



POZNAN UNIVERSITY OF TECHNOLOGY

SKN/SP/570981/2023
Raport końcowy



Minister
Nauki

ACTIVE TRAJECTORY CONTROL

Koło naukowe: PUT Rocketlab

Kierownik projektu: dr inż. Bartosz Ziegler

Autorzy raportu: Tomasz Brodziński, Maksymilian Falbierski, Wojciech Gandziarowski, Mieszko Gaworzewski, Przemysław Huzar, Jagiełłowicz Olaf, Andrzej Rafalski, Kajetan Swoboda, Natalia Wiśniewska, Piotr Zacholski

Contents

1	Wstęp	1
1.1	Cel projektu	2
1.2	Przegląd stosowanych systemów sterowania	2
1.3	Założenia projektowe i ograniczenia	5
2	Mechanizm sterowania	6
2.1	Projekt	6
2.2	Prototyp i wytworzenie	7
3	Charakterystyka dynamiczna modelu rakiety i aktuatorów	9
3.1	Charakterystyka rakiety	9
3.1.1	Badana konfiguracja rakiety i aerodynamicznych powierzchni sterowych	9
3.1.2	Przeprowadzone numeryczne analizy przepływowe	11
3.1.3	Wyniki analiz	12
3.1.4	Wnioski wynikające z przeprowadzonych analiz	14
3.2	Charakterystyka jet vanów	16
3.2.1	Badanie charakterystyki dynamicznej i mechanicznej silnika Maxon	18
3.2.2	Sterowanie silnikiem przez UART: Implementacja wrappera w Pythonie	20
4	Algorytmy sterowania	22
4.1	Projekt	22
4.2	Implementacja	23
4.3	Testy	23
4.3.1	Scenariusz 1	24
4.3.2	Scenariusz 2	27
5	Metryka koszt/manewrowość	29
5.1	Opracowanie	29
6	Rezultaty projektu	32
6.1	Stanowisko Hardware in the Loop	32
6.2	Stanowisko do badania właściwości dynamicznych aktuatorów	35
6.3	Dokumentacja algorytmu sterującego	37
6.4	Dokumentacja wykonawcza mechanizmu sterującego stanowiska HIL	40
7	Podsumowanie	51
	Bibliography	52

Nomenklatura

ω	prędkość rotacji dla poszczególnych osi
AoA	Angle of Attack
C_T	Współczynnik momentowy
C_X	Współczynnik siły wokół osi X
C_Z	Współczynnik siły wokół osi Z
CFD	Computational Fluid Dynamics
CFDA	Control Fin Deflection Angle
CHT	Coupled Heat Transfer
CM_X	Współczynnik momentu wokół osi Y
DC	Direct Current
ESO	Extended State Observer
FVM	Finite Volume Method
HARM	High-speed Anti-Radiation Missile
HIL	Hardware-in-the-Loop
I	Prąd
J	Macierz momentu bezwładności
N	Wektor momentów generowanych przez powierzchnie sterujące
q	Kwaternion opisujący orientację
RCS	Reaction Control System
T	Moment obrotowy
T_k	Czas stabilizacji
TVC	Thrust Vector Control
v	Prędkość boczna w osi Y
w	Prędkość boczna w osi Z
xCP	Rozkład środka parcia na osi X

1 Wstęp

Raport i projekt został wykonany przez koło naukowe PUT Rocketlab, z Politechniki Poznańskiej - zespół studentów zajmujący się budowaniem, testowaniem i badaniami wykonywanymi podczas lotów eksperymentalnych raket sondażowych. Projekt powstał przy wsparciu pracowników uczelni, którzy pomagali w aspektach merytorycznych i zapewniają dostęp do aparatury badawczej.

1.1 Cel projektu

Celem projektu było opracowanie, budowa i testy aktuatorów na stanowisku naziemnym, a także budowa stanowiska do badania ich właściwości dynamicznych. Zaprojektowanie algorytmu sterowania, który umożliwi aktywną kontrolę toru lotu rakiety, pozwalając na stabilizację jej trajektorii. Dodatkowo zadaniem algorytmu, była również minimalizacja prędkości bocznej w apogeum, w celu zapewnienia bezpieczniejszych warunków odzysku rakiet. Niezbędnym w projekcie było także opracowanie narzędzia pozwalającego na symulowanie lotów rakiet eksperymentalnych z różnymi konfiguracjami aktuatorów i układów sterujących, w tym zarówno rozwiązaniami aerodynamicznymi, jak i systemami sterowania wektorowego działającymi w strumieniu gazów wylotowych silnika, mającego zapewnić kompatybilność do różnych platform napędowych.

1.2 Przegląd stosowanych systemów sterowania

Obecny stan wiedzy dotyczącej rakiet wyposażonych w systemy sterowania jest znaczący, jednak szczegółowe informacje na ten temat są często chronione i trudnodostępne. Niemniej możliwe jest określenie głównych cech oraz najistotniejszych czynników projektowych tych konstrukcji. W przypadku działania rakiety w atmosferze, kluczowym elementem projektu jest całkowita powierzchnia aerodynamiczna. Oznacza to, że średnica rakiety, jej długość, geometria i wymiary głowicy oraz powierzchni aerodynamicznych muszą być zoptymalizowane. Ważne jest również uwzględnienie napędu i jego parametrów. Oczywiście jest, że przed każdym startem znane są warunki brzegowe oraz określone wymagania misji.

Do sterowania orientacją rakiety za pomocą sił aerodynamicznych stosuje się stateczniki, które mogą występować w różnych konfiguracjach. W trakcie lotu przez gęstsze warstwy atmosfery ten rodzaj sterowania jest najbardziej efektywny ze względu na wystarczający opór powietrza.

Przykładem rakiety z systemem aerodynamicznego sterowania jest AGM-88A HARM (High-speed Anti-Radiation Missile). Jest to precyzyjnie naprowadzany, naddźwiękowy pocisk klasy powietrze-ziemia, przeznaczony głównie do niszczenia systemów obrony przeciwlotniczej. Typowa wersja tego pocisku ma długość 4170 mm, masę około 361 kg, średnicę 254 mm i maksymalną prędkość około 2.9 Macha. System sterowania oparty jest na czterech płatach umieszczonych centralnie i symetrycznie wokół osi wzdłużnej rakiety.

Manewrowość tego typu rakiet bazuje na siłach aerodynamicznych i momentach generowanych przez powierzchnie sterujące oraz korpus rakiety w różnych konfiguracjach. AGM-88A HARM wykorzystuje stery nazywane „skrzydłami”, umieszczone przed statycznymi płetwami stabilizującymi (ogonowymi). Popularne są również „canardy” – małe powierzchnie sterujące umieszczone ponad środkiem ciężkości w pobliżu nosa rakiety, które zapewniają kontrolę i stabilność, zwłaszcza przy niższych prędkościach. Canardy mogą szybko korygować trajektorię lotu dzięki swojemu przedniemu położeniu. Skrzydła, zwykle zlokalizowane dalej z tyłu, odpowiadają za generowanie siły nośnej i stabilności przy wyższych prędkościach, jednocześnie wpływając na manewrowość w inny sposób niż canardy.

Inna popularna konfiguracja powierzchni sterujących wykorzystuje tylne ruchome stery poprzedzone większymi, statycznymi statecznikami stabilizującymi, które zapewniają stabilność podczas lotu. Przykładem rakiety o tej konfiguracji jest AIM-54 Phoenix – radarowy pocisk dalekiego zasięgu klasy powietrze-powietrze. Charakteryzuje się on długością około 3900 mm, masą 460 kg, średnicą 380 mm i maksymalną prędkością sięgającą 5 Macha.

Kontrola trajektorii rakiety nie ogranicza się wyłącznie do sił aerodynamicznych. Wykorzystuje również sterowanie wektorem ciągu (TVC - Thrust Vector Control), które ma szczególne znaczenie poza atmosferą. W zależności od typu napędu stosuje się różne metody:

Gimbalowanie wektora ciągu (Gimbaled Thrust) – Cały silnik lub jego dysza są obracane w celu zmiany wektora ciągu. W nowoczesnych konstrukcjach rakiet, takich jak silniki Raptor firmy SpaceX, stosuje się również przegubowe systemy obejmujące pompy paliwa i utleniacza.

Systemy reakcyjne (RCS - Reaction Control System) – Służą do precyzyjnego sterowania ruchem postępowym lub obrotowym rakiety poprzez generowanie ciągu w określonym kierunku. Powszechnie

wykorzystywane są w statkach kosmicznych do kontroli orientacji i manewrów dokowania. Przykładem są silniki Verniera.

Sterowanie za pomocą łopatek w dyszy (Jet Vanes) – Zmieniają kierunek przepływu gazów wylotowych przy prędkościach naddźwiękowych. Pozwalają one na kontrolowanie osi rakiety, choć kosztem efektywności. Materiały używane do ich budowy muszą być odporne na wysokie temperatury lub posiadać aktywne chłodzenie. Technologię tę stosowano m.in. w niemieckich raketach V-2, a także w nowoczesnych konstrukcjach.

Każda z tych metod sterowania ma swoje specyficzne zastosowanie i zalety, w zależności od warunków lotu i wymaganej precyzji kontroli. Systemy te, połączone w jeden układ, pozwalają na precyzyjne prowadzenie rakiety w atmosferze i poza nią, niezależnie od jej przeznaczenia.

Poniżej, w Tab. 1 przedstawiono analizę porównawczą wybranych metod wektorowania ciągu, stosowanych w napędach raketowych.

Table 1: Analiza porównawcza wybranych metod wektorowania ciągu silników raketowych

Metoda	Zalety	Wady
Gimbalowana komora spalania	<ul style="list-style-type: none"> • Czas trwania jest ograniczony jedynie dostępnością materiału pędnego, • kąt wychylenia do $\pm 12^\circ$, • pomijalne straty ciągu. 	<ul style="list-style-type: none"> • Wymagane wykorzystanie elastycznych przewodów rurowych
Ruchoma dysza	<ul style="list-style-type: none"> • Sprawdzona, przetestowana technologia, • brak utraty ciągu podczas poruszania całą dyszą, • znaczny zakres ruchu, do $\pm 20^\circ$. 	<ul style="list-style-type: none"> • Wymaga sferycznej ochrony termicznej, • zmienna moc sterowania, • wymaga ciągłego użycia, aby utrzymać szczelność.
Jet vanes	<ul style="list-style-type: none"> • Pojedyncza dysza umożliwia obrót rakiety wzdłuż jej osi wzdłużnej, • umożliwienie zakresu ruchu do $\pm 9^\circ$, • sprawdzona technologia szeroko stosowana w pociskach raketowych, • możliwość szybkiego reagowania, • Niskie zapotrzebowanie na moc i masę siłownika. 	<ul style="list-style-type: none"> • Ograniczenie do raket o krótkim czasie pracy silnika lub wykorzystujących niskotemperaturowe materiały pędne, • straty ciągu od 2 do 5%, • erozja spowodowana oddziaływaniem gorących gazów wylotowych powoduje zużycie i degradację powierzchni łopatek w miarę upływu czasu.
Jet tabs	<ul style="list-style-type: none"> • Siła boczna wprost proporcjonalna do stosunku powierzchni jet taba do powierzchni dyszy, • kompaktowy montaż, • sprawdzona technologia, • gdy wektorowanie nie jest aktywne, nie występuje utrata ciągu. 	<ul style="list-style-type: none"> • Ograniczenie do raket o krótkim czasie pracy silnika lub wykorzystujących niskotemperaturowe materiały pędne, • erozja spowodowana oddziaływaniem gorących gazów wylotowych powoduje zużycie i degradację powierzchni jet taba w miarę upływu czasu, • utrata ciągu podczas pracy (równa generowanej sile bocznej).

Table 1: Analiza porównawcza wybranych metod wektorowania ciągu silników raketowych (Continued)

Metoda	Zalety	Wady
Jetavator	<ul style="list-style-type: none"> • Lekki system, • siła boczna zależna liniowo od kąta wychylenia. 	<ul style="list-style-type: none"> • Ograniczenie do raket o krótkim czasie pracy silnika lub wykorzystujących niskotemperaturowe materiały pędne, • wymagane osłony termiczne do ochrony systemów sterowania i zewnętrznej części dyszy, • utrata ciągu niezależnie od wychylenia.
Wtrysk gorących gazów	<ul style="list-style-type: none"> • Mniejsza masa niż w przypadku systemów wtrysku cieczy, • możliwość szybkiego reagowania, • wymagana niewielka liczba operacji przed uruchomieniem. 	<ul style="list-style-type: none"> • Nie może być używany tam, gdzie wymagana jest precyzyjna kontrola prędkości rakiety, • wymagają specjalnych zaworów kompatybilnych z wtryskiwanym gorącym gazem, • ograniczony czas pracy ze względu na trudne warunki termiczne.
Wtrysk cieczy	<ul style="list-style-type: none"> • czas działania jest ograniczony przez dostępność zapasu cieczy do wtrysku, • umożliwiający ruch dyszy do $\pm 6^\circ$, • impuls właściwy wtryskiwanego czynnika niemal kompensuje negatywny wpływ masy układu wtryskowego, • sprawdzona technologia, • możliwość szybkiego reagowania. 	<ul style="list-style-type: none"> • optymalna efektywność działania wymaga użycia niebezpiecznych cieczy, • wymaga częstej konserwacji, co z kolei zwiększa ryzyko potencjalnych wycieków i uwalniania toksycznych oparów podczas obchodzenia się z niektórymi materiałami pędnymi.
Pomocnicze komory spalania	<ul style="list-style-type: none"> • Niska zapotrzebowanie na moc silownika, zasilane z głównego układu napędowego, • kompaktowa i sprawdzona technologia. 	<ul style="list-style-type: none"> • Zwiększa złożoność i masę konstrukcji układu napędowego., • nieużywane powszechnie w przemyśle od wielu lat (z wyjątkiem zastosowań takich jak rakiety R-7 i ich pochodne - rakiety Sojuz), • generują niskie momenty.

Spośród wszystkich metod wymienionych w tabeli 1, po przeanalizowaniu zalet i wad, można założyć, że metoda wektorowania ciągu za pomocą wychylnych łopatek jet vane będzie najodpowiedniejszym rozwiązaniem dla rakiety sondażowej ze względu na stosunkowo prosty montaż, obsługę i niezawodność. Pomimo wad, takich jak utrata ciągu i erozja podczas pracy, metoda ta jest również najbardziej opłacalnym rozwiązaniem dla studenckiej hybrydowej rakiety sondażowej. Poniżej przedstawiono graficzne reprezentacje kilku wybranych metod wektorowania ciągu w raketowych układach napędowych (Rys. 1 and Rys. 2).

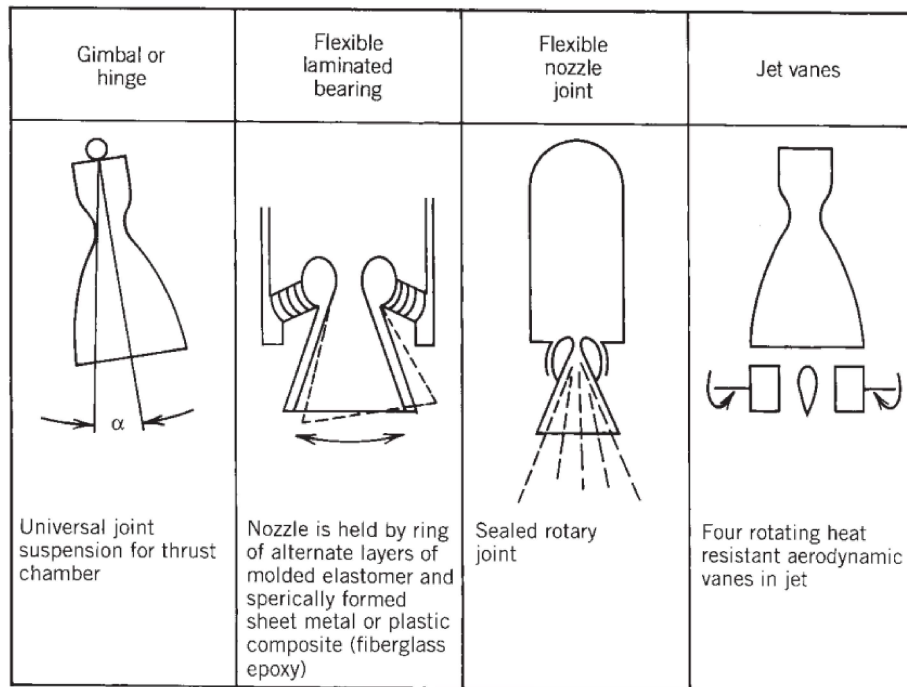


FIGURE 1: Graficzne przedstawienie kilku wybranych technik wektorowania ciągu w raketowych układach napędowych [3].

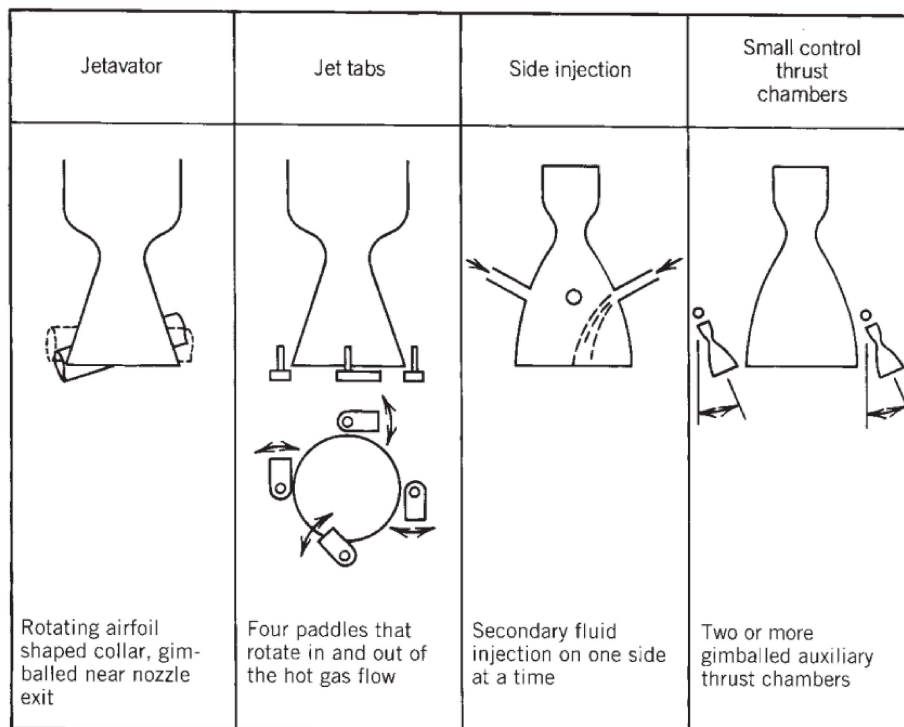


FIGURE 2: Graficzne przedstawienie kilku wybranych technik wektorowania ciągu w raketowych układach napędowych [3].

1.3 Założenia projektowe i ograniczenia

Założenia projektowe:

- Opracowanie, budowa i testy aktuatorów: Kluczowym celem było zaprojektowanie i przetestowanie aktuatorów, które mogłyby skutecznie sterować trajektorią lotu rakiety. Testy przeprowadzono na specjalnie zaprojektowanym stanowisku naziemnym, które miało symulować rzeczywiste warunki działania tych urządzeń.
- Budowa stanowiska do badania właściwości dynamicznych aktuatorów: Stworzono stanowisko badawcze

i operacyjne, np. zmieniające się obciążenia czy wpływ sił aerodynamicznych.

- Zaprojektowanie algorytmu sterowania: Algorytm miał zapewnić aktywną kontrolę toru lotu rakiety, umożliwiając stabilizację jej trajektorii w trakcie różnych faz lotu. W szczególności, wymagano minimalizacji prędkości bocznej w apogeum, co było krytyczne dla zwiększenia bezpieczeństwa i efektywności procesu odzysku rakiety.
- Symulacja lotu raket eksperymentalnych: Opracowano narzędzie pozwalające na symulowanie trajektorii lotu rakiety w różnych konfiguracjach systemów sterowania. Symulator uwzględniał zarówno powierzchnie aerodynamiczne (np. stateczniki), jak i systemy sterowania wektorem ciągu, aby umożliwić integrację z różnymi typami platform napędowych.
- Kompatybilność z różnymi platformami napędowymi: Założono, że rozwiązania projektowe będą uniwersalne i możliwe do zastosowania w raketach z różnymi systemami napędowymi, np. hybrydowymi, ciekłymi lub stałymi.

Ograniczenia projektowe:

- Wpływ środowiska i dynamiki lotu: Aktuatory musiały działać niezawodnie w zmiennych warunkach atmosferycznych oraz podczas dynamicznych zmian trajektorii rakiety. Trzeba było uwzględnić zmienność sił aerodynamicznych i momentów w zależności od prędkości lotu i gęstości atmosfery. Przy użyciu symulacji.
- Zależność od fazy lotu: W początkowej fazie lotu dominowały siły generowane przez jet vany (sterowanie gazami wylotowymi). Jednak ich skuteczność zanikała po zakończeniu pracy silnika, co wymagało przejścia na sterowanie aerodynamiczne, którego efektywność spadała przy niższych prędkościach (np. podczas lądowania).
- Ograniczenia konstrukcyjne: Geometria rakiety musiała zostać zoptymalizowana pod kątem zapewnienia odpowiedniego zapasu stateczności (odległość między środkiem ciężkości a środkiem parcia). Jednocześnie należało zminimalizować wzrost oporu aerodynamicznego wywołany przez dodatkowe powierzchnie sterujące umiejscowione za wylotem gazów z silnika.
- Trudności w modelowaniu: Modele symulacyjne musiały uwzględniać nieliniowości i niepewności wynikające z wypalania paliwa oraz zmieniających się właściwości aerodynamicznych w trakcie lotu. Zmienne takie jak liczba Macha, kąt natarcia i kąt wychylenia powierzchni sterowych wymagały kompleksowej analizy i symulacji CFD (Computational Fluid Dynamics).
- Ograniczenia technologiczne: Sterowanie jet vanami wymagało dobrania materiałów odpornych na ekstremalne temperatury i obciążenia mechaniczne występujące w dyszy wylotowej silnika raketowego. Wysokie wymagania precyzji i szybkości działania sterowników musiały być spełnione przy użyciu dostępnych na rynku komponentów (np. silników i enkoderów firmy Maxon).
- Bezpieczeństwo i odzysk: Algorytm sterowania musiał uwzględniać minimalizację prędkości bocznej w apogeum, co bezpośrednio wpływało na bezpieczeństwo lądowania i odzyskiwania rakiety. Rozwiązania musiały być skalowalne i kompatybilne z gotowymi systemami odzysku.

2 Mechanizm sterowania

2.1 Projekt

W ramach projektu zdecydowano się na opracowanie mechanizmu umożliwiającego niezależne sterowanie powierzchniami sterowymi. Na jeden moduł mechanizmu przypada zintegrowana ze sobą powierzchnia sterowa w formie panelowej w kombinacji z lotką umieszczoną w strumieniu gazów wylotowych, które

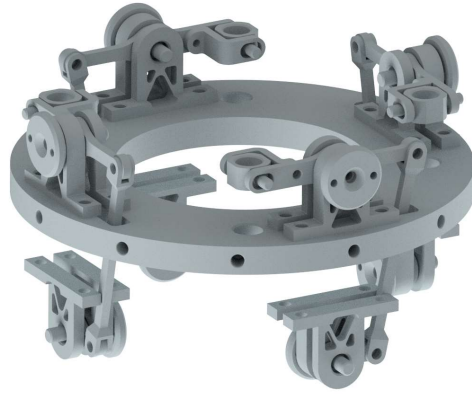


FIGURE 3: Mechanizm sterujący dla czterech powierzchni sterowych.

Pojedynczy moduł jest wyposażony w silnik elektryczny sprzężony z przekładnią liniową przekształcającą moment obrotowy w siłę liniową, umożliwiającą obrót powierzchni sterowej wokół własnej osi.

Bazując na wnioskach uzyskanych podczas analiz różnego typu rozwiązań w zakresie sterowania w ramach przeglądu literaturowego, do realizacji projektu dobrano jedną konkretną ideę, ze względu na jej uniwersalność oraz możliwość integracji z raketami badawczymi wyposażonymi w różnego typu napędy, poprzez wykorzystanie tzw. "blast tube" umożliwiającego wygenerowanie objętości wewnątrz kadłuba rakiety koniecznej do ulokowania systemu mechanizacji. Dodatkowo modułowa forma mechanizmu daje możliwość zastosowania w ograniczonej formie również do sterowania w układzie kaczki (powierzchnie sterowe zorientowane bliżej czubka rakiety), bądź do sterowania jedną powierzchnią sterową w formie panelowej, umieszczonej bliżej części ogonowej lub łopatką umieszczoną w strumieniu gazów wylotowych. Z powodu wskazanej powyżej uniwersalności tego typu mechanizmu w ramach projektu rozwijana była jedna koncepcja, którą optymalizowano pod kątem uproszczenia obróbki, obniżenia kosztów ewentualnej seryjnej produkcji systemu i zapewnienia jak największej skuteczności układu. Dodatkowo modułowość umożliwia stosowanie systemu zarówno dla układów aerodynamicznych wykorzystujących zarówno dwie, trzy, jak i cztery niezależne powierzchnie sterowe (rys. 3 przedstawia układ w zabudowie dla czterech powierzchni sterowych).

2.2 Prototyp i wytworzenie

Głównym celem pojedynczego modułu jest przetworzenie siły liniowej otrzymanej z napędu liniowego dowolnego typu ponownie w moment obrotowy działający na wał powierzchni sterowej. W tym celu wykorzystano serię komponentów zapewniających wymagane stopnie swobody na kolejnych połączeniach oraz dźwignie dopasowaną gabarytowo pod kątem zapotrzebowania na moment obrotowy oraz dostępnej do wykorzystania przestrzeni wewnątrz kadłuba rakiety. Łożyskowanie wału powierzchni sterowej zrealizowano poprzez osadzenie dwóch łożysk ślizgowych, ze względu na ich kompaktowość i parametry wytrzymałościowe oraz ślizgowe, spełniające wymagania projektowe. Tuleje ślizgowe zostały wykorzystane również na połączeniu adaptera pomiędzy napędem liniowym a dźwignią głównej powierzchni sterowej. Układ kinematyczny wskazanego połączenia wymusza zarówno obrotowy stopień swobody wokół osi pionowej adaptera, jak i liniowy wzdłuż ramienia; jest to spowodowane sztywnym montażem silników do struktury nośnej potencjalnej rakiety. Drugi koniec dźwigni jest połączony (z wykorzystaniem tulei ślizgowej) z ciągnem, którego funkcją jest dalsze przekazywanie napędu na dźwignię lotki umieszczonej w strumieniu gazów. Układ umożliwi zmianę przełożenia pomiędzy główną powierzchnią sterową a lotką poprzez zmianę długości ramienia dźwigni i wynikającej z nich długości ciągnia.

Montaż powierzchni do wałków jest realizowany poprzez połączenia śrubowe z wykorzystaniem adaptera dedykowanego konkretnej geometrii powierzchni sterowej, zabezpieczone dodatkowo poprzez połączenie



FIGURE 4: Komponenty modułu mechanizmu sterowania.

Materiały dobrane do wykonania komponentów to głównie stop aluminium 7075, oraz wały wykonane z stali nierdzewnej. Wały wykonano z stali ze względu na lepsze właściwości ślizgowe od aluminium, a ich niewielkie gabaryty nie powodują znacznego przyrostu masy mimo wykorzystania materiału charakteryzującego się niższą wytrzymałością właściwą. Pozostałe komponenty struktury wykonano z wskazanego powyżej aluminium EN AW-7075, ze względu na właściwości wytrzymałościowe (przede wszystkim względnie wysoka wytrzymałość właściwa), oraz właściwości obróbcze - jest to względnie łatwo obrabialny materiał. Technologia obróbki została dobrana tak, aby zminimalizować konieczność korzystania z maszyn numerycznych cztero- i więcej osiowych. Redukuje to koszt wytwarzania komponentów, oraz zwiększa dostępność projektu.

Grafika 5 przedstawia układ wykorzystywanych komponentów dla jednego modułu mechanizmu, komponenty to kolejno:

1. Wręga montażowa - głównym zadaniem wręgi montażowej wykonanej z aluminium jest przenoszenie sił od powierzchni sterowych na kadłub rakiety, oraz zapewnienie punktów montażowych komponentów układu
2. Górny montaż łożyska - punkt montażowy dla łożyska ślizgowego
3. Dolny montaż łożyska - punkt montażowy łożyska ślizgowego
4. Oś powierzchni sterowej
5. Dźwignia "Jet Vane" - przenosi moment obrotowy na powierzchnie sterową umieszczoną w strumieniu gazów wylotowych z silnika
6. Dźwignia głównej powierzchni sterowej - przenosi moment obrotowy na powierzchnię sterową, oraz zapewnia punkty montażowe umożliwiające przenoszenie siły z napędu liniowego
7. Popychacz - zapewnia połączenie pomiędzy dźwigniami powierzchni sterowych w celu wprowadzania ich w ruch z określonym przełożeniem, wykorzystując jedno źródło napędu
8. Adapter umożliwiający połączenie powierzchni sterowej z kadłubem rakiety

9. Łożysko liniowe - konwertuje moment obrotowy z śruby napędzanej poprzez silnik elektryczny na siłę osiową

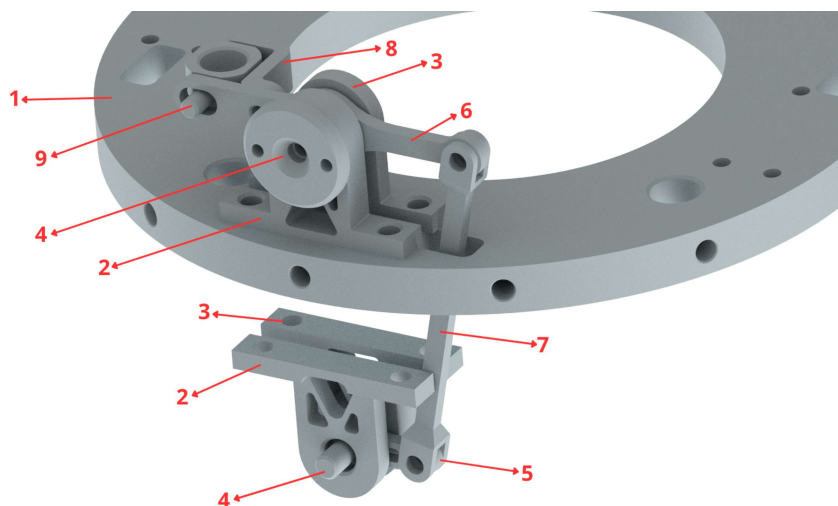


FIGURE 5: Komponenty modułu mechanizmu sterowania. Sprężone ze sobą osie powierzchni aerodynamicznej i łopatki wektorowania ciągu (jet vane) o różnych długościach dźwignie pozwala na niezależne dobieranie zakresu kąta ruchu obu powierzchni

3 Charakterystyka dynamiczna modelu rakiety i aktuatorów

3.1 Charakterystyka rakiety

3.1.1 Badana konfiguracja rakiety i aerodynamicznych powierzchni sterowych

By opracować narzędzie do symulowania lotu rakiety z zaimplementowanym sterowaniem, niezbędne było dobranie konstrukcji, która miałaby być sterowana. Warunkiem koniecznym było to, by dobrana rakietą była już przetestowana lotnie, oraz miała znane osiągi i parametry aerodynamiczne, w celu wyznaczenia sił i momentów na nią działających podczas stosowania sterowania, co dalej przekładałoby się na możliwość wykonania narzędzia symulującego jej lot. Wybór padł na konstrukcję HEXA 4 [Rys. 6], rakieta o napędzie hybrydowym stworzoną przez KN PUT Rocketlab. Zaprojektowana jest w taki sposób, by osiągnąć apogeum 30000 stóp (9144 m) i wylądować przy użyciu dwustopniowego systemu odzysku, wyposażonego w dwa spadochrony. Rakietą jest klasyfikowana jako hybrydowa, ponieważ do napędu wykorzystuje ciekły utleniacz w postaci podtlenku azotu (N_2O) oraz stałe paliwo na bazie parafiny. Ma długość 4.2 metra i kaliber 160 mm. HEXA 4 może osiągnąć maksymalną prędkość Mach 1,5.

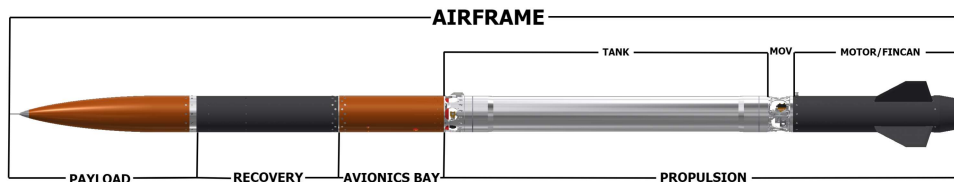


FIGURE 6: Przegląd konstrukcyjny rakiety HEXA 4 [2]

Rakietą ta jest projektem PUT Rocketlab. Mając za sobą dwa loty i zdobycie pierwszego miejsca w kategorii 30k SRAD Liquid/Hybrid podczas zawodów IREC Spaceport America Cup 2023, rakietą ta stanowi doskonały wybór jako konstrukcja, którym ma sterować badany system.

Kolejnym etapem w procesie projektowym aerodynamicznych powierzchni sterowych jest dobranie geometrii tychże powierzchni.

Koniecznym było uwzględnienie wielu czynników, takich jak prostota systemu, odporność na siły

przednia powierzchnia jest statyczna, odpowiedzialna za utrzymanie stabilności rakiety, a powierzchnia bliżej ogona, mniejsza od powierzchni statycznej, jest sterująca. Powierzchnia sterująca może obracać się wzdłuż swojej osi pionowej. Powierzchnie te muszą zapewniać wystarczający zapas stateczności, aby rakieta utrzymywała swoją stabilność statyczną, jednocześnie będąc w stanie wywołać siły i momenty aerodynamiczne (w szczególności przez powierzchnię sterującą), aby móc sterować. Wszystko to musi wiązać się z minimalnym wzrostem całkowitego oporu aerodynamicznego. By dobrać odpowiednią geometrię został wykorzystany program *OpenRocket*, który przewiduje rozkład środka ciężkości rakiety, jak i środka parcia. Im większa odległość między tymi punktami, tym większy zapas stateczności jest spodziewany. geometria została dobrana w taki sposób by odległość między wspomnianymi punktami była na podobnym poziomie, jak w przypadku bazowej konfiguracji rakiety HEXA 4. Porównanie to przedstawione jest na Rys. 7 (Rakieta HEXA 4) i na Rys. 8 (Zaproponowana konfiguracja powierzchni sterowych).

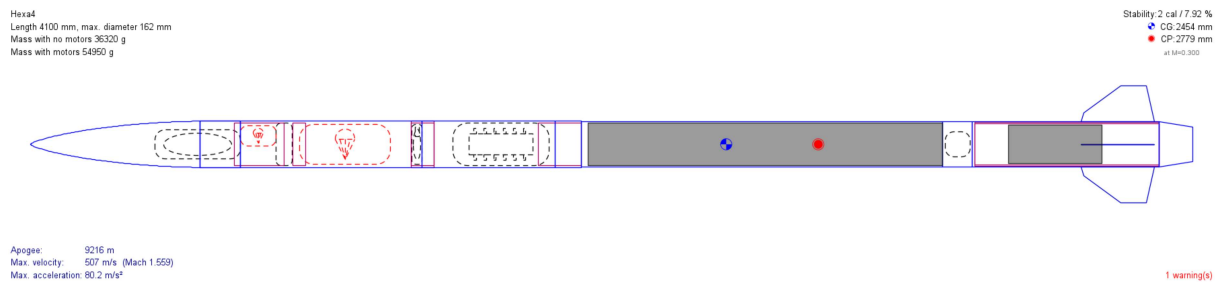


FIGURE 7: HEXA 4 w bazowej konfiguracji [OpenRocket] [1]

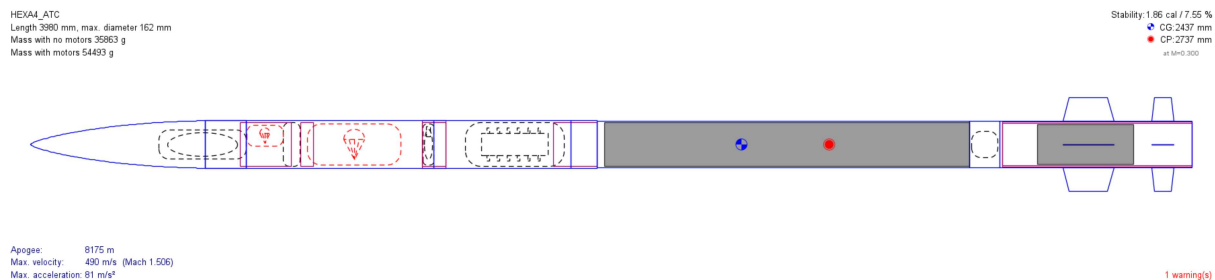


FIGURE 8: HEXA 4 w konfiguracji powierzchni sterujących [OpenRocket] [1]

Na Rys. 9 przedstawiona jest wizualizacja 3D badanej konfiguracji powierzchni statycznej i sterującej w sekcji ogonowej rakiety HEXA 4.

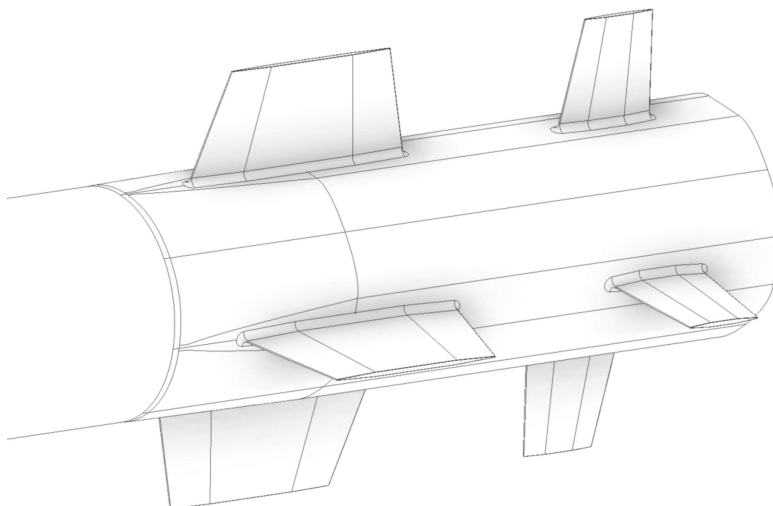


FIGURE 9: Wizualizacja 3D badanej konfiguracji [1].

3.1.2 Przeprowadzone numeryczne analizy przepływowe

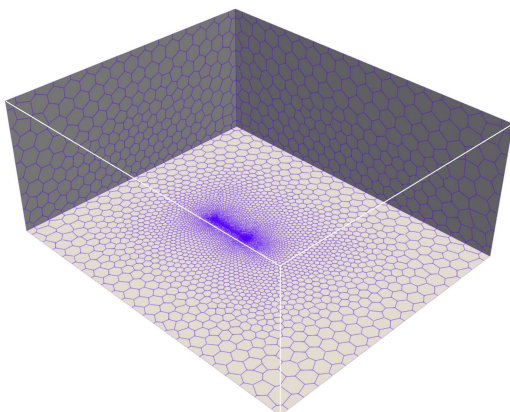
Głównym celem przeprowadzonych analiz była ocena proponowanej konstrukcji płetwy statycznej i sterującej pod kątem generowania sił i momentów aerodynamicznych na płatowcu rakiety. Przetestowano wiele konfiguracji, ze zmienną prędkością (liczbą Macha), kątem natarcia rakiety (względem swobodnego strumienia powietrza) i kątem odchylenia powierzchni sterującej. Wartości tych parametrów zostały uzyskane z profilu lotu HEXA 4, gwarantując, że przeprowadzone analizy przedstawiają wystarczająco różnicowane wyniki, aby na ich podstawie stworzyć charakterystyki aerodynamiczne. Zmienne te zostały przedstawione w Tab. 2. Taka liczba zmiennych parametrów wskazywała na konieczność przeprowadzenia 48 oddzielnych analiz. Po przeprowadzeniu analiz i przetworzeniu wyników, wypadkowe siły i momenty zostały porównane między sobą. Uzyskane dane można było następnie wykorzystać do implementacji algorytmu sterowania i symulacji, które zostały wykonane przez PUT Rocketlab.

Liczba Macha [-]	Kąt natarcia rakiety [°]	Kąt odchylenia powierzchni sterowej [°]
0.6	0.5	0.0
0.9	2.0	1.5
1.3	10.0	5.0
1.5		10.0

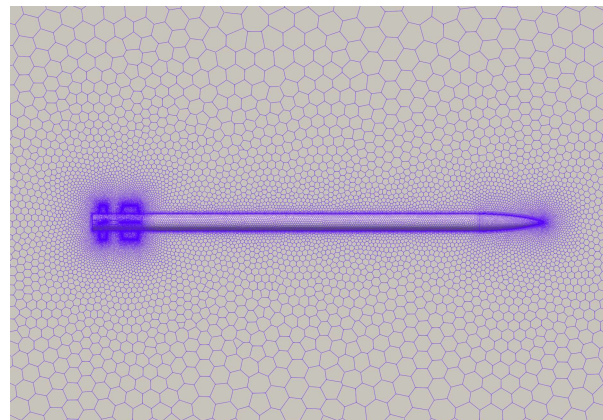
TABLE 2: Parametry zmienne w przeprowadzonych analizach

Metodą wybraną do przeprowadzenia analiz była **Obliczeniowa Mechanika Płynów (CFD)**. Ze względu na brak odpowiednich modeli analitycznych oraz fakt, że przeprowadzenie skalowanych testów w tunelu aerodynamicznym z przepływami okołodźwiękowymi i naddźwiękowymi było wówczas nieosiągalne, CFD było jedyną opcją, która mogła dać wyniki zbliżone do rzeczywistości. CFD wykorzystuje analizę numeryczną i struktury danych do rozwiązywania i analizowania problemów związanych z przepływem płynów wokół lub wewnątrz obiektów. Dzięki zastosowaniu algorytmów opartych na oprogramowaniu obliczeniowym, które wykorzystuje metodę objętości skończonych (FVM) i równania rządzące dynamiką płynów, CFD umożliwia symulację interakcji gazów i cieczy z powierzchniami, określonymi przez parametry początkowe i warunki brzegowe. Służy do przewidywania i modelowania zachowania przepływu płynów w złożonych systemach, optymalizując wydajność i ulepszając projektowanie, jednocześnie zmniejszając potrzebę czasochłonnych i kosztownych eksperymentów fizycznych.

Wybrany środowiskiem do przeprowadzenia analiz CFD został *Ansys Fluent* - program bardzo dobrze znany w środowisku CFD. Pierwszym krokiem by przeprowadzić analizy i uzyskać jakościowo zadowalające wyniki, było wygenerowanie siatki obliczeniowej, czyli przestrzeni wokół badanego obiektu, która podzielona jest na mniejsze elementy pomiędzy którymi zachodzą obliczenia numeryczne. Wygenerowana siatka liczyła około 6.4 miliona elementów, i była odpowiednio zagęszczona przy samej rakiecie, by poprawnie przewidzieć i obliczyć zachodzące zjawiska turbulencji w warstwie przyściennej. Wygenerowana siatka przedstawiona jest na Rys. 10.



(A) Widok na całą domenę obliczeniową



(B) Zbliżenie na rakieta

Kolejnym krokiem w procesie przygotowawczym do analiz CFD było dobranie warunków brzegowych oraz lokalnych i globalnych ustawień analizy. Warunki brzegowe, które zadają podstawowe parametry przepływu na brzegach domeny obliczeniowej, zostały obrane w taki sposób, by jak najbardziej odzwierciedlić warunki fizyczne panujące podczas lotu. Przy każdej z 48 analiz parametry sterujące tych warunków, zmieniane były zgodnie z tabelą Tab. 2. Z kolei dobranie lokalnych i globalnych ustawień analiz polegało na dobraniu typu formulacji analizy, przepływu i schematów dyskretyzacji. Ustawienia te dobrane zostały tak, by zapewnić jak najwyższą stabilność analiz, przy małym koszcie obliczeniowym i minimalnej dyssypacji numerycznej, która mogłaby generować błędy numeryczne.

3.1.3 Wyniki analiz

Wyniki przeprowadzonych analiz CFD pozwalają na porównanie testowanych konfiguracji pod względem istotnych czynników dla konkretnego przypadku i sprawdzenie przyjętych wcześniej założeń teoretycznych. Najważniejszymi czynnikami do porównania między analizami był wpływ zmieniającego się kąta odchylenia powierzchni sterującej (CFDA - Control Fin Deflection Angle) w różnych konfiguracjach kąta natarcia rakiety i liczby Macha. Oprócz wizualizacji, siły i momenty aerodynamiczne zostały uzyskane z każdej analizy w celu późniejszego przetworzenia i wizualizacji współczynników sił i momentów, potrzebnych do walidacji systemu kontroli aerodynamicznej pod względem jego zdolności do wywoływania zmian trajektorii lotu, a później do implementacji algorytmów sterowania. Przetwarzanie końcowe zostało opisane w rozdziale 3.1.4.

Rys. 11 przedstawia porównanie ciśnienia statycznego na korpusie i statecznikach rakiety pomiędzy różnymi wartościami CFDA, w Mach 1.3 i AoA rakiety równym 0.5° . Każda grafika przedstawia dwie strony powierzchni sterującej, aby zobrazować zarówno stronę ciśnieniową (górną połowę), jak i ssącą (dolną połowę). Zgodnie z założeniami, im większe CFDA, tym bardziej widoczne jest niższe ciśnienie generowane po stronie ssącej i wyższe ciśnienie po stronie ciśnieniowej powierzchni sterującej. W przypadku $CFDA = 0.0^\circ$, widoczny jest taki sam rozkład ciśnienia po obu stronach.

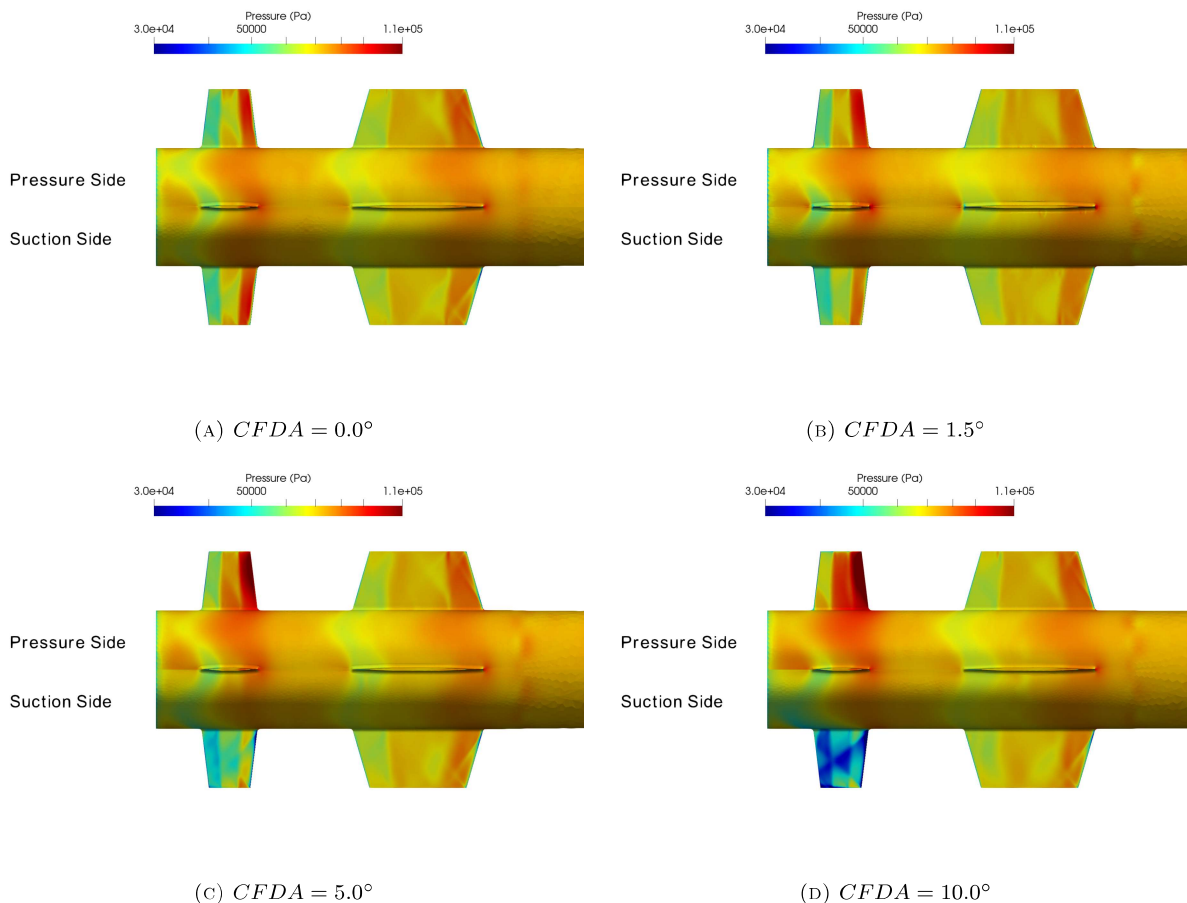


FIGURE 11: Wizualizacja nadciśnienia i podciśnienia na powierzchni sterującej (Mach 1.3, $AoA = 0.5^\circ$) [1].

Następnie na Rys. 12 pokazano wpływ zmiennego CFDA na korpus rakiety i stateczniki. Pola niskiego i wysokiego ciśnienia są widoczne po odchyleniu powierzchni sterującej.

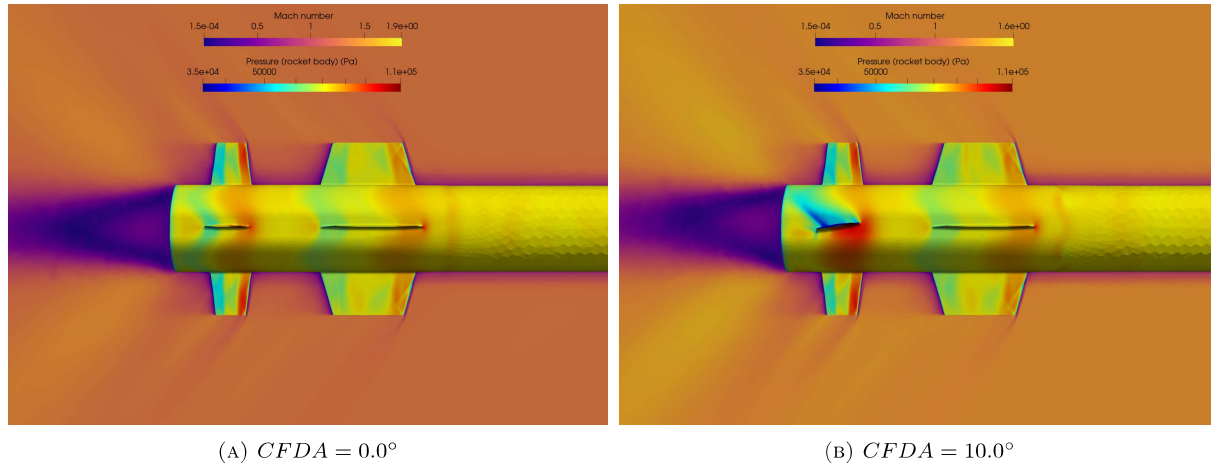


FIGURE 12: Wpływ odchylenia powierzchni sterującej na pola ciśnienia na korpusie i statecznikach (Mach 1.3, $AoA = 0.5^\circ$) [1].

Oprócz efektów zmiennego CFDA, uchwycono również efekty przepływu naddźwiękowego. Rys. 13 pokazuje porównanie rakiety w przepływie poddźwiękowym (Mach 0.6) i naddźwiękowym (Mach 1.3). Naddźwiękowe fale uderzeniowe, tworzące się na czubku nosa, a także w najbardziej wystających punktach stateczników, są wyraźnie widoczne w Mach 1.3. Ciśnienie w punkcie stagnacji na czubku nosa jest również większe w porównaniu do przepływu poddźwiękowego. Patrząc na cień aerodynamiczny (w polu przepływu Macha) pozostawiony przez ракетę, widać, że jego rozmiar jest znacznie większy w Mach 1.3 niż w Mach 0.6. Wszystkie te efekty są oczekiwane w reżimie przepływu naddźwiękowego.

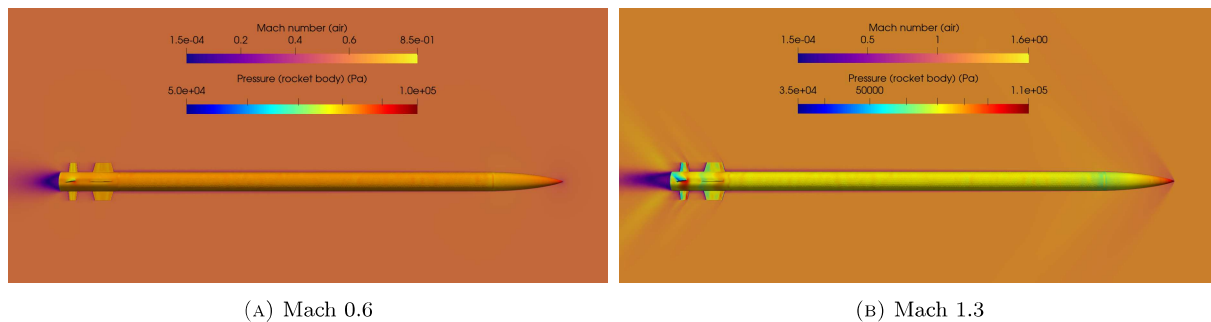


FIGURE 13: Pola przepływu poddźwiękowe i naddźwiękowe ($AoA = 0.5^\circ$, $CFDA = 10.0^\circ$) [1].

Stateczniki rakiety mają duży wpływ na dynamikę lotu naddźwiękowego. Oprócz widocznych różnic ciśnienia na ich powierzchni, z widocznymi falami uderzeniowymi wpływającymi na rozkład ciśnienia, same stateczniki wytwarzają własne fale uderzeniowe. Są one widoczne na rys. 14. Poza tym, aerodynamiczny cień stateczników jest również widoczny w konturach liczby Macha w polu przepływu w pobliżu ogona rakiety.

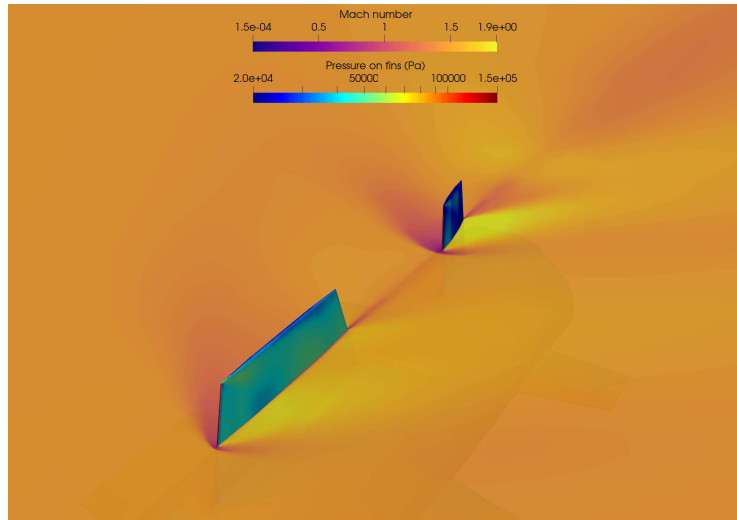


FIGURE 14: Kontury Liczby Macha dla wokół stateczników (Mach 1.5, $AoA = 10.0^\circ$, $CFDA = 10.0^\circ$) [1].

W warunkach lotu, gdy AoA rakiety jest bliskie zeru, wpływ powierzchni statycznej na przepływ wokół powierzchni sterującej jest pomijalny. Przy wyższych wartościach kąta natarcia powierzchnia statyczna tworzy wiry wpływające na przepływ wokół powierzchni sterującej. Ten negatywny efekt może prowadzić do mniejszej siły i momentu indukowanego przez powierzchnię sterującą. Oczekuje się zatem, że algorytm sterowania nie pozwoli na wprowadzenie tak wysokich wartości kąta natarcia rakiety. Efekt ten pokazano na rysunku Rys. 15.

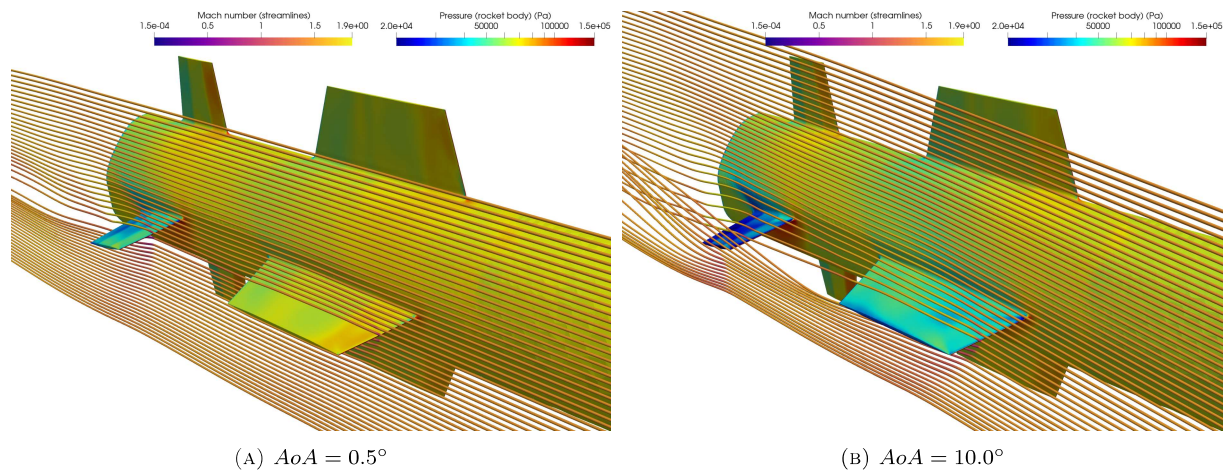


FIGURE 15: Porównanie przepływu przy niskim i wysokim kącie natarcia rakiety (Mach 1.5, $CFDA = 10.0^\circ$) [1].

3.1.4 Wnioski wynikające z przeprowadzonych analiz

Po przeprowadzeniu analiz uzyskano wynikowe dane siły i momentu w postaci raportów Ansys Fluent. Raporty te, mające skomplikowaną strukturę danych, musiały zostać przepisane na uporządkowaną tablicę danych. Dokonano tego za pomocą skryptu Python. Skrypt ten w późniejszej fazie obliczył współczynniki aerodynamiczne. Przykładowe współczynniki przedstawiono w formie wykresu (po interpolacji) na Rys. 16.

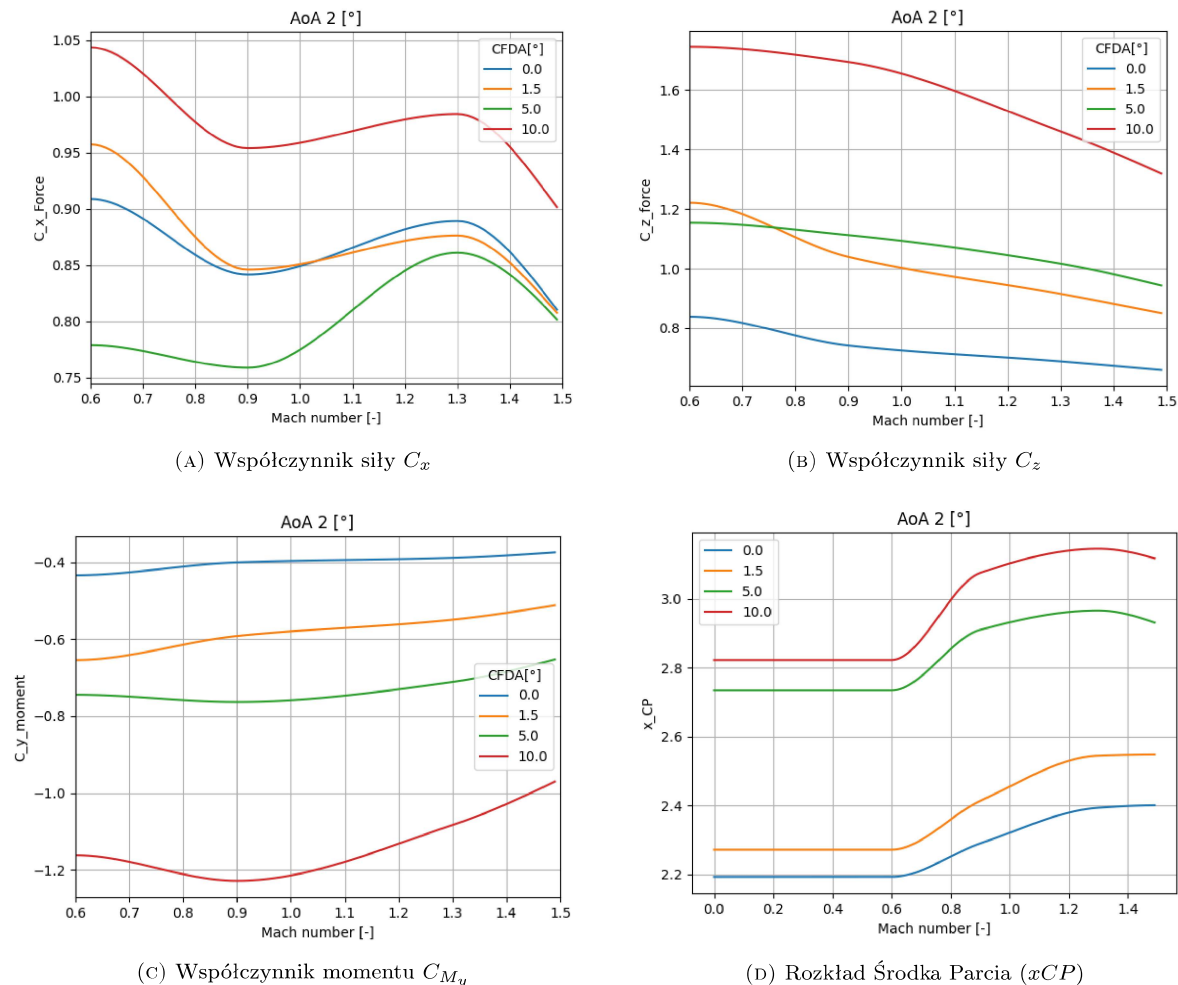


FIGURE 16: Wykresy interpolowanych wartości współczynników i rozkładu środka parcia w funkcji liczby Macha przy $AoA = 2.0^\circ$ [1].

Wykresy pokazują zmienne generowanie sił i momentów aerodynamicznych w zależności od liczby Macha. Współczynnik C_x zmienia się, wskazując na wpływ charakterystyki przepływu na generowane siły. W zakresie do Mach 1 wartość C_x maleje, a następnie po wejściu w reżim naddźwiękowy zaczyna rosnąć. Jest to związane ze zjawiskiem powstawania fali uderzeniowej w przepływach naddźwiękowych, podczas którego indukowany jest opór falowy. Następnie wartość C_x osiąga krytyczny poziom naddźwiękowy (około Mach 1.3), po czym zaczyna spadać.

Analizując te wyniki, należy wziąć pod uwagę, że do obliczenia współczynników ciśnienie dynamiczne jest używane w postaci stosowanej dla przepływów nieściśliwych. Implikuje to również nieliniowe skalowanie składowych charakterystycznych przepływu z liczbą Macha. W przypadku współczynników C_z i C_{M_y} można zauważyć, mając na uwadze skalowanie Macha, że mają one tendencję do zmniejszania się wraz ze wzrostem liczby Macha. Należy również zwrócić uwagę na zmianę rozkładu środka parcia na osi X (x_{CP}). Jego tendencja wzrostowa (co oznacza, że CP przesuwa się do tyłu, zwiększając w ten sposób stabilność pojazdu) w reżimie naddźwiękowym sugeruje, że C_z maleje szybciej niż współczynnik momentu C_{M_y} , co może sugerować, że nos rakiety traci C_z szybciej niż stateczniki ogonowe.

Zebranie tych danych spełnia założenia celu analizy. Uzyskano siły i momenty, obliczono współczynniki i wykonano charakterystyki aerodynamiczne. Dodatkowo, aby potwierdzić, że proponowana konfiguracja jest wystarczająca, uzyskane wartości zostały zaimplementowane do algorytmu sterowania i symulacji trajektorii rakiety oraz jej reakcji na sygnały wejściowe. Wstępne symulacje sterowania pokazują, że proponowana konfiguracja powierzchni aerodynamicznych generuje zadowalające siły i momenty, umożliwiając w ten sposób precyzyjne sterowanie i kontrolę trajektorii. Na Rys. 19 przedstawiono charakterystykę ESO (Extended State Observer) w funkcji czasu. ESO jest jednym z kontrolerów używanych w algorytmie sterowania ATC. Po lewej stronie przedstawiono wykres przedstawiający prędkość kątową przechyłu w funkcji czasu. Pokazuje on, że gdy rakieta napotka perturbację na swojej trajektorii lotu.

kontroler trajektorii jest w stanie przywrócić stabilność rakiety i przywrócić ją na pożądaną trajektorię, sprawiając, że prędkość kątowna przechyłu dąży do zera. Po prawej stronie przedstawiono wykres przedstawiający prędkości boczne v i w w funkcji czasu. Pokazuje on, że gdy rakieta napotyka perturbacje, a prędkości boczne zaczynają oscylować z powodu chwilowego braku stabilności dynamicznej, kontroler, wykorzystując zmienne CFDA, jest w stanie zmniejszyć częstotliwość oscylacji, a następnie zredukować je do wartości pomijalnych.

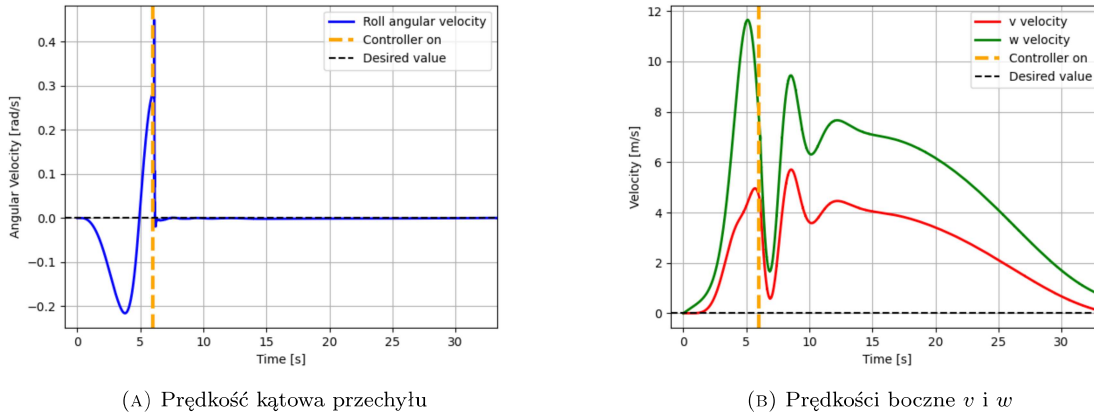


FIGURE 17: Przykład kontroli trajektorii lotu z użyciem kontrolera ESO [4]

3.2 Charakterystyka jet vanów

Charakterystyka sił nośnych i oporu jet vanów została przedstawiona na Figure 18. Proces projektowania obejmuje skrupulatną analizę z wykorzystaniem obliczeniowej dynamiki płynów (CFD) i sprzężonego transferu ciepła (CHT). Analizy te mają kluczowe znaczenie dla zrozumienia termicznego i aerodynamicznego zachowania łopatek odrzutowych w wysokotemperaturowych gazach spalinowych silników rakietowych. Analiza CFD pomaga w optymalizacji wydajności aerodynamicznej i możliwości wektorowania ciągu, podczas gdy analiza CHT koncentruje się na naprężeniach termicznych i odporności cieplnej materiału w ekstremalnych warunkach. To podwójne podejście zapewnia, że jet vany są zarówno skuteczne w kontrolowaniu trajektorii rakiety, jak i wystarczająco odporne, aby wytrzymać trudne warunki pracy.

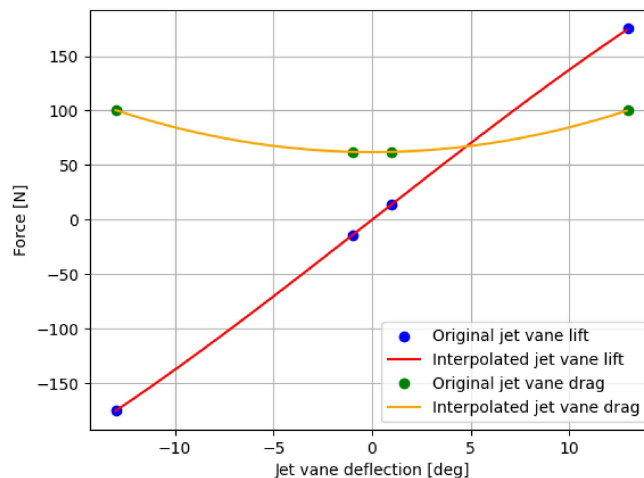


FIGURE 18: Interpolacja siły nośnej i oporu jet vanów [4].

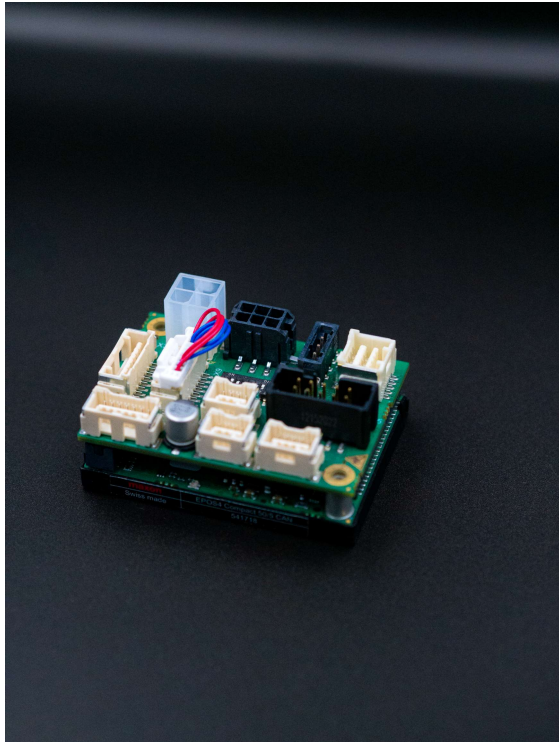
Układ testowy składa się z komponentów firmy Maxon, które razem tworzą precyzyjny system napędowy, przeznaczony do testowania i regulacji wychylenia powierzchni sterowych oraz jet vanów. Na jego skład wchodzi następujące elementy: napęd, enkoder, sterownik oraz przekładnia.

Napęd i przekładnia stanowią o przykładanej sile przez system. Zastosowana przekładnia ma średnicę zewnętrzną 32 mm i oferuje przełożenie 5.8:1, co zapewnia optymalne połączenie momentu obrotowego z prędkością napędu. Wraz z wrzecionem, wyposażonym w gwint metryczny, jest w stanie przenieść maksymalną siłę posuwu do 742 N (w trybie chwilowym) przy maksymalnej prędkości posuwu wynoszącej 23 mm/s. Dzięki precyzyjnej budowie przekładnia posiada ona minimalny luz wynoszący 0.7°, co zapewnia wyjątkową precyzję i dokładność pozycji na poziomie 0.033 mm. Mechaniczne obciążenie wrzeciona zostało zoptymalizowane pod kątem maksymalnych sił osiowych (2700 N) i promieniowych (200 N) w różnych punktach, co zapewnia dużą stabilność operacyjną w zmiennych warunkach.

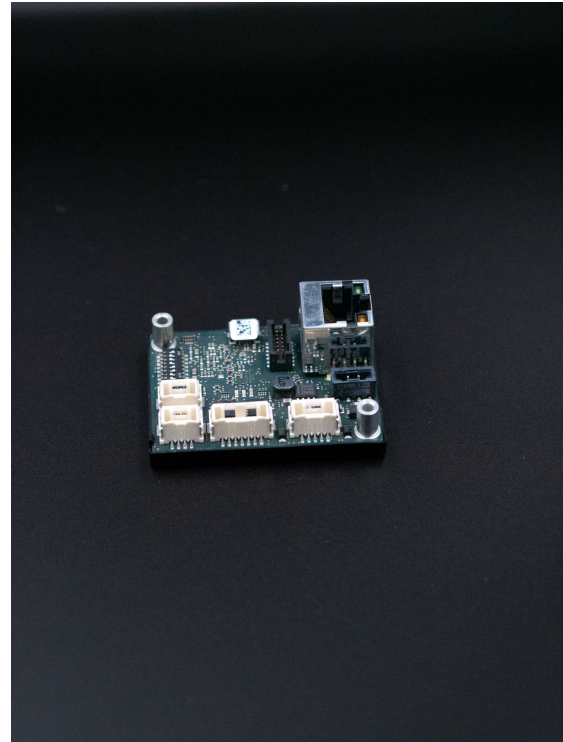
Silnik DC, który napędza system, charakteryzuje się nominalnym napięciem 12 V i zapewnia moc ciągłą wynoszącą 20 W. Silnik ten może osiągnąć maksymalną prędkość 14,000 obr./min i dostarcza moment obrotowy rzędu 27.5 mNm w trybie nominalnym, osiągając moment zatrzymania do 268 mNm przy maksymalnym prądzie zwarciovym 23.2 A. Silnik jest wyposażony w szczotki grafitowe, co zapewnia stabilną pracę w szerokim zakresie temperatur od -30°C do +100°C, oraz wysoką sprawność na poziomie 84%. Dzięki wykorzystaniu łożysk kulkowych, charakteryzuje się również minimalnym luzem promieniowym (0.025 mm), co dodatkowo podnosi precyzję i żywotność urządzenia.

Enkoder optyczny HEDL 5540 umożliwia dokładny pomiar pozycji i prędkości obrotowej. Dzięki rozdzielczości 500 impulsów na obrót i wyjściu zgodnemu ze standardem RS422, enkoder umożliwia precyzyjne śledzenie ruchu przy maksymalnej prędkości obrotowej 12,000 obr./min. Jest odporny na zmiany temperatury, działając w szerokim zakresie od -40°C do +100°C, co czyni go idealnym elementem do zastosowań w trudnych warunkach jakie występują przy silniku raketowym. Enkoder generuje sygnały w fazach 90°, co pozwala na precyzyjne śledzenie ruchu z minimalnym opóźnieniem, a jego waga wynosi tylko 27 g.

Sterownik EPOS4 Compact 50/5 CAN jest odpowiedzialny za kontrolowanie napędu i precyzyjne zarządzanie parametrami ruchu. Jego maksymalna wydajność prądowa wynosi 15 A (chwilowo) oraz 5 A (ciągła). Dzięki zaawansowanemu sterowaniu prędkością, momentem oraz pozycją, sterownik umożliwia precyzyjne osiągnięcie zadanych parametrów, a także integrację opisanym silnikiem DC. Wbudowana obsługa protokołu CANopen, a także interfejsów RS232 i USB, umożliwia łatwą komunikację z systemami zewnętrznymi, co czyni sterownik elastycznym narzędziem w konfiguracji systemu. Jego funkcje obejmują m.in. kontrolę momentu obrotowego, prędkości i pozycji, synchronizację oraz funkcję „homing” do automatycznego określania pozycji początkowej. Z kolei możliwość programowania i konfiguracji sterownika za pomocą oprogramowania EPOS Studio zapewnia łatwość w implementacji i optymalizacji sterowania. Dodatkowo użyto i skonfigurowano sterownik MicroMACS6, który umożliwia synchronizację sterowania czterema powierzchniami sterowymi oraz jet vaniami z poziomu jednego nadrzędnego kontrolera.



(A) Sterownik EPOS4 Compact



(B) Sterownik MicroMACS6

FIGURE 19: Sterowniki używane przy identyfikacji i sterowaniu

3.2.1 Badanie charakterystyki dynamicznej i mechanicznej silnika Maxon

Celem niniejszego badania jest wyznaczenie rzeczywistych parametrów silnika Maxon, w tym charakterystyki momentu obrotowego w zależności od natężenia prądu zasilającego. Wyznaczone parametry będą służyły do stworzenia dokładnego modelu silnika, który zostanie następnie uwzględniony w algorytmach sterujących. Pozwoli to na optymalizację sterowania silnikiem, zapewniając precyzyjne i efektywne działanie.

Opis platformy pomiarowej

- Silnik Maxon to napęd, który odpowiada za wprowadzenie w ruch mechanizmu śrubowego. Jest sterowany przez sterownik EPOS 4, który reguluje natężenie prądu w oparciu o zadany moment obrotowy, zapewniając precyzyjną i wydajną pracę systemu.
- Ruch obrotowy silnika Maxon jest przekazywany na śrubę M10x1mm, powodując ruch wózka. Dzięki temu możliwe jest obliczenie ruchu liniowego na podstawie odległości, o jaką przesunął się wózek.
- Sprężyna: Używana do generowania oporu w układzie, pozwala na analizę momentu obrotowego oraz wpływu obciążenia na zachowanie silnika.
- Układ sterowania – EPOS 4: Sterownik EPOS 4 od firmy Maxon odpowiada za precyzyjne sterowanie silnikiem, wymuszając odpowiednie natężenie prądu na podstawie zadanego momentu obrotowego. Sterownik zapewnia także monitorowanie wszystkich kluczowych parametrów pracy silnika, takich jak prędkość obrotowa i moment obrotowy, co pozwala na dokładną regulację warunków eksperymentu.
- Układ pomiarowy Enkoder: Używany do pomiaru prędkości obrotowej silnika, co pozwala na określenie, jak szybko obraca się wałek. Zmiana prędkości obrotowej silnika przekłada się na przesunięcie wałka w mechanizmie śrubowym.
Pomiar odległości: Na podstawie przesunięcia miedzianego wałka na śrubie obliczany jest ruch



FIGURE 20: Układ fizyczny z sprężyną

Identyfikacja parametrów fizycznych silnika

Do identyfikacji parametrów fizycznych silnika Maxon wykorzystano oprogramowanie dostarczone przez producenta EPOS Studio, które pozwala na szczegółowe testowanie i strojenie układu napędowego. Proces rozpoczął się od wprowadzenia podstawowych parametrów silnika, zgodnych z dokumentacją techniczną. Ustawiono takie wartości jak maksymalne przyspieszenie, maksymalna prędkość oraz maksymalny pobór prądu, co umożliwiło dopasowanie sterownika do specyfikacji urządzenia.

Kolejnym krokiem była identyfikacja układu elektrycznego, przeprowadzona w sekcji Tuning w trybie Current. W tej części badania podawano sygnały sterujące w postaci impulsów o określonych parametrach, analizując reakcję układu elektrycznego na te sygnały. Obserwacje te pozwoliły ocenić dynamikę systemu i jego odpowiedź na zmiany w sterowaniu, co stanowiło podstawę do dalszego strojenia.

Następnie przeprowadzono testy układu mechanicznego, wykorzystując tryby Tuning w sekcjach Velocity i Position. W trybie Velocity analizowano reakcję silnika na zmiany sygnałów sterujących prędkością, natomiast w trybie Position skupiono się na dokładności pozycjonowania. Podobnie jak w przypadku układu elektrycznego, testy polegały na podawaniu sygnałów wejściowych i sprawdzaniu, jak silnik reaguje na zadane wartości.

Reakcje układu mechanicznego na testowe sygnały sterujące pozwoliły na precyzyjną ocenę właściwości systemu, takich jak płynność regulacji prędkości czy dokładność pozycjonowania. Uzyskane dane umożliwiły optymalizację pracy całego układu napędowego, zapewniając jego wysoką wydajność i precyzję.

3.2.2 Sterowanie silnikiem przez UART: Implementacja wrappera w Pythonie

Do łatwej implementacji sterowania silnikiem w rozważanej aplikacji, napisany został interfejs pośredni między programem hardware in the loop, a dedykowaną biblioteką EPOS Command Library w języku Python. Wrapper został napisany metodą obiektową, zatem aby zdefiniować sterownik wykorzystujemy konstruktor.

```

1 class Maxon:
2     def __init__(self, node_id, baudrate, timeout):
3         self.path = '.\EposCmd64.dll'
4         self.epos = CDLL(self.path)
5         self.error_code = c_uint()
6         self.node_id = node_id
7         self.baudrate = baudrate
8         self.timeout = timeout
9         self.acceleration = 30000
10        self.deceleration = 30000
11        self.torque_coefficient= 12.008 * 10 ** (-3)
12        self.key_handle = self.epos.VCS_OpenDevice(b'EPOS4', b'MAXON SERIAL V2', b'USB',
13            ↪ b'USB0', byref(self.error_code))
14        self.epos.VCS_SetProtocolStackSettings(self.key_handle, baudrate, timeout,
15            ↪ byref(self.error_code))
16        self.epos.VCS_ClearFault(self.key_handle, self.node_id, byref(self.error_code))
17        self.epos.VCS_ActivateProfilePositionMode(self.key_handle, self.node_id,
18            ↪ byref(self.error_code))
19        self.epos.VCS_SetEnableState(self.key_handle, self.node_id, byref(self.error_code))
20        print('Motor active')

```

LISTING 3.1: Konstruktor klasy Maxon

Moment obrotowy jest wprost proporcjonalny do prądu na zaciskach zasilających.

$$T = C_T \cdot I, \quad (1)$$

gdzie,

T - moment obrotowy [Nm],

C_T - współczynnik momentowy [$\frac{Nm}{A}$],

I - prąd [A].

Aby zrealizować zadanie sterowania momentem przez wymuszenie prądowe należy przekształcić powyższy wzór.

$$I = \frac{T}{C_T} \quad (2)$$

Po zmianie jednostek i implementacji otrzymujemy metodę **force_torque**

```

1  def force_torque(self, torque):
2      current = int(torque / self.torque_coefficient* 1000)
3      self.epos.VCS_ActivateCurrentMode(self.key_handle, self.node_id, byref(self.error_code))
4      self.epos.VCS_SetCurrentMust(self.key_handle, self.node_id, c_short(current),
      ↪ byref(self.error_code))

```

LISTING 3.2: Metoda wymuszająca moment obrotowy

Aby zakończyć pracę z silnikiem należy wykorzystać metodę **disable_device**. Bez poprawnego zamknięcia komunikacji sterownik jest cały czas w trybie pracy co utrudnia ponowne uruchomienie programu.

```

1  def disable_device(self):
2      self.epos.VCS_SetDisableState(self.key_handle, self.node_id, byref(self.error_code))
3      self.epos.VCS_CloseDevice(self.key_handle, byref(self.error_{}code))

```

LISTING 3.3: Metoda zamykająca komunikację

Silnik dysponuje różnymi trybami pracy. Na potrzeby projektu zostały wykorzystane tylko dwa. W celu manewrowania między nimi zaimplementowano dodatkową strukturę **op_mode** i metodę **set_operation_mode**

```

1  class op_mode(Enum):
2      profile_position_mode = 1
3      current_mode = 10

```

```

1  def set_operation_mode(self, mode):
2      match mode.name:
3          case 'profile_position_mode':
4              set_mode = c_uint8(op_mode.profile_position_mode.value)
5          case 'current_mode':
6              set_mode = c_uint8(op_mode.current_mode.value)
7          case _:
8              print('Invalid mode!')
9              return 0
10     self.epos.VCS_SetOperationMode(self.key_handle, self.node_id, set_mode,
      ↪ byref(self.error_code))

```

LISTING 3.5: Metoda do zmiany trybu pracy

Przedstawiony powyżej kod jest wystarczający do odpowiedniego sterowania silnikiem w rozważanej aplikacji. W trakcie rozwijania projektu zostały wprowadzone dodatkowe funkcjonalności w celu testowania poprawności działania kodu.

Metody `profile_position_move` i `get_position` zostały wprowadzone jako pierwsze, aby sprawdzić podstawowe funkcje biblioteki producenta, natomiast `get_operation_mode` i `get_current` powstały w następnych iteracjach programu.

```

1  def profile_position_move(self, target_position, target_speed):
2      while True:
3          if target_speed != 0:
4              self.epos.VCS_SetPositionProfile(self.key_handle, self.node_id, target_speed,
5                  ↪ self.acceleration, self.deceleration, byref(self.error_code))
6              self.epos.VCS_MoveToPosition(self.key_handle, self.node_id, target_position,
7                  ↪ True, True, byref(self.error_code))
8          elif target_speed == 0:
9              self.epos.VCS_HaltPositionMovement(self.key_handle, self.node_id,
10                 ↪ byref(self.error_code))
11             true_position = self.get_position()
12             if true_position == target_position:
13                 break
14
15 def get_position(self):
16     position = c_long()
17     self.error_code = c_uint()
18     ret = self.epos.VCS_GetPositionIs(self.key_handle, self.node_id, byref(position),
19         ↪ byref(self.error_code))
20     return position.value
21
22 def get_operation_mode(self):
23     p_mode = c_uint8()
24     self.error_code = c_uint()
25     ret = self.epos.VCS_GetOperationMode(self.key_handle, self.node_id, byref(p_mode),
26         ↪ byref(self.error_code))
27     return p_mode.value
28
29 def get_current(self):
30     current = c_short()
31     self.epos.VCS_GetCurrentIs(self.key_handle, self.node_id, byref(current),
32         ↪ byref(self.error_code))
33     return current.value

```

LISTING 3.6: Dodatkowe metody

4 Algorytmy sterowania

4.1 Projekt

Aby móc kontrolować tor lotu rakiety niezbędne było zaprojektowanie algorytmów sterowania. Rozważanym systemem jest rakieta posiadająca aerodynamiczne powierzchnie sterujące oraz możliwość wektorowania ciągu za pomocą powierzchni zwanych jet vanami. W poniższych rozważaniach przyjęto, że rakieta wykorzystuje sprzężone działanie obu rodzaj powierzchni sterowych z przełożeniem jednostkowym. Obiekt sterowania jest traktowany jako model nieliniowy o wysokiej niepewności parametrycznej spowodowanej

wypalającym się silnikiem oraz zmieniającymi się zależnościami aerodynamicznymi:

$$\begin{cases} \dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \otimes \mathbf{q} \\ \dot{\boldsymbol{\omega}} = \mathbf{J}^{-1} [\mathbf{N} - \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega})] \end{cases} \quad (3)$$

gdzie q oznacza kwaternion opisujący orientację, $\boldsymbol{\omega}$ opisuje prędkość rotacji dla poszczególnych osi, \mathbf{J} opisuje macierz momentu bezwładności oraz \mathbf{N} jest to wektor momentów generowanych przez powierzchnie sterujące.

Chcąc kontrolować tor lotu rakiety, za cel ustalono sterowanie kątem odchylenia osi podłużnej od pozycji pionowej. Zadanie sterowania zostaje uznane za spełnione, gdy błąd odchylenia mieści się w zakresie $0 - 2^\circ$. Czas stabilizacji wynosi $T_k \leq 2$ s, gdzie T_k oznacza czas potrzebny do osiągnięcia wartości referencyjnej. Przyjęto również, że maksymalna wartością dla wybranych powierzchni sterowych jest $\pm 10^\circ$.

4.2 Implementacja

Schemat ideowy zaimplementowanego algorytmu sterowania został przedstawiony na rysunku 21.

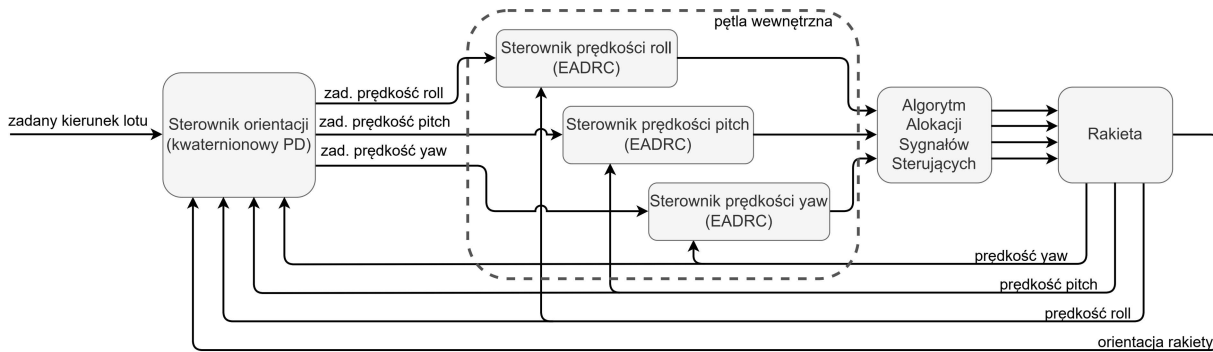


FIGURE 21: Schemat ideowy zaimplementowanego sterownika

Został on zaprojektowany w formie sterownika kaskadowego. W pętli zewnętrznej znajduje się nieliniowy kwaternionowy sterownik proporcjonalno-różniczkujący (PD). Na jego wejście zadawany jest zadany kierunek lotu rakiety wyrażony w formie wektora jednostkowego w układzie globalnym, a także aktualna orientacja rakiety oraz prędkości kątowe wyrażone w jej lokalnym układzie współrzędnych. Sygnałami sterującymi pętli zewnętrznej są referencyjne prędkości kątowe, które zadawane są na pętlę wewnętrzną sterownika. Pętla wewnętrzna składa się z 3 sterowników ADRC (ang. *Active Disturbance Rejection Control*) w wersji uchybowej (EADRC), po jednym na każdą oś układu współrzędnych. Poza zadanymi prędkościami kątowymi na ich wejście podawane są także aktualne wartości prędkości kątowych rakiety. Sygnały sterujące generowane przez pętlę wewnętrzną podawane są blok realizujący alokację sygnałów sterujących na poszczególne powierzchnie sterowe rakiety.

Takie podejście polegające na konwersji orientacji na prędkości kątovej znacznie upraszcza syntezę algorytmu sterowania poprzez usunięcie nieliniowych zależności kwaternionów z pętli wewnętrznej sterownika. Jednocześnie możliwe jest pełne wykorzystanie zalet orientacji wyrażonej w kwaternionach takich jak brak niejednoznacznych orientacji. Dodatkowo wykorzystanie sterowników EADRC zapewnia odporność na zaburzenia.

4.3 Testy

Testy zostały przeprowadzone symulacyjnie. Rozważono dwa scenariusze. Pierwszy z nich zakłada start rakiety pod kątem 45° . Drugi natomiast sprawdza działanie sterowania w utrzymywaniu 10° wychylenia.

4.3.1 Scenariusz 1

Błąd wychylenia osi podłużnej rakiety od referencyjnego kierunku dla testu pierwszego przedstawiony jest na rys. 22. Można zauważyć, że akceptowalny zakres błędu wychylenia osiągnany jest po 1.53s.

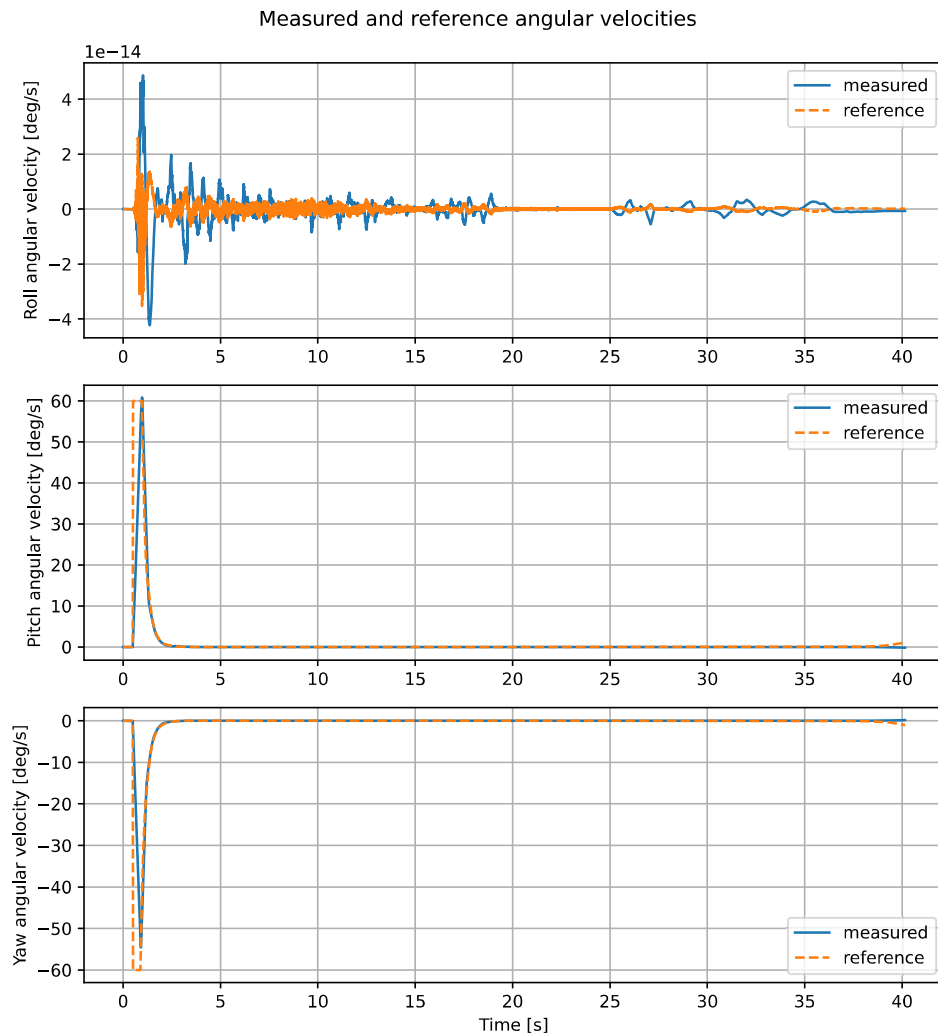


FIGURE 22: Wykres błędu kąta osi podłużnej rakiety od referencyjnego kierunku dla scenariusza 1. Czerwona linia przerywana oznacza zakres akceptacji, natomiast zielona linia przerywana czas akceptacji.

Sterowania podawane na powierzchnie sterowe przedstawione są na rys. 23, natomiast prędkości kątowe na rys. 25. Można zauważyć, że wewnętrzna pętla sterownika kaskadowego EADRC odtwarza referencyjne prędkości kątowe. Na rys. 26 przedstawiona jest wynikowa trajektoria rakiety. Widoczne jest, że rakieta nie zachowuje się nieprzewidywalnie.

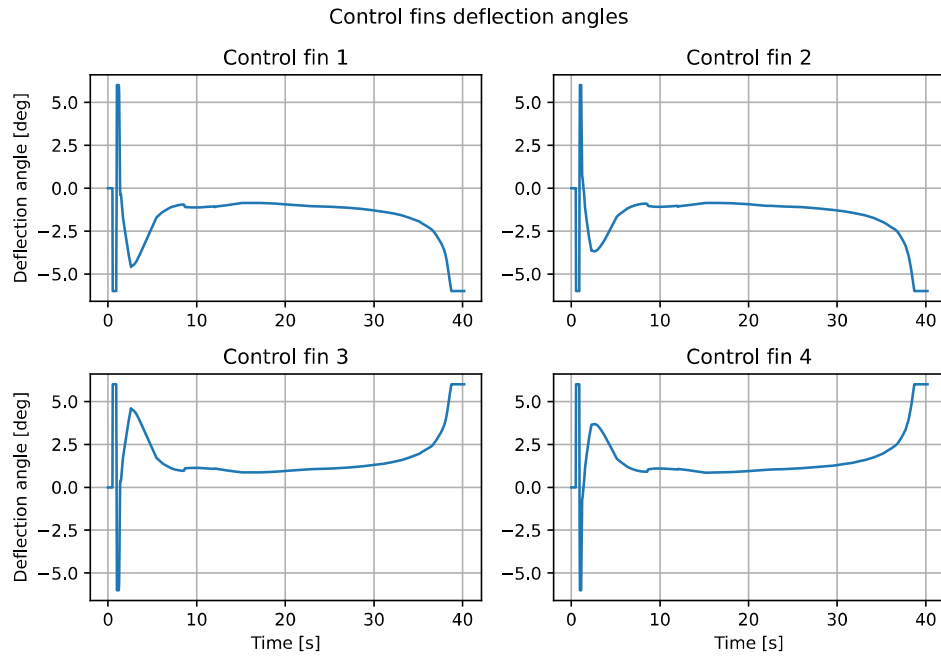


FIGURE 23: Wykresy wychyleń powierzchni sterowych dla scenariusza 1.

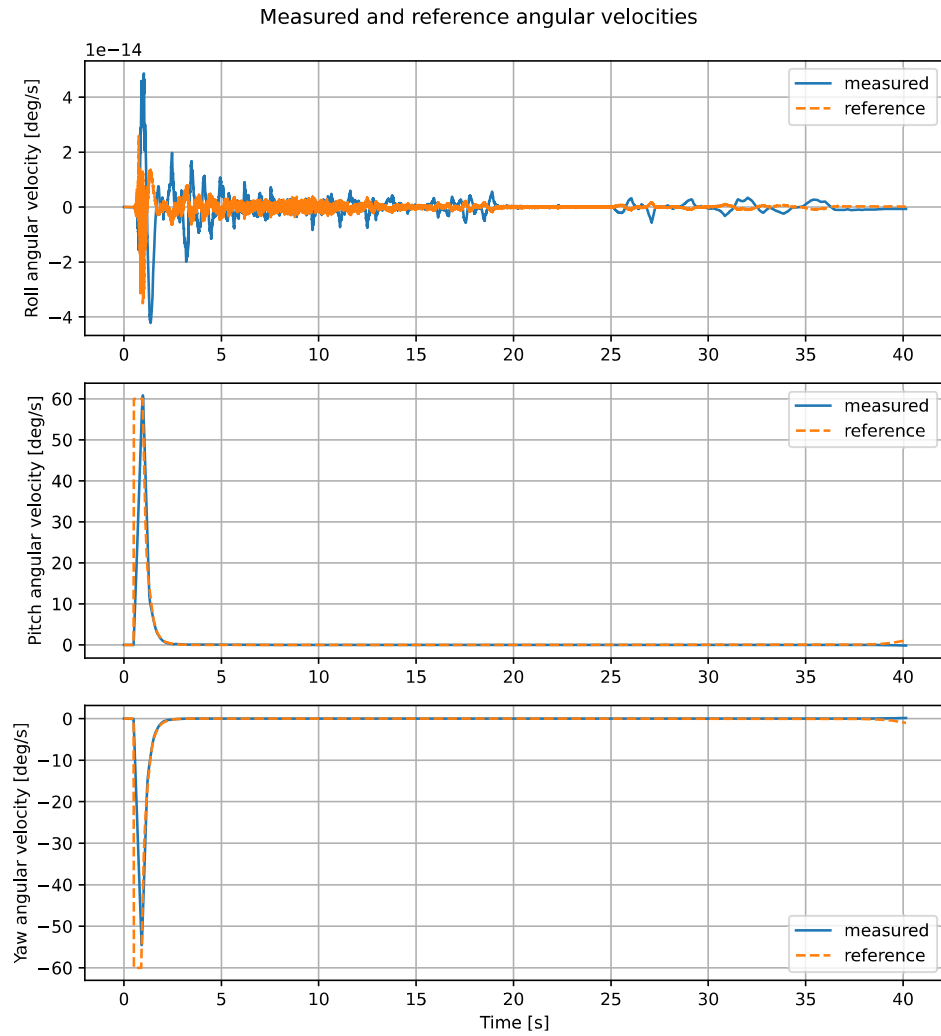


FIGURE 24: Prędkości kątowe dla scenariusza 1.

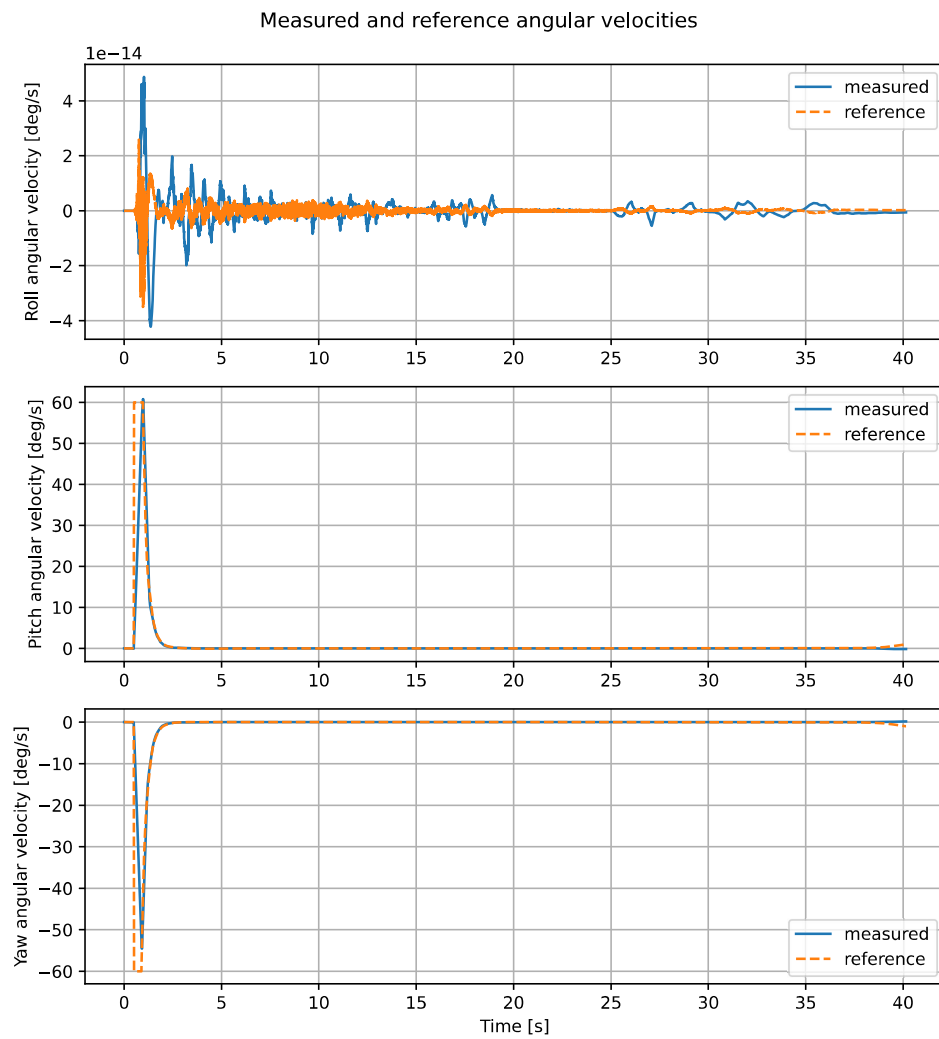


FIGURE 25: Prędkości kątowe dla scenariusza 1.

Position of the rocket during flight

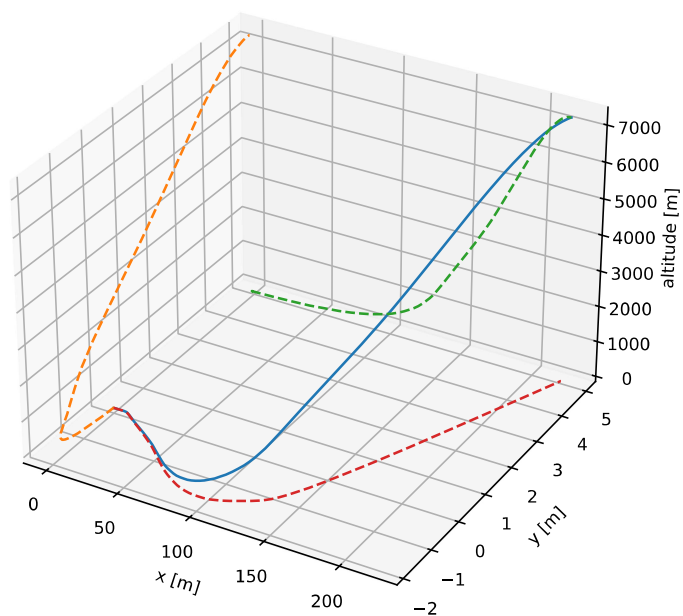


FIGURE 26: Pozycja rakiety dla scenariusza 1. Przerwane linie oznaczają projekcję trajektorii rakiety na poszczególne płaszczyzny.

4.3.2 Scenariusz 2

Błąd wychylenia dla scenariusza drugiego przedstawiony jest na rys. 27. Widoczne jest, że rakieta osiąga żądany kierunek. Przed 30 sekundą rakieta zaczyna stopniowo tracić kierunek. Na wykresach wychyleń powierzchni sterowych widocznych na rys. 28, można zauważyć, że sygnały sterujące się nasycają. Oznacza to, że prędkość rakiety jest zbyt mała, by wygenerować pożądaný moment aerodynamiczny, żeby utrzymać raketę w wybranym kierunku. Rakieta również nie generuje już ciągu w tym momencie co też nie pozwala wykorzystać powierzchni używanych do wektorowania ciągu.

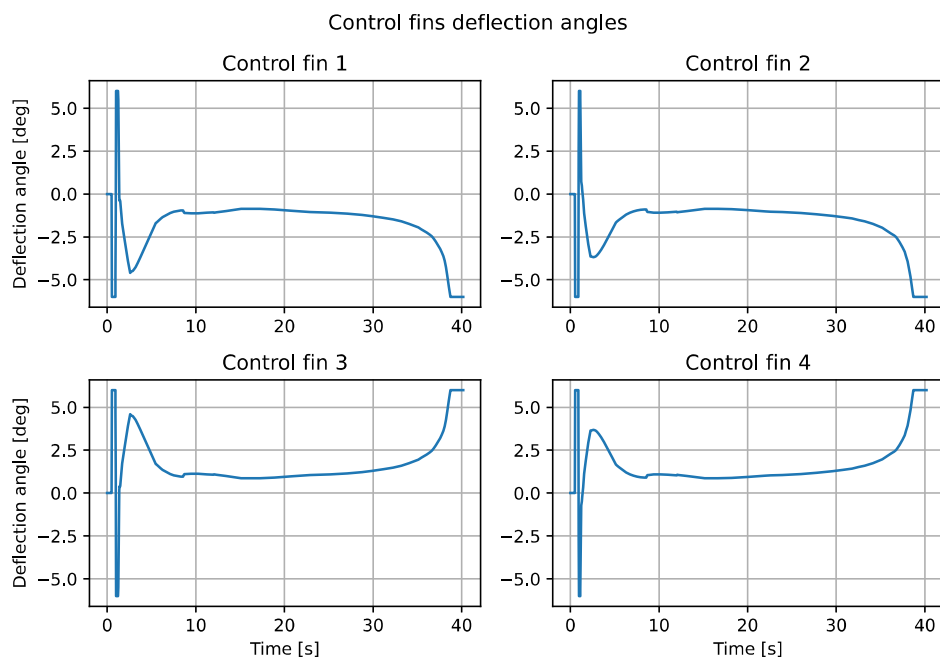


FIGURE 27: Wykres błędu kąta osi podłużnej rakiety od referencyjnego kierunku dla scenariusza 2. Czerwona linia przerywana oznacza zakres akceptacji, natomiast zielona linia przerywana czas akceptacji.

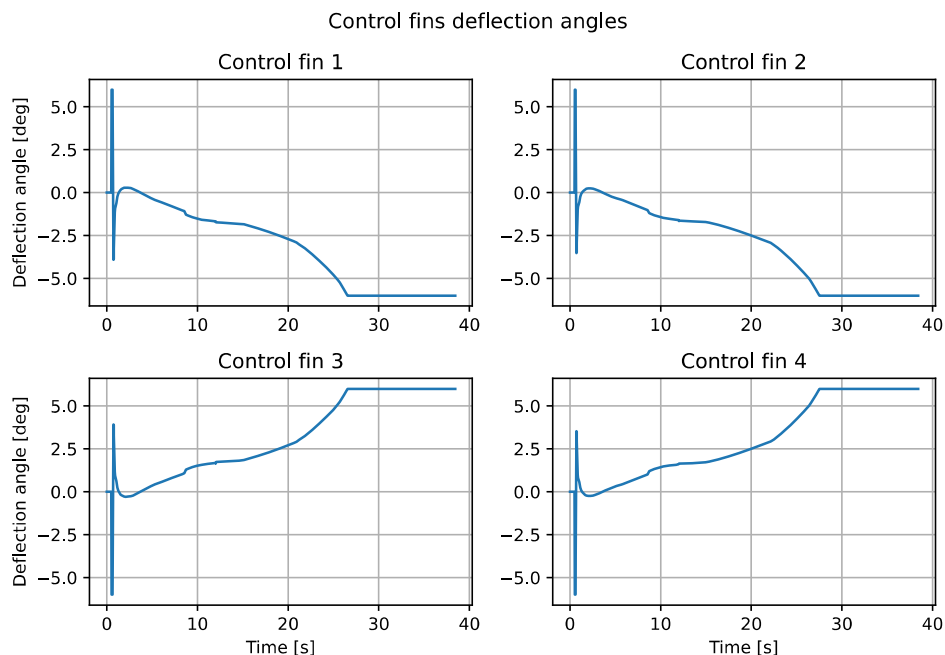


FIGURE 28: Wykresy wychyleń powierzchni sterowych dla scenariusza 2.

Wykresy prędkości rotacji przedstawione są na rys. 29. Sterowniki w pętli wewnętrznej skutecznie śledzą zadane referencyjne wartości prędkości kątowych. Na rys. 30 przedstawiona jest trajektoria rakiety podczas tego testu

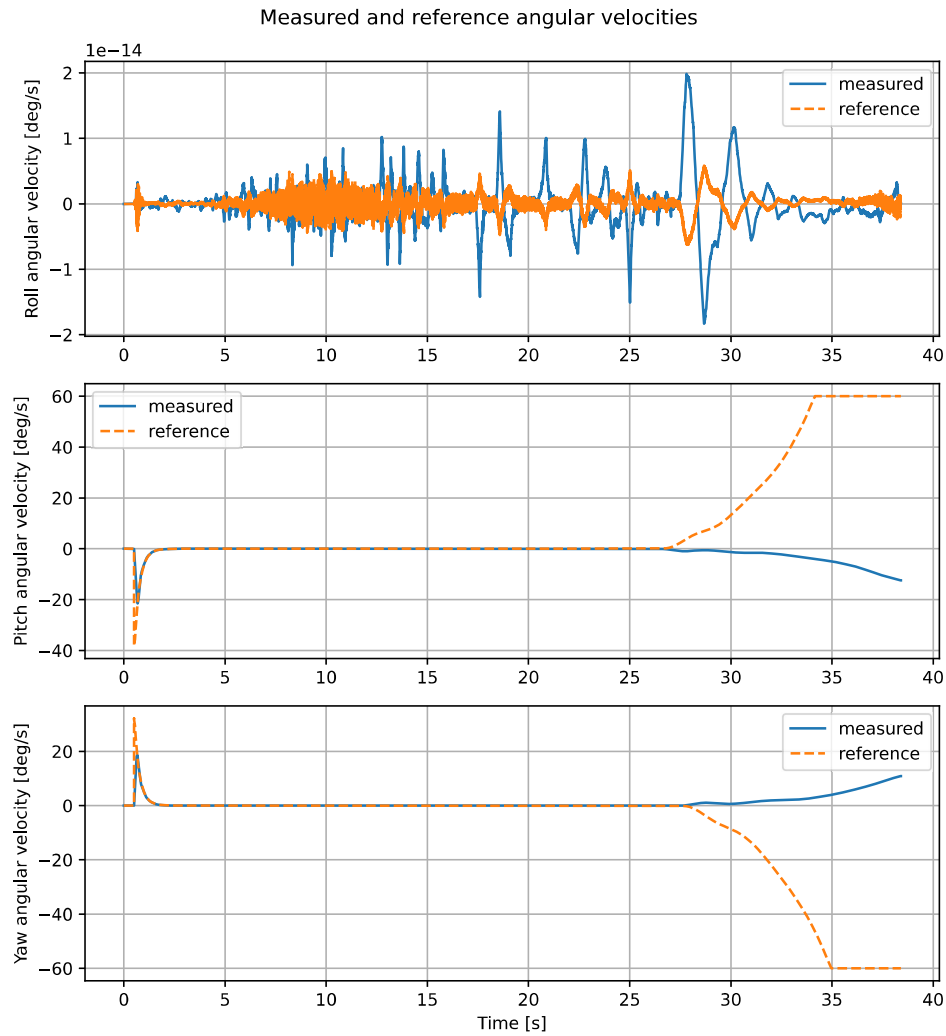


FIGURE 29: Prędkości kątowe dla scenariusza 2.

Position of the rocket during flight

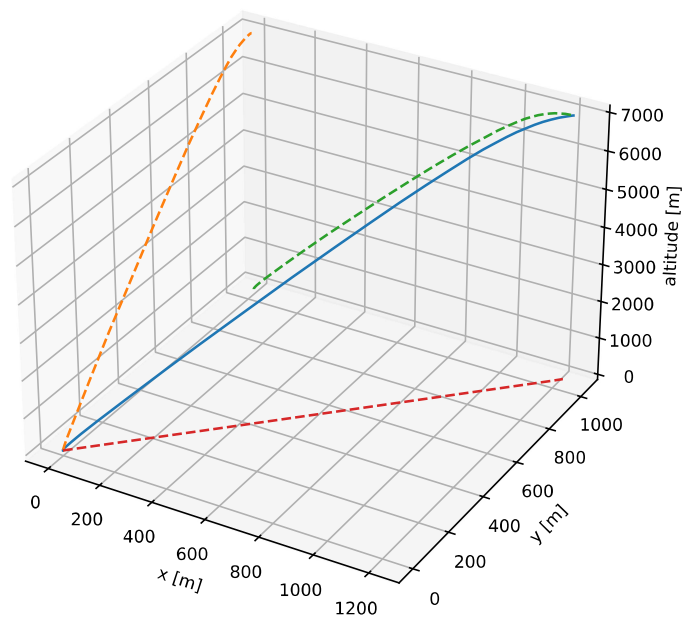


FIGURE 30: Pozycja rakiety dla scenariusza 2. Przerwane linie oznaczają projekcję trajektorii rakiety na poszczególne płaszczyzny.

5 Metryka koszt/manewrowość

5.1 Opracowanie

Ze względu na wybrany zestaw aktuatorów, tj. Aby dokładnie określić zakres dostępnych sił i momentów kluczowych dla kontroli trajektorii rakiety, wykorzystano napisany symulator i stworzono w nim dedykowane funkcje, pozwalające estymować parametry układów. Symulacje te koncentrowały się na weryfikacji transferu danych dotyczących sił i momentów z programu Fluent do symulatora. Po wyeksportowaniu tych wartości do specyficznych plików .txt, proces konwersji zapewnił ich poprawne przeskalowanie do bezwymiarowej reprezentacji zgodnej z modelem rakiety w symulatorze, co pozwoliło na realistyczne odwzorowanie jej charakterystyki. Kluczowym aspektem tej weryfikacji było określenie, jak różne kąty wychylenia sterów oraz łopatek dyszowych, zarówno osobno, jak i w kombinacji, wpływają na osiągi rakiety i jej zdolność do wykonywania manewrów. Ustalenie takich charakterystyk jest niezbędne do określenia granicznych możliwości układu sterowania w generowaniu sił podczas lotu.

Jet vany są operacyjne jedynie przez pierwszą fazę lotu, ponieważ ich zdolność do generowania momentów i sił bocznych zależy od tego czy silnik pracuje. Z kolei stery aerodynamiczne są efektywne przez cały lot, jednak ich skuteczność maleje wraz ze spadkiem liczby Macha. Spadek ten ma charakter kwadratowy względem prędkości rakiety oraz liniowy względem gęstości atmosfery.

W związku z tym charakterystyki sił i momentów generowanych przez mechanizmy sterujące należy podzielić na dwa odrębne etapy: pierwszy, gdy silnik jest aktywny, i drugi, gdy rakieta nie jest już napędzana. Ponadto skuteczność tych mechanizmów zależy od wielu czynników, takich jak liczba Macha, kąt natarcia i ześlizgu rakiety oraz specyficzne wychylenia poszczególnych elementów sterujących. W rezultacie konieczne jest zdefiniowanie dwóch zestawów funkcji. Pierwszy dotyczy fazy aktywnego napędu, uwzględniając złożoną interakcję sił aerodynamicznych i elementów sterujących. Opracowanie tych funkcji jest kluczowe dla zrozumienia zachowania rakiety w fazie napędzanej, umożliwiając precyzyjną kontrolę i optymalizację trajektorii.

W celu wyjaśnienia zależności między tymi czynnikami wykorzystano symulator. Dla uproszczenia wyników model wykluczył efekty kąta ześlizgu, co jest uzasadnione obserwacją, że zmiany sił i momentów przy zmianie kąta ześlizgu są analogiczne do zmian przy zmianie kąta natarcia. Pozwoliło to na prostsze określenie i analizę wyników sił i momentów.

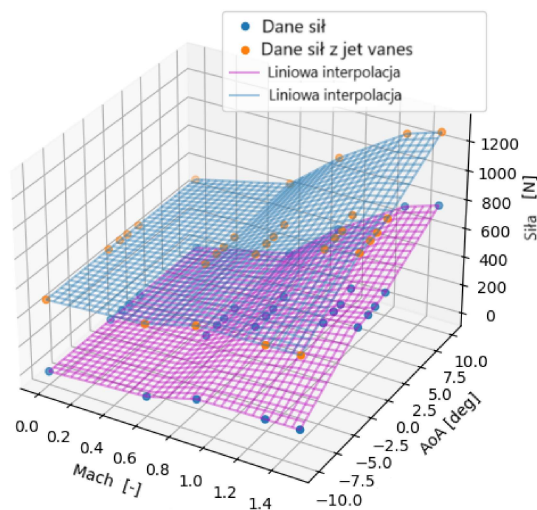


FIGURE 31: Maksymalne siły generowane przez mechanizm sterowania [4].

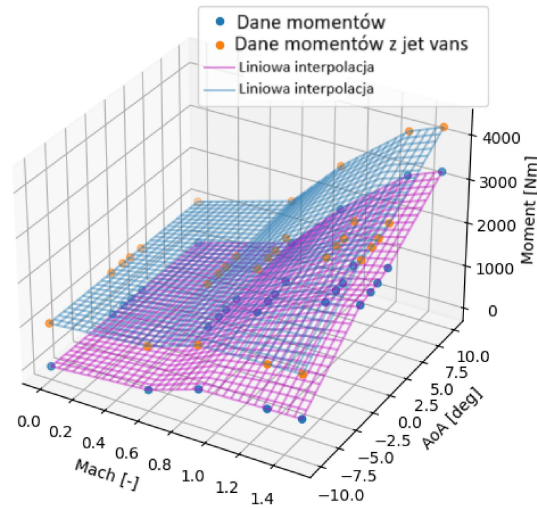


FIGURE 32: Maksymalne momenty generowane przez mechanizm sterowania [4].

Aby ocenić efektywność systemu sterowania rakiety pod względem zarówno generowanych sił i momentów, jak i kosztów ich uzyskania, wprowadzono metrykę koszt/manewrowość. Metryka ta umożliwia ilościowe określenie efektywności mechanizmów sterujących przy uwzględnieniu ograniczeń budżetowych oraz zasobów, takich jak masa, zasilanie i objętość rakiety.

Metrykę koszt/manewrowość zdefiniowano jako stosunek zdolności systemu sterowania do generowania sił i momentów do jego kosztu zasobowego. W uogólnionej formie, metrykę można wyrazić za pomocą wzoru:

$$\mu_{KM} = \frac{C_{manewrowo}}{C_{zasoby}}, \quad (4)$$

gdzie:

- μ_{KM} - wartość metryki koszt/manewrowość
- $C_{manewrowo}$ - zdolność do generowania sił i momentów, definiowana jako suma maksymalnych sił i momentów generowanych przez mechanizmy sterujące:

$$C_{manewrowo} = \sum_i (F_i^{max} + M_i^{max}), \quad (5)$$

- C_{zasoby} - koszt zasobowy, uwzględniający masę (m), zużycie energii (P) oraz zajmowaną objętość (V):

$$C_{zasoby} = \alpha m + \beta P^\phi + \gamma V^\psi, \quad (6)$$

gdzie

- α, β, γ są współczynnikami wagowymi odzwierciedlającymi znaczenie poszczególnych zasobów w kontekście misji,
- ϕ 1 współczynnik penalizacji dla wysokiej mocy,
- ψ 1 współczynnik penalizacji dla objętości.

Masa i moc mają równe znaczenie, ponieważ wpływają na stabilność i efektywność systemu, dla przykładowej analizy zostały przyjęte wartości 0.4. Objętość ma mniejsze znaczenie, ale jest uwzględniana dla kompatybilności z geometrią rakiety, dla przykładowej analizy została przyjęta wartość 0.2. Penalizacja mocy (ϕ) jest istotna, ponieważ wysokie wartości mocy mogą znacząco obciążać system, dla przykładowej analizy została przyjęta wartość 1.5. Penalizacja objętości (ψ) jest silniejsza, aby uwzględnić ograniczenia przestrzenne, dla przykładowej analizy została przyjęta wartość 1.8.

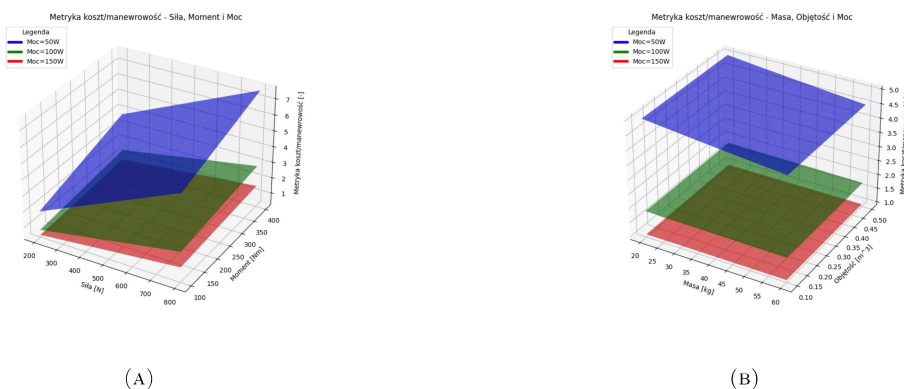


FIGURE 33: Wykresy obrazujące zależności pomiędzy parametrami metryki. Dla przyjętych wartości współczynników: $\alpha = 0.4, \beta = 0.4, \gamma = 0.2, \phi = 1.5, \psi = 1.8$.

Ze względu na różne fazy lotu rakiety, metryka jest liczona osobno dla fazy aktywnego napędu oraz fazy beznapędowej. Wartości maksymalnych sił i momentów oraz koszty zasobowe są estymowane na podstawie wyników symulacji, uwzględniających rodzaj aktuatorów (np. jet vany, stery aerodynamiczne), charakterystyki aerodynamiczne rakiety, oraz ograniczenia wynikające z masy, objętości oraz dostępnej energii.

Wynik metryki μ_{KM} stanowi miarę efektywności systemu sterowania przy zadanych ograniczeniach zasobowych. Wyższa wartość metryki wskazuje na lepsze wykorzystanie zasobów do generowania sił i momentów, co może wpływać na wybór optymalnego projektu mechanizmów sterujących dla danej misji.

Aby uzyskane metryki koszt/manewrowość mogły być użyteczne w kontekście ekonomicznym, wprowadzono ich przeliczenie na wartości finansowe. Skalowanie to realizowane jest przy wykorzystaniu logarytmicznej funkcji przeliczeniowej, co pozwala na modelowanie zmienności kosztów lub oszczędności w sposób nieliniowy. Wartość pieniężna ($W_{ekonomiczna}$) obliczana jest zgodnie z poniższym wzorem:

$$W_{ekonomiczna} = C_{metryka} * \log(1 + \alpha \mu_{KM}), \quad (7)$$

gdzie:

- $C_{metryka}$ - przelicznik wartości podstawowej metryki, wyrażony w PLN/jednostkę,
- α - współczynnik wpływu wartości metryki na wynik finansowy,
- μ_{KM} - wartość metryki koszt/manewrowość.

Zastosowanie logarytmicznego skalowania pozwala uwzględnić malejące przyrosty korzyści finansowych przy bardzo wysokich wartościach metryki, co jest realistycznym odwzorowaniem rzeczywistości. Przyjęcie współczynnika α i przelicznika $C_{metryka}$ zależy od specyficznych wymagań misji oraz ograniczeń budżetowych. Przykładowo, dla $C_{metryka} = 500$ PLN/jednostkę oraz $\alpha = 2$, uzyskane wartości metryki są skalowane do realiów finansowych projektu.

Dalsze analizy umożliwiają ocenę efektywności kosztowej systemów sterowania w kontekście wymagań misji i dostępnego budżetu. Wprowadzenie skalowania finansowego jest szczególnie istotne w procesie projektowania, gdzie kluczowe jest zrównoważenie efektywności technicznej i kosztowej

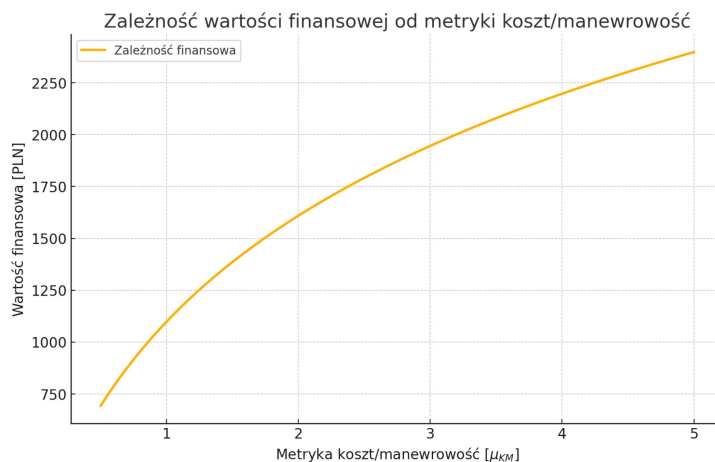


FIGURE 34: Wykres przedstawia zależność wartości finansowej od metryki koszt/manewrowość

6 Rezultaty projektu

6.1 Stanowisko Hardware in the Loop

Stanowisko testowe typu Hardware-in-the-Loop (HIL) zostało zaprojektowane w celu przeprowadzenia badań mechanizmu sterowania za pomocą sterów strumieniowych (jet vanes) oraz powierzchni sterowych. Głównym założeniem jest ocena poprawności działania tego mechanizmu oraz algorytmu sterowania poprzez symulację rzeczywistych warunków pracy. Testy realizowane na stanowisku umożliwiają:

- wymuszenie oporu za pomocą silnika prądu stałego (DC), ma on symulować moment generowany przez opór aerodynamiczny na powierzchniach sterowych
- generowanie momentu przy użyciu silnika Maxon
- podłączenie układu pozwalającego na chwilową symulację warunków przepływowych w dyszy

Konstrukcja stanowiska opiera się na aluminiowych profilach konstrukcyjnych, ułożonych w formie klatki, zapewniającej stabilność całego układu. Mechanizm sterowania został zamontowany na boku konstrukcji, co umożliwi precyzyjne dostosowanie jego położenia do wymogów testowych.

Taka konstrukcja pozwala na łatwą modyfikację położenia względnych pomiędzy komponentami, umożliwiając jej dostosowanie do różnych konfiguracji testowych oraz wszechstronne zastosowanie stanowiska w badaniach.

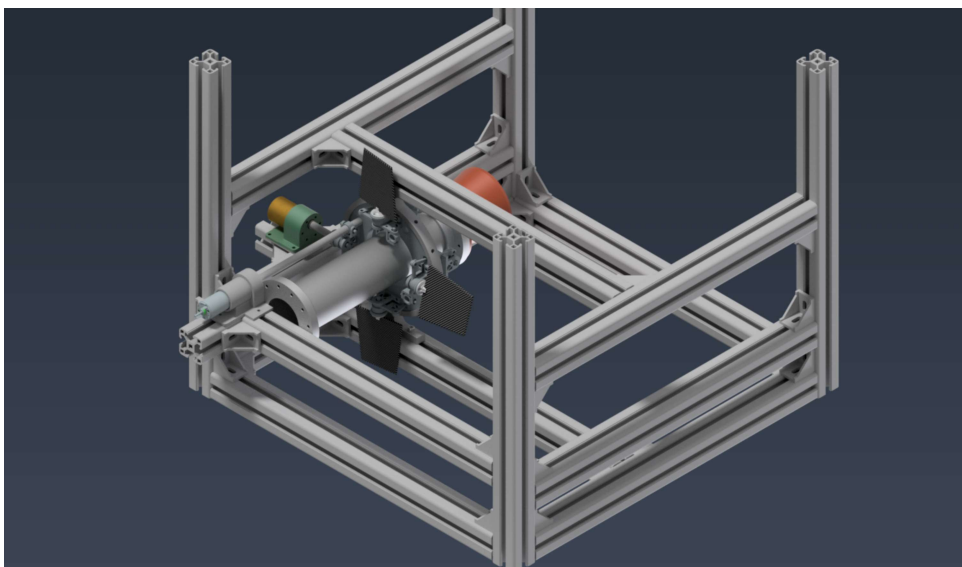


FIGURE 35: Zdjęcie przedstawiające stanowisko Hardware in the loop

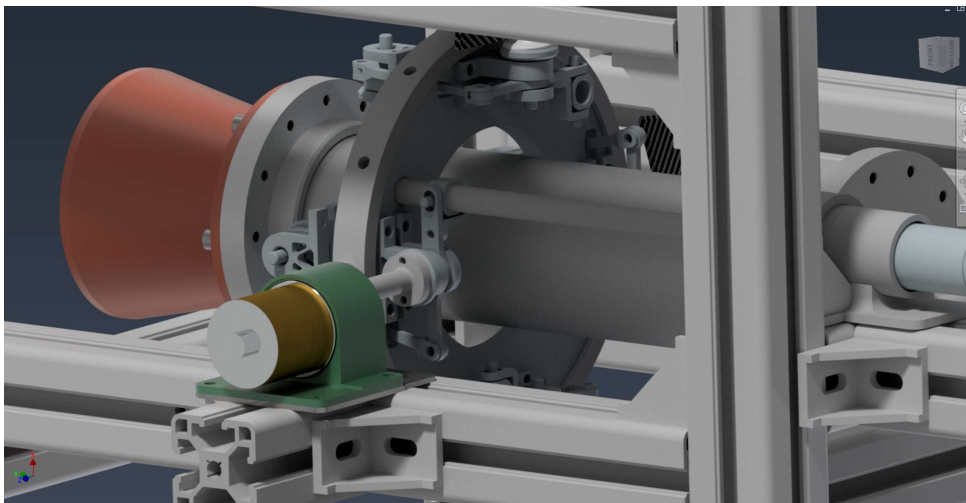


FIGURE 36: Zdjęcie przedstawiające mechanizm w stanowisku Hardware in the loop.

Oprogramowanie do stanowiska Hardware in the loop

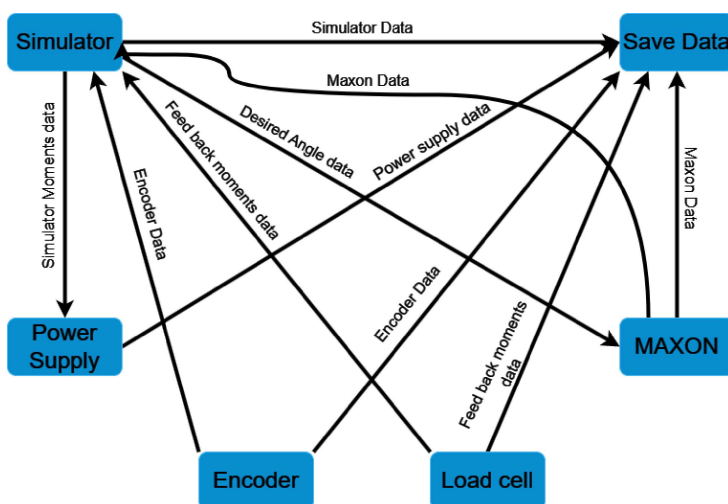


FIGURE 37: Diagram przedstawiający strukturę programu obsługującego stanowisko testowe HIL

- Moduł zarządzania bazą danych

Moduł database/database.py realizuje obsługę bazy danych MongoDB, zapewniając interfejs do przechowywania i pobierania danych. Klasa DatabaseManager umożliwia inicjalizację połączenia, łącząc się z serwerem MongoDB na podstawie konfiguracji wczytywanej z pliku JSON. Podczas inicjalizacji wykonywana jest weryfikacja połączenia poprzez polecenie ping, a ewentualne błędy są logowane. Główną metodą tej klasy jest save, która umożliwia zapis danych w wybranej kolekcji bazy.

- Obsługa enkodera

Moduł encoder/encoder_manager.py jest odpowiedzialny za odczyt i przetwarzanie danych z enkodera. Może działać w trybie rzeczywistym, odczytując dane przez port szeregowy, lub w trybie symulacyjnym, generując dane testowe. Klasa EncoderManager umożliwia przetwarzanie statusu enkodera na czytelne komunikaty oraz przesyłanie tych danych do innych modułów za pomocą kolejek. Główna pętla programu („run”) przetwarza status i wartości kątowe, a następnie przesyła je do modułów odpowiedzialnych za symulację oraz zapisywanie danych.

- Moduł zarządzania kontrolerem Maxon

Moduł maxon/maxon_manager.py jest kluczowym elementem systemu, odpowiedzialnym za komunikację z urządzeniem Maxon. Klasa MaxonManager korzysta z klasy Maxon zaimplementowanej w

z symulatorem oraz modulem zapisującym dane. Główna pętla odbiera dane z kolejki symulatora i na ich podstawie steruje urządzeniem, jednocześnie monitorując bieżący stan urządzenia.

- Moduł symulacji

Moduł odpowiedzialny za symulację integruje funkcjonalności związane z generowaniem danych testowych i wirtualnym odtwarzaniem zachowań systemu. Przyjmuje dane wejściowe z innych modułów, takich jak enkoder czy kontroler Maxon, a następnie przetwarza je zgodnie z założonymi scenariuszami testowymi. Symulacja umożliwia weryfikację działania systemu w kontrolowanych warunkach, co jest szczególnie przydatne w procesie rozwoju i testowania. Dzięki wykorzystaniu kolejek komunikacyjnych, moduł ten może w czasie rzeczywistym wymieniać dane z innymi komponentami, zapewniając ich spójność i synchronizację.

- Moduł sterowania silnikiem generującym symulacyjny moment aerodynamiczny

Moduł sterowania silnikiem odpowiada za implementację algorytmów zarządzających pracą silnika, który generuje moment aerodynamiczny. Głównym celem tego modułu jest symulacja i analiza wpływu momentu na pracę całego systemu. Moduł komunikuje się bezpośrednio z kontrolerem Maxon, przekazując odpowiednie komendy sterujące, takie jak moment, prędkość lub pozycja. Dzięki precyzyjnej kontroli i monitorowaniu parametrów pracy, moduł ten odgrywa kluczową rolę w ocenie stabilności i wydajności systemu w różnych warunkach symulacyjnych.

- Zarządzanie wieloprocusowością

Moduł `process/process.py` implementuje mechanizmy wieloprocusowości poprzez opakowanie klasy `multiprocessing.Process`. Klasa `Process` umożliwia definiowanie źródeł i celów danych za pomocą kolejek oraz uruchamianie procesów w osobnych wątkach. Taka architektura zapewnia możliwość równoległego przetwarzania danych, co jest kluczowe w systemach czasu rzeczywistego.

- Narzędzia wspomagające

Moduł `utils/utils.py` dostarcza zestaw narzędzi, takich jak logger, funkcje do pracy z konfiguracją oraz wsparcie dla trybów symulacyjnych. Logger konfigurowany w tym module jest używany przez wszystkie inne komponenty systemu, co pozwala na centralizację i standaryzację logowania zdarzeń.

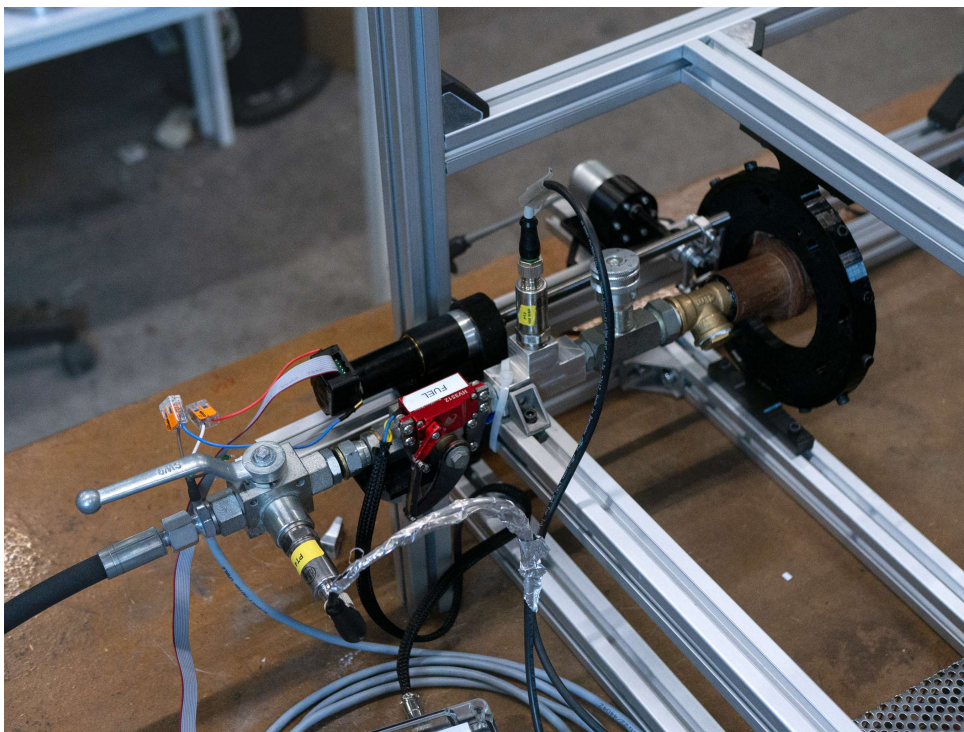


FIGURE 38: Zdjęcie przedstawiające rzeczywiste stanowisko Hardware in the loop.

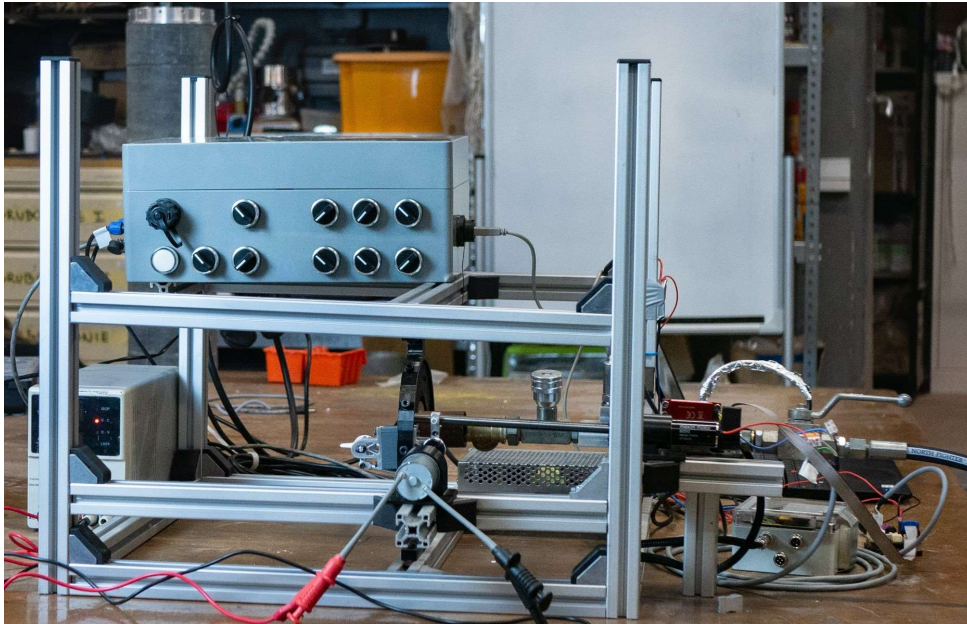


FIGURE 39: Zdjęcie przedstawiające rzeczywisty mechanizm w stanowisku Hardware in the loop.

6.2 Stanowisko do badania właściwości dynamicznych aktuatorów

Do odpowiedniego przetestowania zaprojektowanych algorytmów sterowania wykorzystano stanowisko przedstawione na rys. 40. Jest to platforma testowa, która wznosi się na wybraną wysokość, jednocześnie sterując kierunkiem odchylenia osi podłużnej od kierunku pionowego. Dzięki temu, możliwy jest stabilny lot na wybranej wysokości. Jest to bezpośrednia analogia do rakiety posiadającej możliwość wektorowego sterowania ciągiem. Składa się z kanału, w którym znajdują się silniki dronowe. Silniki te generują przepływ powietrza, którego kierunek zmieniany jest za pomocą powierzchni sterowych umieszczonych za dyszą. Taka właściwość umożliwia efektywne sterowanie, które przyczynia się do stabilnego lotu.



FIGURE 40: Zdjęcie przedstawiające stanowisko do badania właściwości dynamicznych aktuatorów.

Stanowisko to pozwoliło w dokładny sposób przetestować działanie algorytmów sterowania wykorzystując wybrane mechanizmy wykonawcze. Na rys. 41, 42 przedstawione są kolejno wykresy błędu

uzyskiwanych przez stanowisko testowe podczas fazy testowej. Na stanowisko testowe podczas realizacji tego eksperymentu, celowo były podawane dodatkowe wymuszenia sprawdzając tym samym skuteczność działania układu sterowania. Tym samym, testy potwierdziły kompatybilność wybranego rozwiązania algorytmicznego z rzeczywistą implementacją sprzętową mechanizmu wykonawczego.

Error angle

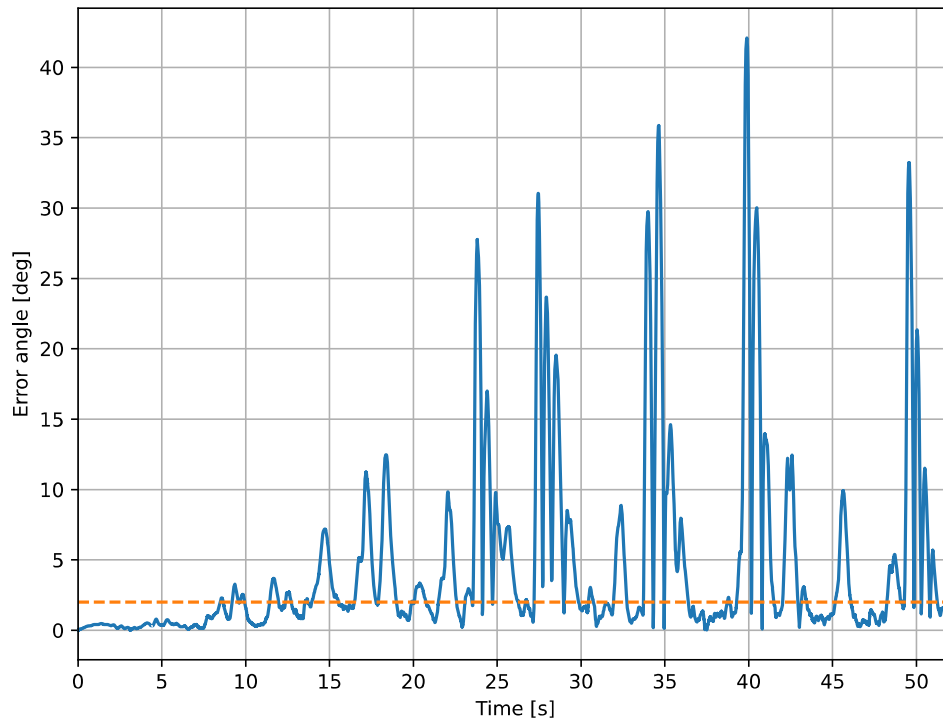


FIGURE 41: Wykres przedstawiający błąd odchylenia osi podłużnej od kierunku pionowego stanowiska testowego. Przerywaną linią została oznaczona górna granica zakresu akceptowalnego błędu.

Control surfaces deflection angles

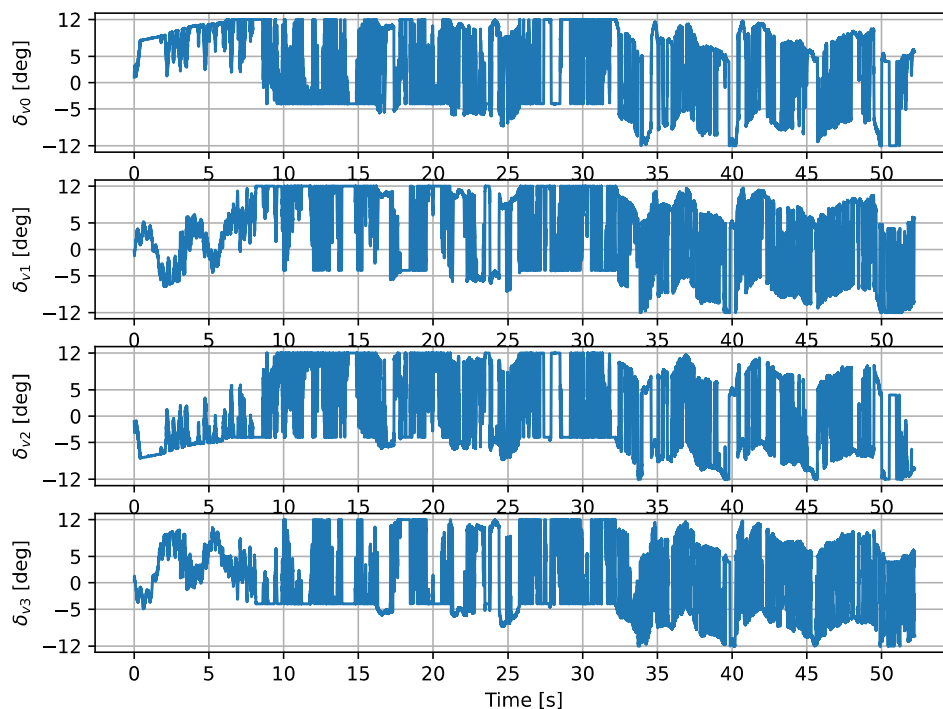
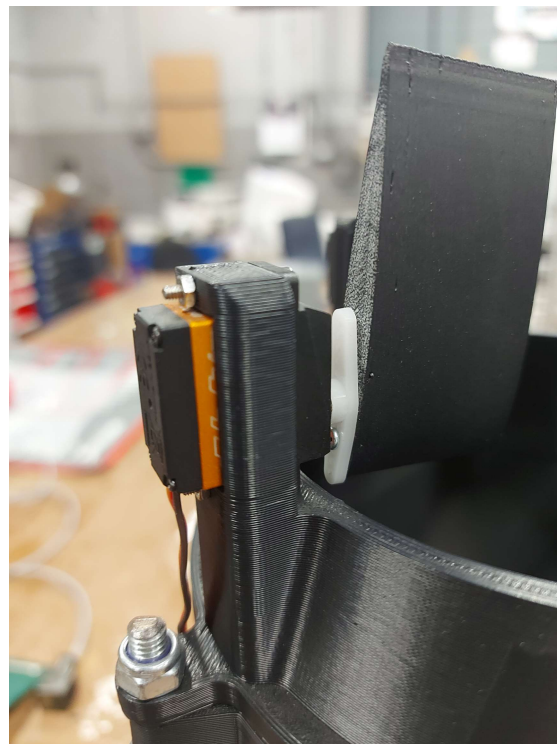


FIGURE 42: Wykres przedstawiający wartości odchylenia poszczególnych powierzchni sterowych.



(A) Silnik z kontrolerem prędkości obrotowych



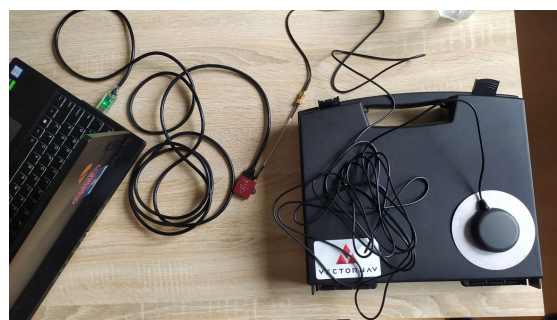
(B) Serwomechanizm kontrolujący wychylenia powierzchni sterowych

FIGURE 43: Zaimplementowane na stanowisku akulatory

Stanowisko było wyposażone w zaawansowany system sterowania, którego sercem była jednostka Raspberry Pi, pracująca na systemie operacyjnym Linux RT (Real-Time). Raspberry Pi pełniło kluczową rolę w realizacji algorytmów sterowania, zapewniając wysoką precyzję i niezawodność w czasie rzeczywistym. W celu efektywnego sterowania mechanizmami wykonawczymi wykorzystano serwomechanizmy, które odpowiadały za precyzyjne ustawienie powierzchni sterowych. Dodatkowo, dane niezbędne do nawigacji i stabilizacji były pozyskiwane z zaawansowanego systemu VectorNav, integrującego funkcje INS (Inertial Navigation System) i GPS. Dzięki temu możliwe było uzyskanie dokładnych informacji o orientacji, prędkości oraz pozycji, co znacząco zwiększyło skuteczność działania stanowiska testowego i pozwoliło na precyzyjną weryfikację opracowanych algorytmów.



(A) RaspberryPi 5



(B) VectorNav

FIGURE 44: Dwie platformy sprzętowe umożliwiające pomiary i wykonywanie obliczeń

6.3 Dokumentacja algorytmu sterującego

Model dynamiki rakiety został zaimplementowany z wykorzystaniem języka Python oraz pakietów *NumPy* i *SciPy*. Model został podzielony na moduły tak jak przedstawiono na rysunku 45:


```

16     pitch_ctrl_sig = self.pitch_controller.calc_control(ref=angular_vel_ctrl[1],
    ↪     measured=angular_velocity[1], last_ctrl_sig=last_pitch_ctrl)
17     yaw_ctrl_sig = self.yaw_controller.calc_control(ref=angular_vel_ctrl[2],
    ↪     measured=angular_velocity[2], last_ctrl_sig=last_yaw_ctrl)
18
19     return roll_ctrl_sig, pitch_ctrl_sig, yaw_ctrl_sig

```

LISTING 6.1: Implementacja sterownika w języku Python

Klasa implementująca sterownik zawiera metodę `calc_control`, której zadaniem jest obliczanie sygnałów sterujących. Przyjmuje ona wektor zadanego kierunku lotu, orientacja rakiety wyrażone za pomocą kwaternionu oraz sygnały sterujące wykorzystane podczas poprzedniego okresu próbkowania jako argumenty, a następnie wywołuje wszystkie sterowniki w odpowiedniej kolejności.

Implementacja kwaternionowego sterownika PD pętli zewnętrznej przedstawiona jest na listingu 6.2. Podobnie do klasy `CascadeController`, konstruktor klasy `QuatPdController` jako argumenty wzmocnienia sterownika oraz wartość saturacji sygnałów sterujących. Posiada ona także metodę `calc_control` wyznaczającą sygnały sterujące zadane na pętlę wewnętrzną.

```

1 class QuatPdController:
2     def __init__(self, kp, kd, sat_signal):
3         self.kp = kp
4         self.kd = kd
5         self.sat_signal = sat_signal
6
7     def calc_control(self, ref_dir, orientation, angular_velocity):
8         ref_dir_body = orientation.rot_global_to_body(ref_dir)
9         error_angle = np.arccos(np.clip(ref_dir_body[0], -1.0, 1.0))
10        error_dir = np.cross(np.array([1, 0, 0]), ref_dir_body) / np.sin(error_angle) if not
    ↪        isclose(error_angle, 0.0) else np.zeros(3)
11        error_quat_scalar = np.cos(error_angle / 2)
12        error_quat_vec = np.sin(error_angle / 2) * error_dir
13
14        ctrl_sig = self.kp * np.sign(error_quat_scalar) * error_quat_vec + self.kd * (1 +
    ↪        error_quat_vec @ error_quat_vec) * angular_velocity
15
16        return np.clip(ctrl_sig, a_min=-self.sat_signal, a_max=self.sat_signal)

```

LISTING 6.2: Implementacja kwaternionowego sterownika PD

Sterowniki EADRC znajdujące się w pętli wewnętrznej sterownika zaimplementowane zostały w postaci klasy `EADRC`, która została przedstawiona na listingu 6.3. Jej konstruktor przyjmuje jako parametry: okres próbkowania pętli wewnętrznej, pasmo przenoszenia obserwatora LESO, pasmo przenoszenia sterownika oraz estymowaną wartość wzmocnienia na wejściu sterowanego obiektu. Podobnie jak w przypadku kwaternionowego sterownika PD klasa `EADRC` implementuje metodę `calc_control`, której zadaniem jest obliczanie aktualnej wartości sygnału sterującego.

```

1 class EADRC:
2     def __init__(self, Tp, omega_o, omega_c, b_est):
3         self.Tp = Tp
4         self.kp = omega_c
5         self.b_est = b_est
6
7         self.leso_state = np.zeros(2)
8         self.A = np.eye(2) + Tp * np.array([[0, 1], [0, 0]])
9         self.B = Tp * np.array([-b_est, 0])

```

```

11
12 def calc_control(self, ref, measured, last_ctrl_sig):
13     self.leso_state = self.A @ self.leso_state + self.B * last_ctrl_sig + self.L * (ref -
14         ↪ measured - self.leso_state[0])
15     ctrl_sig = (self.kp * error + self.leso_state[1]) / self.b_est
16     return ctrl_sig

```

LISTING 6.3: Implementacja sterownika EADRC

Algorytm alokacji sygnałów sterujących został zaimplementowany w sposób przedstawiony na listingu 6.4. Wychylenia poszczególnych powierzchni sterowych zależą od sygnałów sterujących sterownikami EADRC pętli wewnętrznej. Przed wykonaniem stosownych obliczeń, przechodzą one przez blok saturacji, aby ograniczyć ich wartości do możliwości układu wykonawczego.

```

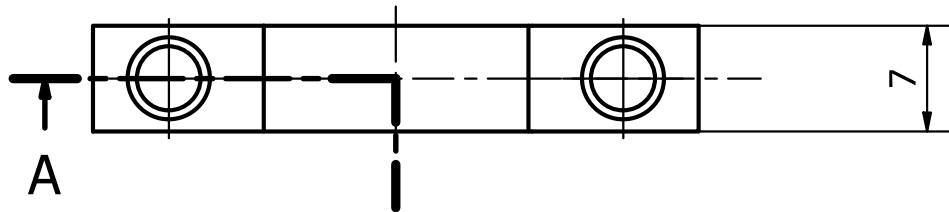
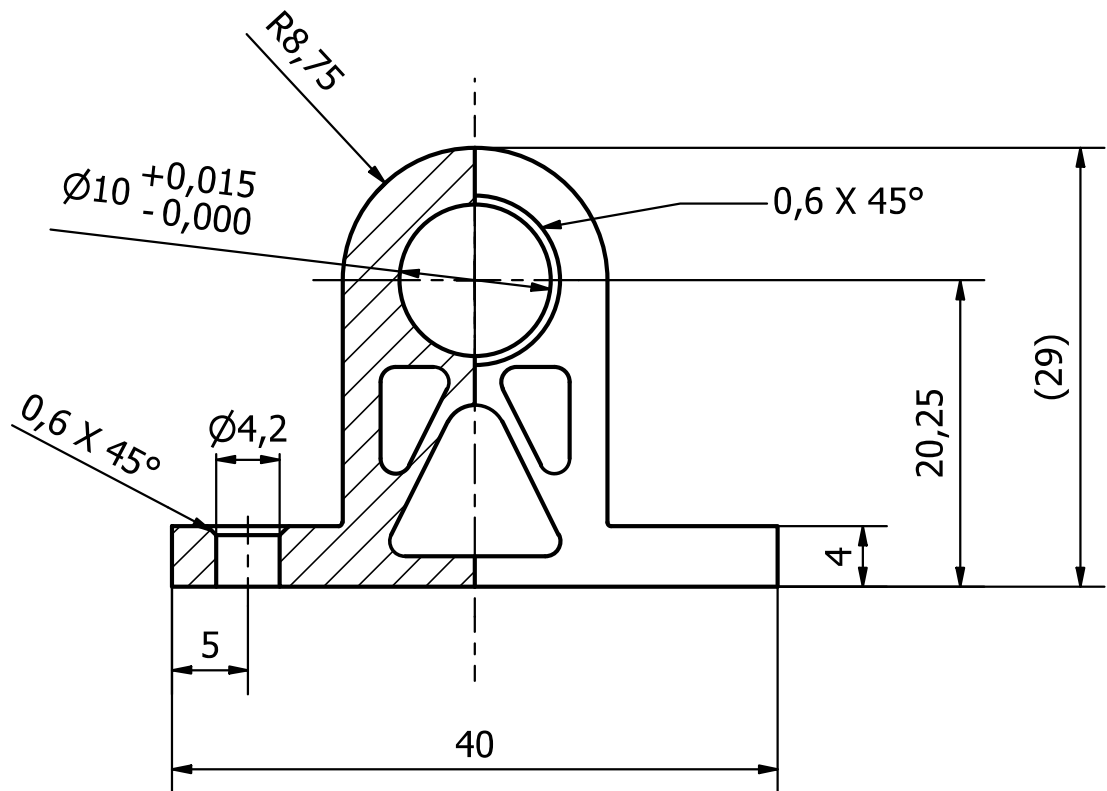
1 roll_ctrl_sig = np.clip(roll_sig, -np.deg2rad(4.0), np.deg2rad(4.0))
2 pitch_ctrl_sig = np.clip(pitch_sig, -np.deg2rad(6.0), np.deg2rad(6.0))
3 yaw_ctrl_sig = np.clip(yaw_sig, -np.deg2rad(6.0), np.deg2rad(6.0))
4
5 ctrl_fins_deflect = np.array([-roll_ctrl_sig - pitch_ctrl_sig, -roll_ctrl_sig + yaw_ctrl_sig,
6     ↪ -roll_ctrl_sig + pitch_ctrl_sig, -roll_ctrl_sig - yaw_ctrl_sig])
7 jet_vanes_deflect = ctrl_fins_deflect

```

LISTING 6.4: Algorytm alokacji sygnałów sterujących

6.4 Dokumentacja wykonawcza mechanizmu sterującego stanowiska HIL

A-A (2 : 1)

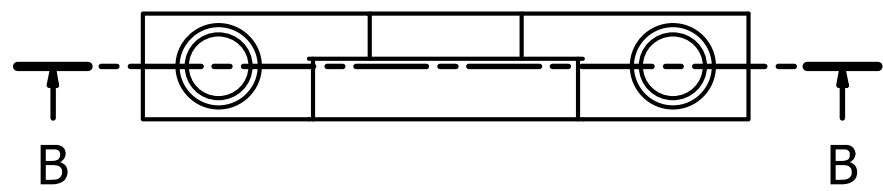
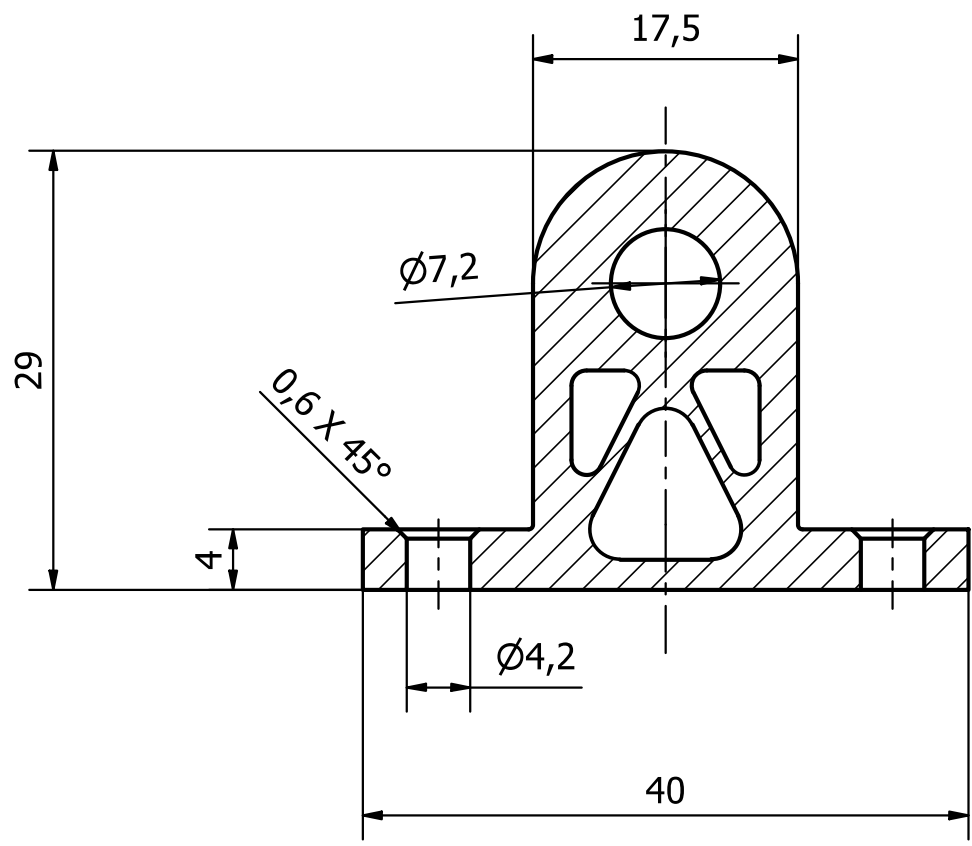


1. Ostre krawędzie zatępić.

Designer	30.05.2024	Max Falbierski	Quantity: 2	Sheet size: A4	Material: Aluminium 7075
Draftsman	09.07.2024	M.Gaworzewski	Revision: 1	Project: ATC	
Engineer	09.07.2024	Max Falbierski			
Scale:		Part:	CC_01_0001_BearingHolder_rev2.prt		

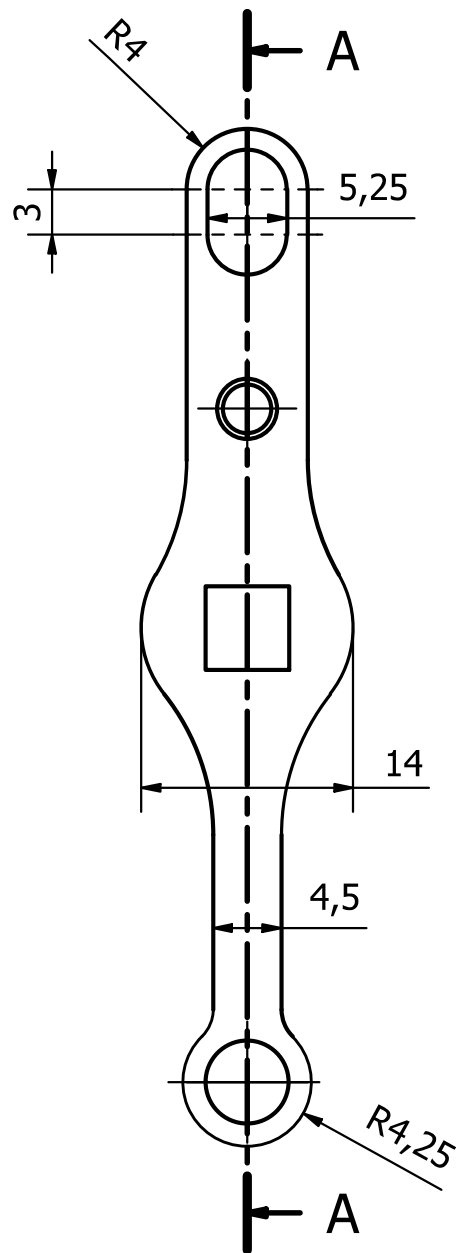
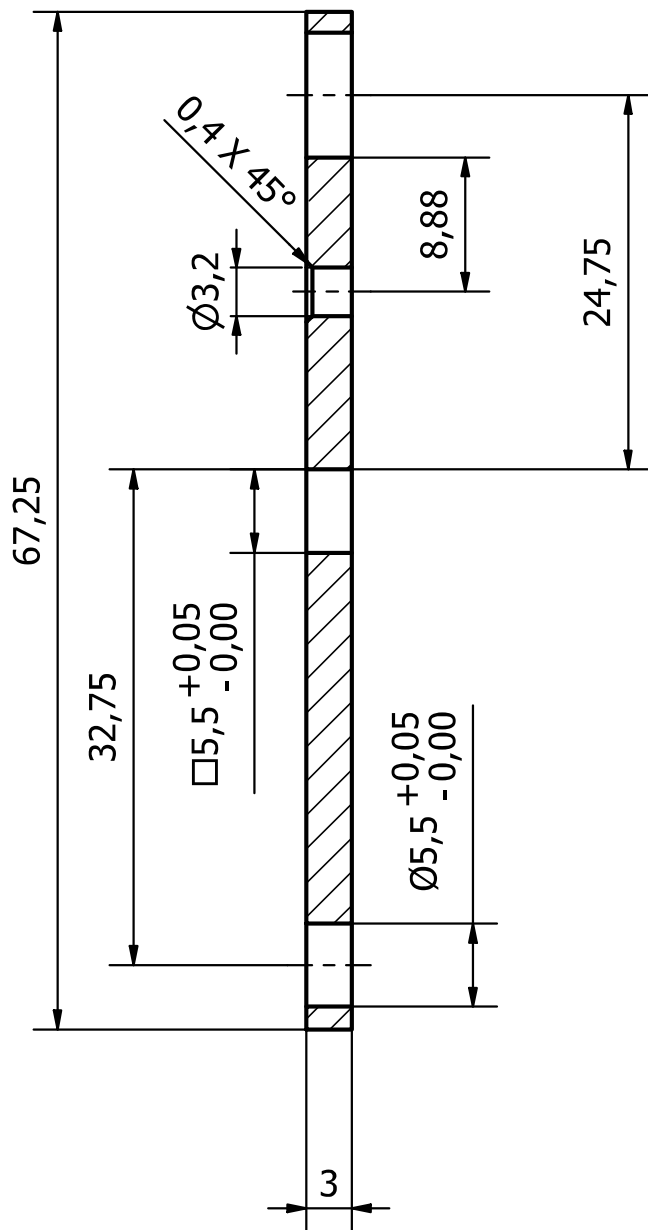


B-B (2 : 1)



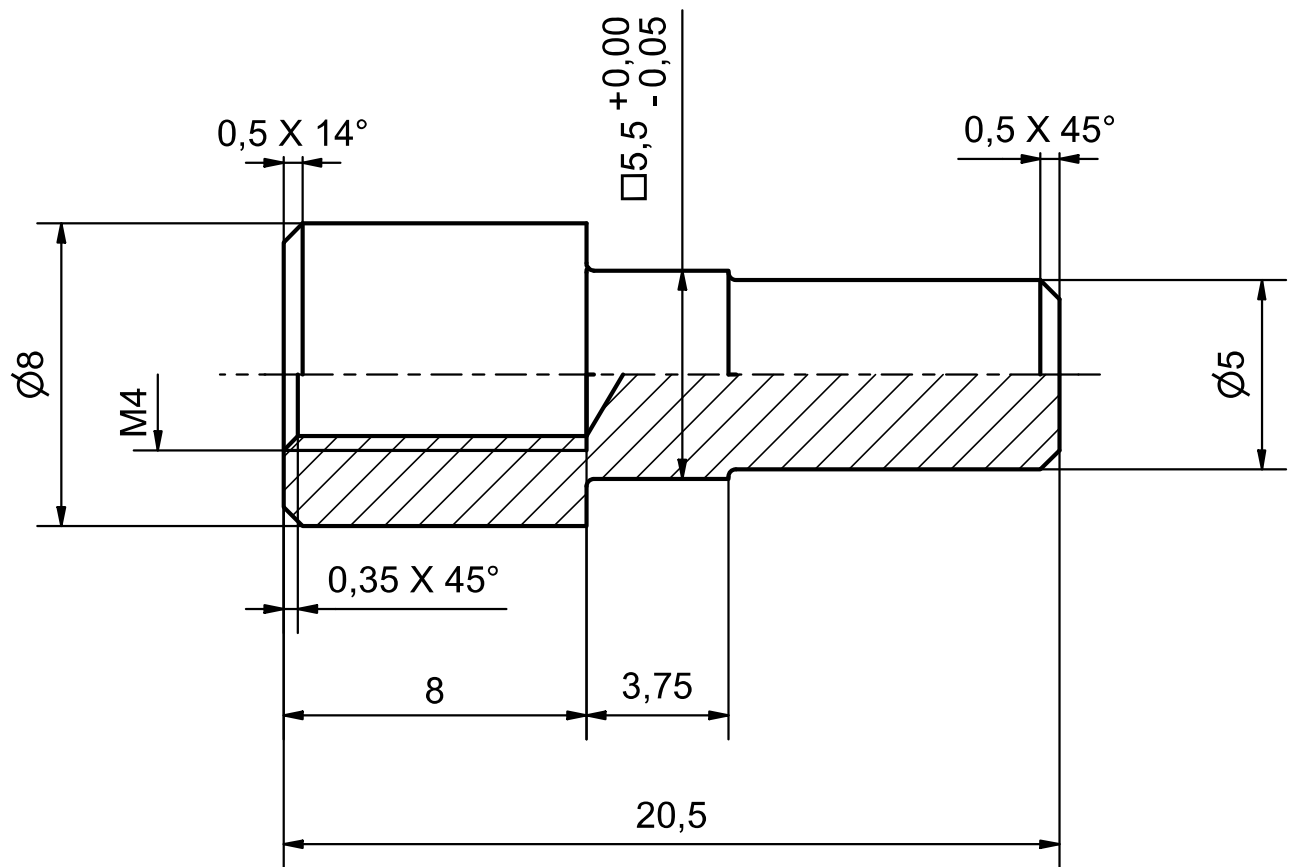
Designer	12.12.2023	Max Falbierski	Quantity: 2	Sheet size: A4	Material: Aluminium 7075
Draftsman	04.11.2024	PC	Revision:	Project: ATC	
Engineer	04.11.2024	Max Falbierski			
Scale:	Part:		ca. 01.0002 Bearing holder		

A-A (2 : 1)



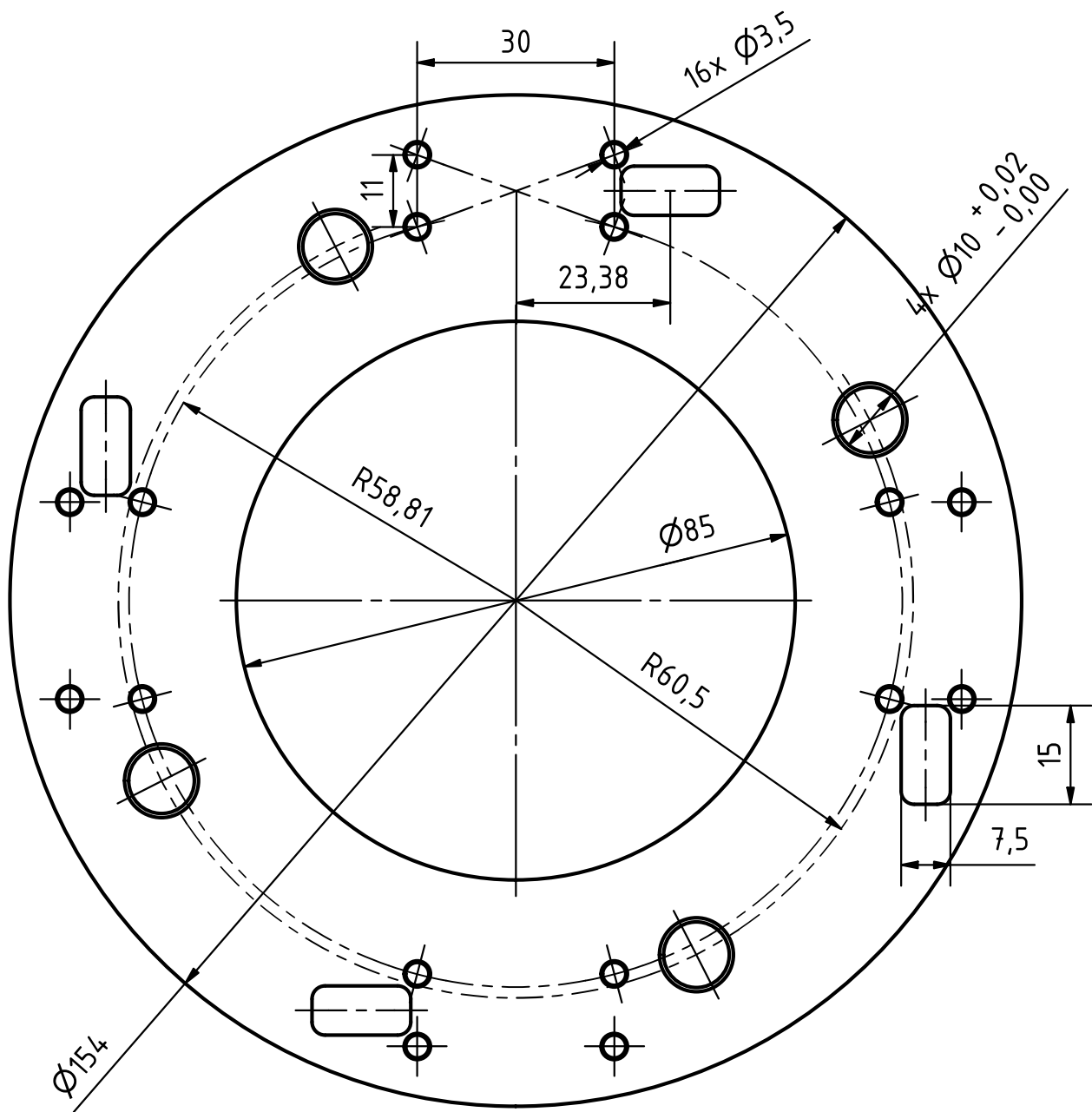
1. Ostre krawędzie zatępić.

Designer	30.05.2024	Max Falbieski	Quantity: 1	Sheet size: A4	Material: Aluminium 7075
Draftsman	09.07.2024	M.Gaworzewski	Revision: 1	Project:	ATC
Engineer	09.07.2024	Max Falbieski			
Scale:		Part:	CC_01_0003_FinMotorLever_rev2 prt		



1. Ostre krawędzie zatępić

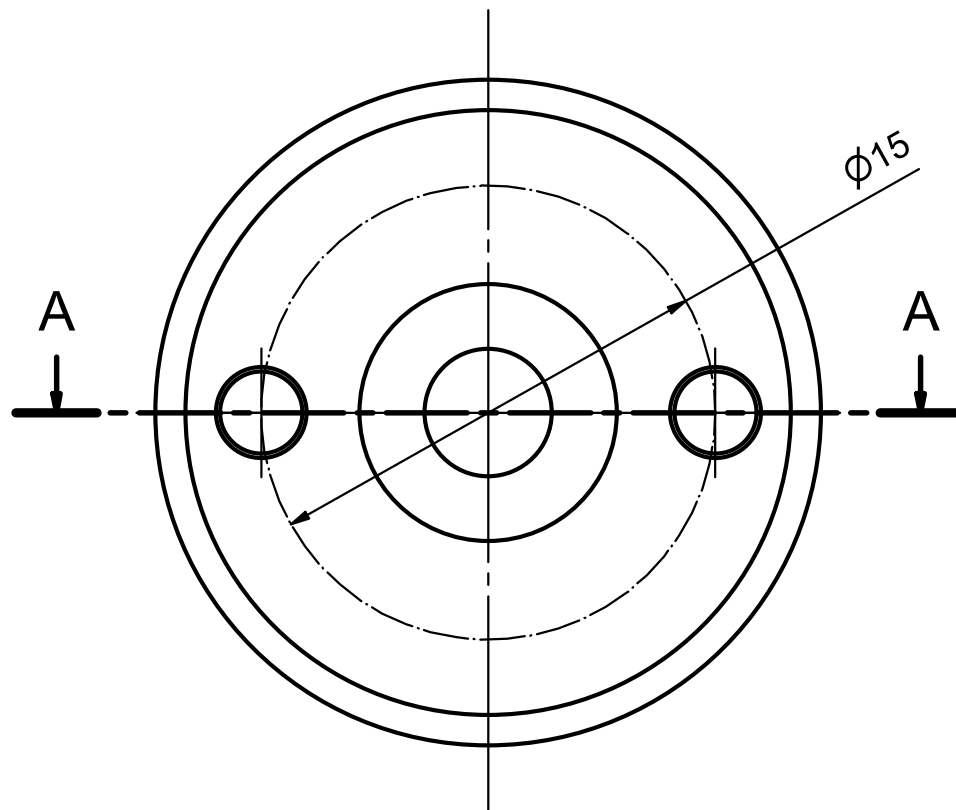
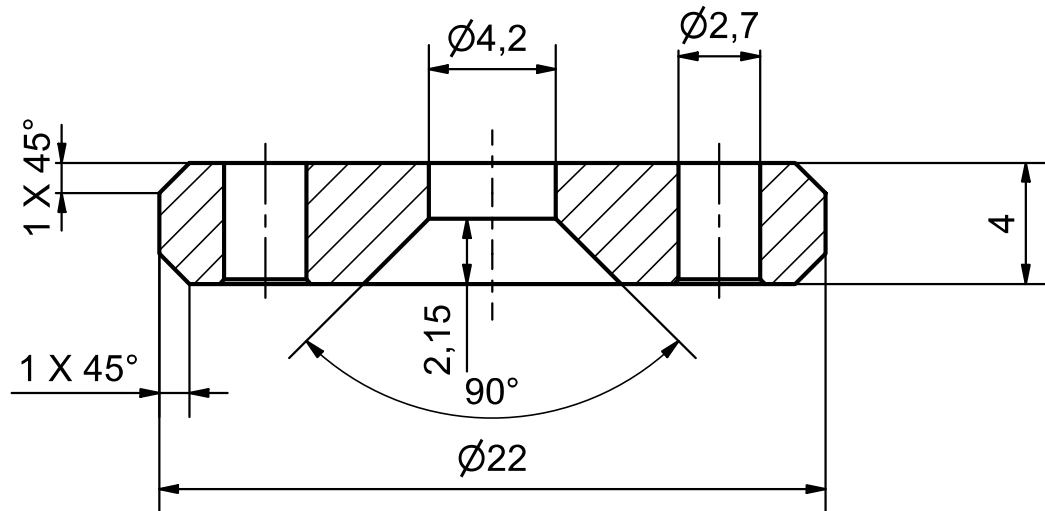
Designer	11.07.2024	Max Falbierski	Quantity:	Sheet size:	Material:
Draftsman	11.07.2024	M.Gaworzewski	2	A4	Aluminum 7075
Engineer	11.07.2024	Max Falbierski	Revision:	Project:	
			1	ATC	



ostre krawędzie stępić

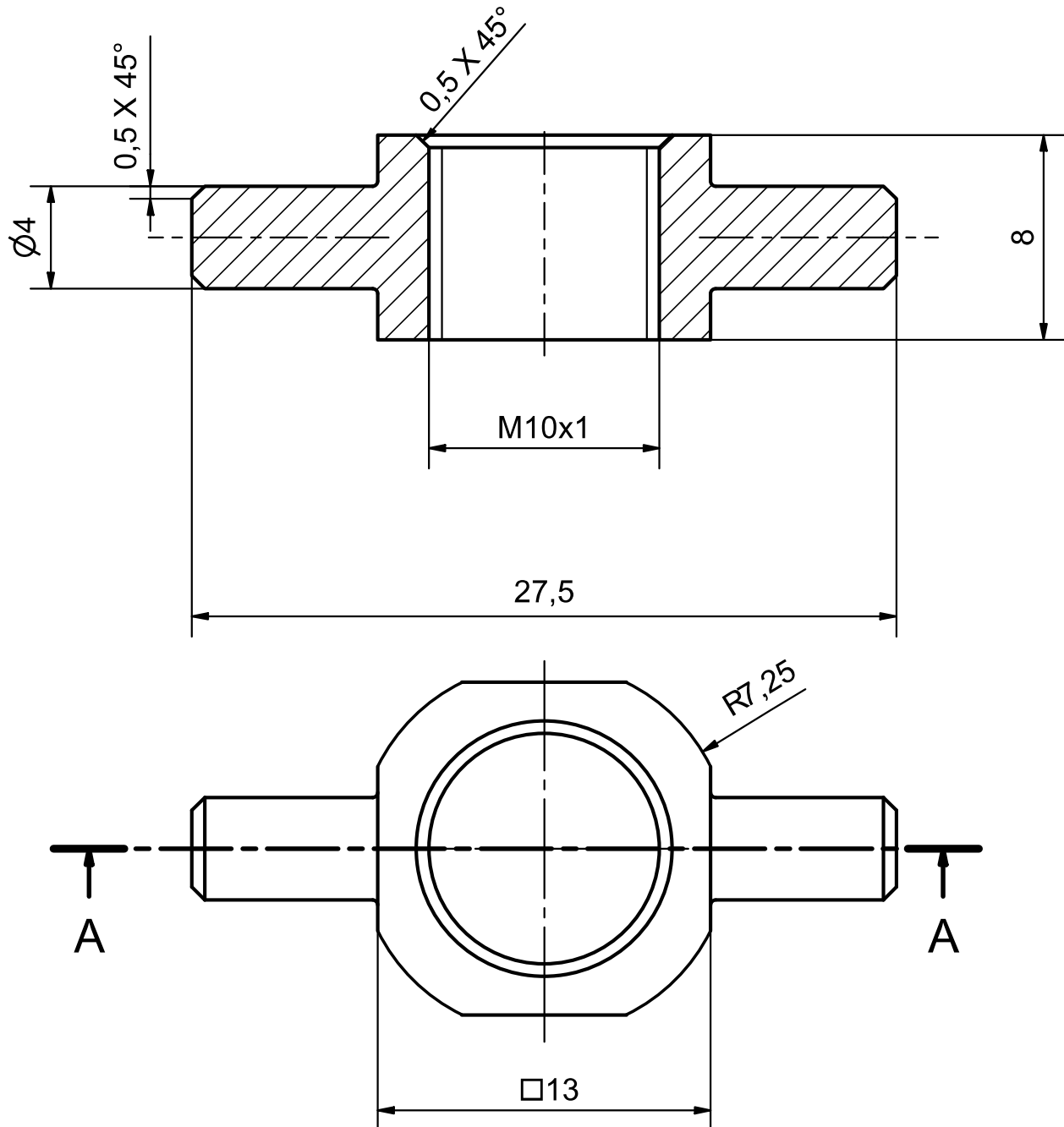
Designer	05.12.2023	PC	Quantity:	Sheet size:	Material:
Draftsman	18.07.2024	Administrator	1	A4	Ogólny
Engineer	18.07.2024	Mieszko Gaworzewski	Revision:	Project:	
Scale:		Part:			

A-A (4 : 1)

1. Otwory $\text{Ø}2,7$ fazowawc $0,15 \times 45^\circ$

Designer	30.05.2024	Max Falbierski	Quantity:	Sheet size:	Material:
			2	A4	Aluminium 7075
Draftsman	11.07.2024	M.Gaworzewski	Revision:	Project:	
			1	ATC	
Engineer	11.07.2024	Max Falbierski			
Scale:		Part:			

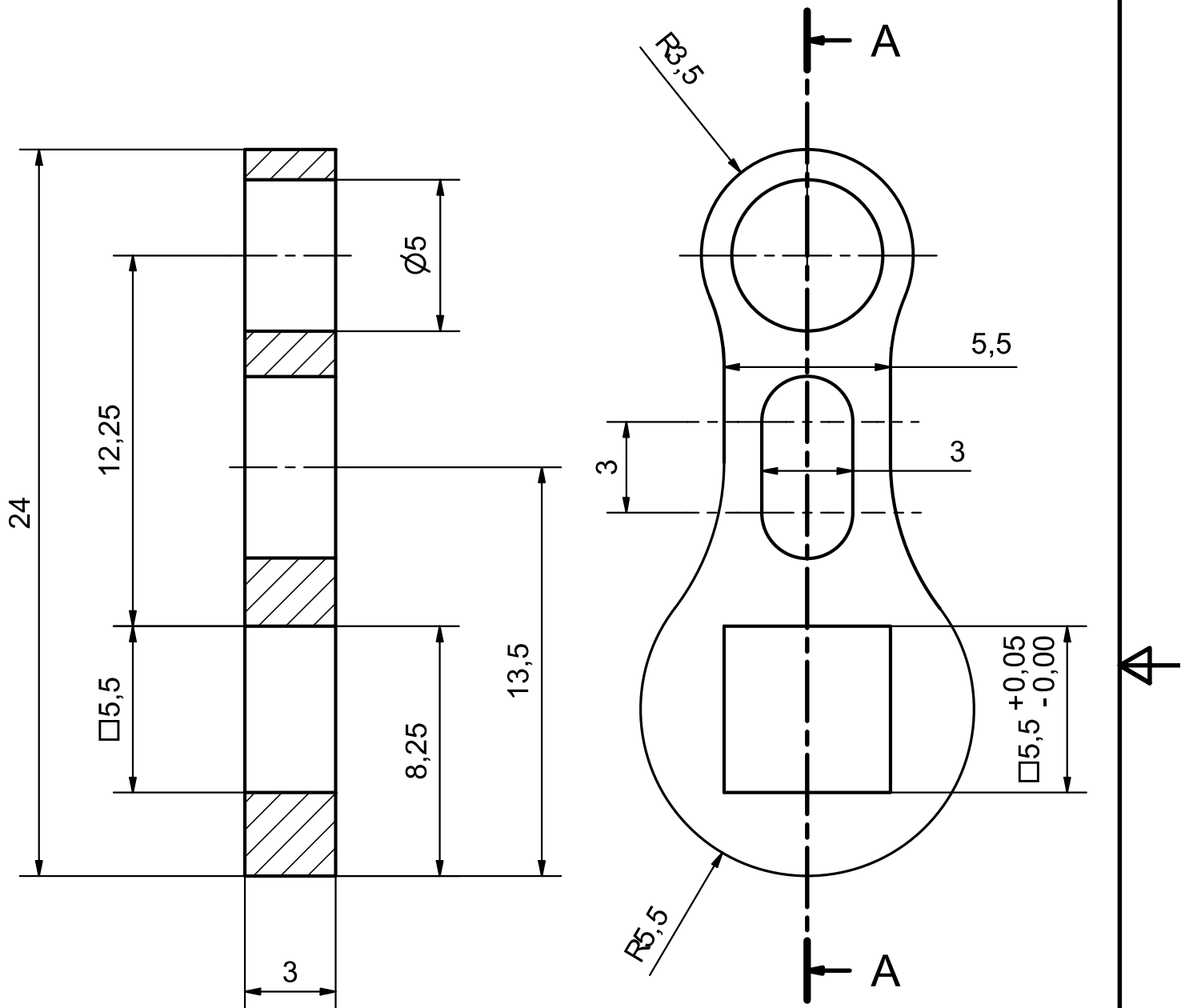
A-A (4 : 1)



1. Ostre krawędzie zatępić

Designer	30.05.2024	Max Falbierski	Quantity: 1	Sheet size: A4	Material: Aluminium 7075
Draftsman	11.07.2024	M.Gaworzewski	Revision: 1	Project:	ATC
Engineer	11.07.2024	Max Falbierski			
Scale:		Part:	66_03_0001_DriveAdapter_rev2.prt		

A-A (5 : 1)



1. Ostre krawędzie zatępić

Designer	30.05.2024	Max Falbierski	Quantity: 1	Sheet size: A4	Material: Aluminium 7075
Draftsman	11.07.2024	M.Gaworzewski	Revision: 1	Project:	ATC
Engineer	11.07.2024	Max Falbierski			
Scale:		Part:			

7 Podsumowanie

Projekt zakończył się sukcesem, osiągając założone cele zarówno w zakresie projektowym, jak i badawczym. Stworzono stanowisko testowe umożliwiające testowanie systemu aktywnego sterowania trajektorią rakiety, który pozwala na stabilizację lotu, precyzyjną kontrolę trajektorii.

Efektywność Mechanizmów Sterowania: Wdrożony system sterowania oparty na modułowych powierzchniach sterowych i łopatkach jet vanes w strumieniu gazów wylotowych według przeprowadzonych symulacji i testów na stanowiskach okazał się skuteczny. Testy symulacyjne wykazały, że czas stabilizacji mieścił się w przyjętych normach (poniżej 2 sekund), a błąd odchylenia trajektorii w większości przypadków pozostawał w zakresie $\pm 2^\circ$.

Skuteczność Algorytmów Sterowania: Algorytmy sterowania kaskadowego, oparte na sterownikach EADRC i kwaternionowych, pozwoliły na precyzyjne wystereowanie rakiety nawet w dynamicznie zmieniających się warunkach. Działanie algorytmów zostało przetestowane w wielu scenariuszach (dwa scenariusze zostały przykładowo zaprezentowane), które potwierdziły ich zdolność do stabilizacji rakiety oraz utrzymania zadanych parametrów trajektorii.

Osiągnięcia Technologiczne: Użycie obliczeń numerycznych przy projektowaniu systemów sterowania i następnej ich analizie pozwoliło na dokładne odwzorowanie warunków lotu i zoptymalizowanie mechanizmów sterujących pod kątem maksymalizowania generowanych momentów przy minimalnych kątach natarcia. Łopatki jet vanes, mimo ograniczonej fazy operacyjnej (działające jedynie podczas pracy silnika), wykazują wysoką skuteczność w generowaniu sił i momentów niezbędnych do korekcji trajektorii.

Stanowiska Testowe i Symulacyjne: Utworzenie stanowisk typu Hardware-in-the-Loop oraz innych platform testowych umożliwiło realne odwzorowanie warunków lotu, co znacząco zwiększyło wiarygodność wyników.

Wnioski Końcowe: Projekt „Active Trajectory Control” demonstruje, jak integracja zaawansowanych technologii mechanicznych, algorytmicznych i symulacyjnych może przyczynić się do opracowania nowoczesnych rozwiązań dla branży kosmicznej. Opracowany system jest nie tylko skuteczny, ale również wystarczająco uniwersalny, aby dostosować go do różnych konfiguracji rakiet i systemów napędowych. Wyniki projektu mogą posłużyć jako fundament dla przyszłych, bardziej zaawansowanych systemów sterowania raketowego.

Bibliography

- [1] Tomasz Brodzinski. Design and analysis of a supersonic rocket aerodynamic control system, 2024.
- [2] PUT Rocketlab. Hexa 4 - team 76 project technical report to the 2023 spaceport america cup. 2023.
- [3] G. P. Sutton. *Rocket Propulsion Elements*. JOHN WILEY & SONS, INC., 9th edition, 2017.
- [4] Piotr Zacholski. Modeling and motion control of a sounding rocket, 2024.