiers in applied mathematics*, 2000.

[150] S. Forth, "An efficient overloaded implementation of forward mode automatic differentiation in MATLAB," *ACM Trans. Math. Softw.*, vol. 32, pp. 195–222, 06 2006.

[151] D. Oddenino, "AUTOCODING WORKING GROUP Automatic Code Generation for AOCS Flight SW," 2018.

[152] N. N. Moiseev, *Elementy teorii systemów optymalnych.* Warszawa: Wydawnictwa Naukowo-Techniczne, wyd. i. ed., 1983.

[153] R. Pytlak, *Numerical Methods for Optimal Control Problems with State Constraints.* Lecture Notes in Mathematics: 1707, Springer, 1999.

[154] R. Pytlak, *Interactive computer environment for solving optimal control problems - IDOS.* BEL Studio Sp. z o.o., 2012.

Spis rysunków

Spis tabel

Spis algorytmów