

**Kauno technologijos universitetas**

Mechanikos inžinerijos ir dizaino fakultetas

# **Orlaivių aeroservotamprumo modelio kūrimas ir tyrimas**

Baigiamasis magistro projektas

---

**Mindaugas Dagilis**

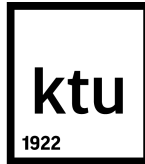
Projekto autorius

**Doc. Sigitas Kilikevičius**

Vadovas

---

**Kaunas, 2022**



**Kauno technologijos universitetas**

Mechanikos inžinerijos ir dizaino fakultetas

## **Orlaivių aeroservotamprumo modelio kūrimas ir tyrimas**

Baigiamasis magistro projektas

Aeronautikos inžinerija (6211EX024)

---

**Mindaugas Dagilis**

Projekto autorius

**Doc. Sigitas Kilikevičius**

Vadovas

**Prof. Algimantas Fedaravičius**

Recenzentas

---

**Kaunas, 2022**



**Kauno technologijos universitetas**

Mechanikos inžinerijos ir dizaino fakultetas

Mindaugas Dagilis

## **Orlaivių aeroservotamprumo modelio kūrimas ir tyrimas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Mindaugas Dagilis

*Patvirtinta elektroniniu būdu*



**Kauno technologijos universitetas**  
Mechanikos inžinerijos ir dizaino fakultetas  
Studijų programa – Aeronautikos inžinerija (6211EX024)

## **Magistro baigiamojo projekto užduotis**

Studentui Mindaugui Dagiliui

### **1. Baigiamojo projekto tema:**

Orlaivių aeroservotamprumo modelio kūrimas ir tyrimas

Development and Research of an Aircraft Aeroservoelastic Model

### **2. Projekto tikslas:**

Sudaryti kompiuterinį aeroservotamprumo modelį ir atlikti jo eksperimentinį validavimą.

### **3. Projekto uždaviniai:**

1. Atlikti literatūros apie aeroservotamprumo modelius ir jų taikymą analizę
2. Sudaryti aeroservotamprumo fizikinį modelį
3. Sukurti kompiuterinę fizikinio aeroservotamprumo modelio versiją
4. Atlikti eksperimentinius lėktuvo skrydžio bandymus ir validuoti sukurtą aeroservotamprumo modelį
5. Atliekant surinktų duomenų analizę įvertinti aeroservotamprumo modelio veiksmingumą

### **4. Projekto aprašomosios dalies struktūra:**

1. Literatūros apie aeroservotamprumo modelius ir jų taikymą analizė
2. Sudaryto aeroservotamprumo modelio aprašymas
3. Modelio validavimo metodikos aprašas
4. Modelio validavimo rezultatų aprašymas

### **5. Projekto konsultantai:**

Baigiamojo projekto autorius	Mindaugas Dagilis	2020-10-09
	<i>(vardas, pavardė, data)</i>	
Baigiamojo projekto vadovas	Doc. Sigitas Kilikevičius	2020-10-09
	<i>(pareigų sutrumpinimas, vardas, pavardė, data)</i>	
Krypties studijų programų vadovas	Doc. Saulius Japertas	2020-10-09
	<i>(pareigų sutrumpinimas, vardas, pavardė, data)</i>	

Mindaugas Dagilis. Orlaivių aeroservotamprumo modelio kūrimas ir tyrimas. Magistro baigiamasis projektas / vadovas doc. Sigitas Kilikevičius; Kauno technologijos universitetas, Mechanikos inžinerijos ir dizaino fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Aeronautikos inžinerija (E14), Inžinerijos mokslai.

Reikšminiai žodžiai: aeroservotamprumas; skrydžio modeliavimas; sūkurių tinklelio metodas; aerotamprumas.

Kaunas, 2022. 50 p.

## Santrauka

Konstruojant orlaivius nuolat siekiama kuo mažesnės konstrukcijos masės, todėl aviacijoje vis plačiau naudojamos kompozitinės medžiagos. Nors jos turi labai gerą stiprumo-masės santykį, palyginus su aviacijoje naudojamais metalo lydiniais ir mediena, jų maksimali galima deformacija yra didesnė, todėl kompozitiniai orlaiviai dažnai skrydžio metu patiria didesnes deformacijas už metalinius ir medinius. Mažesnis standumas lemia ankstesnę flaterį ir prastesnes valdymo charakteristikas. Norint įvertinti ir kompensuoti neigiamus mažo standumo struktūros efektus, reikalingas geras aeroservotamprumo modelis.

Atlikus aeroservotamprumo literatūros analizę nuspręsta sukurti aeroservotamprumo modelį, sudarytą iš klasikinio kinematinio modelio, struktūrinio modelio, paremto standžiais elementais sujungtais kampinėmis spyruoklėmis, ir sūkurių tinklelio metodu paremto aerodinaminio modelio. Šis fizikinis aeroservotamprumo modelis įgyvendinamas programiškai naudojant C programavimo kalbą. Rezultatų palyginimui taip pat sudaroma standaus lėktuvo modelio versija, naudojanti tą patį aerodinaminį ir kinematinį modelį, bet neatsižvelgianti į lėktuvo struktūros deformacijas.

Modeliui validuoti sudaryta skrydžio bandymų metodika. Skrydžio bandymai atlikti naudojant tam sukonstruotą fiksuoto sparno bepilotį orlaivį. Skrydžio metu orlaivis filmuotas dviem vaizdo kameromis, taip fiksuota orlaivio padėtis erdvėje. Pačiame orlaivyje esantis mikrovaldiklis fiksavo akcelerometro, giroskopo ir servovariklių valdymo signalų duomenis.

Pagal skrydžio bandymams naudotą bepilotį orlaivį buvo sudarytas skaičiuojamasis lėktuvo modelis. Jo inercinės savybės nustatytos sveriant atskirus lėktuvo dalis ir skaičiuojant inercijos momentus, o struktūrinės savybės nustatytos apkraunant lėktuvo dalis žinoma jėga, matuojant jų poslinkius ir pagal tai apskaičiuojant kampinių spyruoklių standumą.

Naudojant pradinę orlaivio padėtį, nustatytą iš filmuotos medžiagos, sumodeliuoti atskiri orlaivio skrydžio fragmentai, jų metu apskaičiuoti pagreičiai palyginti su eksperimentiškai išmatuotais skrydžio metu. Sukurto aeroservotamprumo modelio paklaidos žymiai mažesnės palyginus su standžiu modeliu, bet per didelės kad modelis būtų naudingas be tolimesnių tobulinimų.

Mindaugas Dagilis. Development and Research of an Aircraft Aeroservoelastic Model. Master's Final Degree Project / supervisor assoc. prof. Sigita Kilikevičius; Faculty of Mechanical Engineering and Design, Kaunas University of Technology.

Study field and area (study field group): Aeronautical Engineering (E14), Engineering Science.

Keywords: aeroservoelasticity; flight modelling; vortex lattice method; aeroelasticity.

Kaunas, 2022. 50 pages.

### **Summary**

The use of composite materials in aviation is increasing because of constant need to minimise the mass of aircraft. Composite materials have excellent strength to weight ratios when compared with metal alloys and timber used in aviation, but they have a higher maximum deformation, leading to higher structural deformations in flight. Reduced stiffness can cause flutter at lower speeds and degrade the handling characteristics of the aircraft. To evaluate and compensate the negative effects of low structural stiffness, a good aeroservoelasticity model is required.

An aeroservoelasticity literature analysis was performed and it was decided to develop an aeroservoelasticity model, composed of a classical kinematics model, a structural model based on stiff elements connected with torsion springs, and a vortex lattice method based aerodynamic model. This physical aeroservoelasticity model was implemented computationally using the C programming language. For comparison of results a rigid version of the flight model was also developed using the same kinematic and aerodynamic models, but ignoring the structural deformations of the aircraft.

An experimental methodology was created for the validation of the developed model. A fixed-wing unmanned aircraft was built for flight testing. During flight the aircraft was filmed using two video cameras to capture its position in 3D space. A microcontroller onboard the aircraft recorded accelerometer, gyroscope and servomotor control signal data.

A computational aircraft model was created based on the unmanned aircraft used for flight testing. The inertial parameters were determined by weighing parts of the aircraft and calculating moments of inertia. The structural parameters were determined by applying a force on the structure on the aircraft, measuring the displacement and from this data calculating the stiffness of the torsion springs.

Using the initial positions of the aircraft determined from the video of the flight tests, individual parts of the flight tests were modelled. The calculated accelerations were compared with the ones measured experimentally during flight testing. The error of the aeroservoelastic model were much lower than those of the rigid model, but too high for the model to be useful without further improvements.

## Turinys

<b>Lentelių sąrašas</b> .....	<b>8</b>
<b>Paveikslų sąrašas</b> .....	<b>9</b>
<b>Įvadas</b> .....	<b>10</b>
<b>1. Aeroservotamprumo literatūros analizė</b> .....	<b>11</b>
1.1. Problemos aktualumas .....	11
1.2. Aeroservotamprumo analizės taikymas .....	12
1.2.1. Analizės taikymas manevravimo apkrovų mažinimui .....	12
1.2.2. Aktyvaus flaterio slopinimo optimizavimas .....	14
1.2.3. Aeroservotamprumo analizės taikymas flaterio elgesiui optimizuoti .....	19
1.3. Aeroservotamprumo analizės metodai .....	19
<b>2. Aeroservotamprumo modelio aprašymas</b> .....	<b>24</b>
2.1. Fizikinis modelis .....	24
2.1.1. Kinematinis modelis .....	24
2.1.2. Struktūrinis modelis .....	25
2.1.3. Aerodinaminis modelis .....	25
2.2. Standaus lėktuvo modelis .....	27
2.3. Modelio programinis įgyvendinimas .....	27
2.3.1. Kompiuterinio modelio aprašas .....	28
<b>3. Modelio validavimo metodika</b> .....	<b>30</b>
3.1. Skrydžio bandymų metodika .....	30
3.1.1. Bandymų orlaivis .....	30
3.1.2. Bandymo sąlygos .....	32
3.1.3. Duomenų rinkimas .....	32
3.2. Valdymo paviršių kalibravimo metodika .....	34
3.3. Skrydžio modeliavimo metodika .....	34
<b>4. Tyrimo rezultatai</b> .....	<b>36</b>
4.1. Valdymo paviršių kalibravimas .....	36
4.2. Atlikti skrydžio bandymai .....	36
4.2.1. Pirmasis analizuojamas fragmentas .....	36
4.2.2. Antrasis analizuojamas fragmentas .....	38
4.3. Modeliavimo rezultatai .....	39
4.3.1. Sudarytas lėktuvo modelis .....	39
4.3.2. Pirmojo atvejo modeliavimas .....	40
4.3.3. Antrojo atvejo modeliavimas .....	42
4.4. Modelio efektyvumas .....	44
<b>Išvados</b> .....	<b>45</b>
<b>Darbų sąrašas</b> .....	<b>46</b>
<b>Literatūros sąrašas</b> .....	<b>47</b>
<b>Priedai</b> .....	<b>51</b>
1 priedas. Aerodinaminių jėgų skaičiavimo kodas .....	51
2 priedas. Konferencijos „Transport Means 2021“ straipsnis .....	59

## Lentelių sąrašas

<b>1 lentelė.</b> Išskirtų aviacijoje naudojamų medžiagų mechaninės savybės [3–8] .....	11
<b>2 lentelė.</b> Lėktuvo modelio jungtį aprašantys kintamieji .....	28
<b>3 lentelė.</b> Lėktuvo modelio elementą aprašantys kintamieji.....	28
<b>4 lentelė.</b> Lėktuvo atskirų dalių masės ir jų padėtys (atstumai pateikti nuo sparno priekinės briaunos) .....	31
<b>5 lentelė.</b> Skrydžių bandymo metu renkami duomenys ir rinkimui naudojami prietaisai .....	33
<b>6 lentelė.</b> Lėktuvo modelio elementus aprašantys kintamieji .....	39
<b>7 lentelė.</b> Pirmojo atvejo modeliavimo sąlygos.....	40
<b>8 lentelė.</b> Antrojo atvejo modeliavimo sąlygos .....	42



## Paveikslų sąrašas

<b>1 pav.</b> Lėktuvų konstrukcijos masės dalis, kurią sudaro kompozitinės medžiagos, per laiką [2] ..	11
<b>2 pav.</b> Lėktuvo sparno apkrovos sumažinimas, išlaikant keliamąją jėgą (sumažinta apkrova pažymėta punktyrine linija) [12] .....	13
<b>3 pav.</b> Perkrovos manevro metu ir sparno lenkimo apkrovos palyginimas taikant ir netaikant apkrovos sumažinimo ir skrydžio valdymo valdiklius, esant $M = 0,4$ [13].....	13
<b>4 pav.</b> Sparno galo poslinkis su ir be aktyvaus flaterio slopinimo prie 18,0 m/s srauto greičio [18]	15
<b>5 pav.</b> Lenkimo momentas sparno šaknyje netaikant aktyvaus apkrovos mažinimo (punktyrinė linija), taikant statinės išvesties grįžtamojo ryšio metodą (ištinė linija) ir taikant tiesinį-kvadratinį Gauso/uždaro ciklo perdavimo atstatymo metodą (taškinė linija) [23] .....	17
<b>6 pav.</b> Keturių sparno laisvės laipsnių kitimas naudojant vieną valdymo paviršių (TE) ir du valdymo paviršius (LE&TE). $h$ – vertikalus poslinkis, $\alpha$ – kampinis poslinkis, $\beta$ – galinio paviršiaus atsilenkimas, $\gamma$ – priekinio paviršiaus atsilenkimas [25] .....	18
<b>7 pav.</b> Tipinės aeroservotamprumo sistemos blokinė schema [28].....	19
<b>8 pav.</b> Standartinė schema aeroservotamprumui nagrinėti [31].....	20
<b>9 pav.</b> Išmatuotos svyravimo amplitudės palyginimas su apskaičiuotos dažninės sintezės metodu [33] .....	22
<b>10 pav.</b> Dažninės sintezės metodu apskaičiuotos amplitudės palyginimas su apskaičiuotos Padé'o polinomų metodu [33] .....	22
<b>11 pav.</b> Blogiausia atitiktį turintis modelio validavimo atvejis ( $M = 1,2, H = 10000$ ft) [41].....	23
<b>12 pav.</b> Blogiausia atitiktį turintis modelio atsparumo testavimo atvejis ( $M = 1,3, H = 25000$ ft) [41] .....	23
<b>13 pav.</b> STM vienos plokštelės sūkurio pavyzdys [42] .....	26
<b>14 pav.</b> Sukurtos programos lango vaizdas .....	29
<b>15 pav.</b> Bandymams naudojamo lėktuvo gabaritiniai matmenys .....	30
<b>16 pav.</b> Bandymams naudoto lėktuvo sparno lonžerono schema .....	31
<b>17 pav.</b> Bandymams naudoto lėktuvo elektrinės sistemos blokinė diagrama.....	32
<b>18 pav.</b> Skrydžio duomenų failo trys pirmos eilutės.....	33
<b>19 pav.</b> Aerodinaminių paviršių atlenkimo priklausomybės nuo valdymo signalo trukmės priklausomybės kalibravimo schema .....	34
<b>20 pav.</b> Valdymo paviršių atlenkimo kampo priklausomybė nuo servovariklio gaunamos signalo trukmės.....	36
<b>21 pav.</b> Pirmojo analizuojamo skrydžio fragmento vaizdas (pateikta lėktuvo padėtis kas 0,5 s) ..	37
<b>22 pav.</b> Pirmojo analizuojamo skrydžio bandymo fragmento duomenys.....	37
<b>23 pav.</b> Antrojo analizuojamo skrydžio fragmento vaizdas (pateikta lėktuvo padėtis kas 0,25 s) ..	38
<b>24 pav.</b> Antrojo analizuojamo skrydžio bandymo fragmento duomenys .....	38
<b>25 pav.</b> Elementų ir jungčių schema modeliuojamame lėktuve.....	40
<b>26 pav.</b> Pirmojo atvejo modeliavimo pagreičiai .....	41
<b>27 pav.</b> Pirmojo atvejo modeliavimo pagreičių absoliutinė paklaida .....	41
<b>28 pav.</b> Pirmojo atvejo modeliavimo pagreičių absoliutinė paklaida .....	42
<b>29 pav.</b> Antrojo atvejo modeliavimo pagreičiai .....	43
<b>30 pav.</b> Antrojo atvejo modeliavimo pagreičių absoliutinė paklaida .....	43
<b>31 pav.</b> Antrojo atvejo modeliavimo pagreičių absoliutinė paklaida .....	44

## Įvadas

Konstruojant orlaivius nuolat siekiama kuo mažesnės konstrukcijos masės, todėl aviacijoje vis plačiau naudojamos kompozitinės medžiagos. Jos pasižymi labai geru stiprumo ir masės santykiu, dėl to yra labai gerai tinkamos lėktuvų konstrukcijoms.

Vis dėlto, nors kompozitinės medžiagos turi labai gerą stiprumo-masės santykį, palyginus su aviacijoje naudojamais metalo lydiniais ir mediena, jų maksimali galima deformacija yra didesnė, todėl kompozitiniai orlaiviai dažnai skrydžio metu patiria didesnes deformacijas už metalinius ir medinius. Kadangi mažesnio standumo konstrukcijos pasižymi žemesnio dažnio ir didesnės amplitudės svyravimais, dėl to gali stipriai nukentėti skrydžio charakteristikos.

Sumažėjęs konstrukcijos standumas ir žemesni savieji svyravimo dažniai taip pat leidžia prie mažesnių greičių įvykti flateriui. Siekiant išvengti flaterio esant žemesniam nei maksimaliam projektiniam orlaivio greičiui, reikia didinti konstrukcijos standumą, dėl to išauga lėktuvo masė ir iš kompozitinės konstrukcijos gaunama mažiau naudos, nei būtų galima tikėtis pagal pagerėjusį stiprumo-masės santykį.

Norint iš kompozitinių konstrukcijų išgauti didžiausią galimą naudą, reikia išspręsti aukščiau minėtas problemas, kylančias iš mažėjančio konstrukcijos standumo. Norint efektyviai analizuoti mažo standumo keliamas problemas yra būtinas geras tiriamo lėktuvo aeroservotamprumo (angl. *aeroservoelasticity*) modelis, nusakantis ryšį tarp aerodinamikos, struktūrinės deformacijos, ir valdymo sistemų.

Šiame darbe aprašomas siūlomas aeroservotamprumo skrydžio modelis, pilnai modeliuojantis lėktuvo judėjimą erdvėje. Siekiant įvertinti siūlomo modelio patikimumą, reikalingas eksperimentinis validavimas. Jis bus atliekamas vykdant eksperimentinius skrydžio bandymus su bepiločiu orlaiviu ir jų metu surinktus rezultatus palyginant su modelio atliktų skaičiavimų rezultatais. Taip bus įvertintas siūlomo modelio tikslumas.

Šio darbo tikslas - Sudaryti kompiuterinį aeroservotamprumo modelį ir atlikti jo eksperimentinį validavimą.

Darbo uždaviniai:

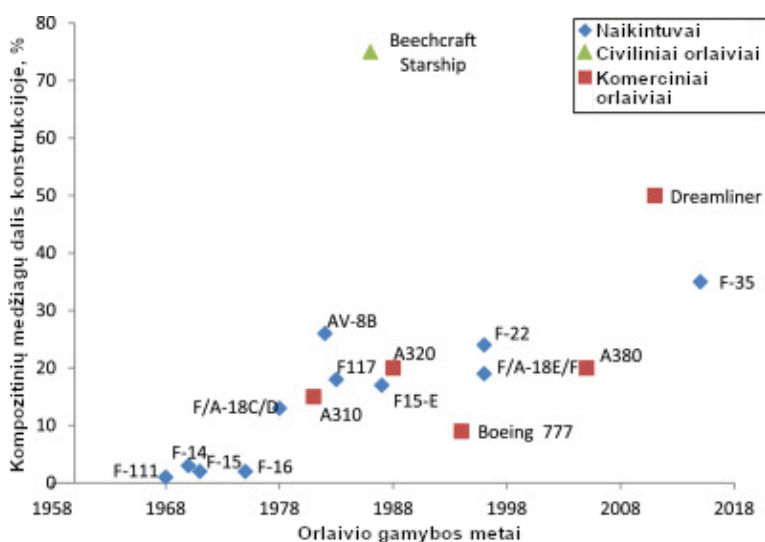
1. išanalizuoti aeroservotamprumo modelius ir jų taikymą,
2. sudaryti aeroservotamprumo fizikinį modelį,
3. sukurti kompiuterinę fizikinio aeroservotamprumo modelio versiją,
4. atlikti eksperimentinius lėktuvo skrydžio bandymus ir validuoti sukurtą aeroservotamprumo modelį,
5. atliekant surinktų duomenų analizę įvertinti aeroservotamprumo modelio veiksmingumą.

## 1. Aeroservotamprumo literatūros analizė

### 1.1. Problemos aktualumas

Lėktuvų konstrukcijose vis plačiau yra taikomos kompozitinės medžiagos. Pirmą fiksuotą sparnų orlaivių kategorija, kurioje šios medžiagos pradėtos naudoti plačiai, yra aukštos klasės sklandytuvai. Pirmasis sklandytuvas, kurio konstrukcijoje naudotos kompozitinės medžiagos buvo „Akaflied Stuttgart fs24 Phönix“, baigtas konstruoti 1957 metais. Septintajame dešimtmetyje tokios konstrukcijos buvo pradėtos taikyti plačiau, o aštuntajame pradėjo laimėti sklandymo varžybas [1]. Visi šiuo metu gaminami aukštos klasės sklandytuvai yra kompozitinės konstrukcijos.

Komercinėje ir karinėje aviacijoje kompozitinės konstrukcijos pradėtos taikyti vėliau, bet pagal 1 pav. galima matyti, kad einant laikui jos naudojamos vis plačiau. Pirmas pritaikymas buvo septintajame dešimtmetyje kariniuose orlaiviuose, bet sudarė mažiau nei 10% konstrukcijos masės. Einant laikui, sudaroma masės dalis didėjo, ir naujausiuose kariniuose lėktuvuose viršija 30%, o komerciniuose net 50%.



1 pav. Lėktuvų konstrukcijos masės dalis, kurią sudaro kompozitinės medžiagos, per laiką [2]

Kadangi anglies ir stiklo pluoštu sutvirtintų kompozitinių medžiagų standumo-stiprumo santykis yra mažesnis, nei prieš šias medžiagas aviacijoje plačiai naudotų (ir vis dar naudojamų) aliuminio lydinių (1 lentelė), tokį patį stiprumo atsargos koeficientą turinti kompozitinė konstrukcija yra mažiau standi nei pagaminta iš aliuminio lydinių. Kadangi pakankamas stiprumas yra pagrindinis struktūros tinkamumo kriterijus, kompozitinės konstrukcijos lėktuvai įprastai yra mažiau standūs nei aliumininės.

1 lentelė. Išskirtų aviacijoje naudojamų medžiagų mechaninės savybės [3–8]

Medžiagos pavadinimas	Stiprumo riba $\sigma_u$ , MPa	Tamprumo modulis $E$ , GPa	Santykis $E/\sigma_u$
Aliuminis 7075-T6	572	72	125,4
Aliuminis 6061-T6	310	69	222,3
442 anglies audinys – epoksidinė derva	710	51	71,83
450-5 anglies audinys – epoksidinė derva	730	51	69,86
92110 stiklo audinys – epoksidinė derva	335	19	56,72

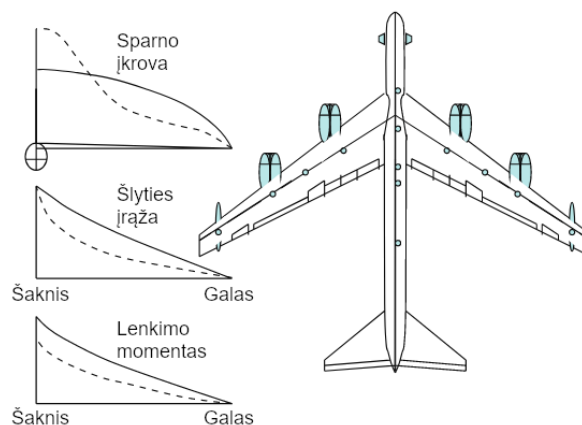
Pagal Waszak'ą ir Schmidt'ą [9], mažėjant lėktuvo standumui, jo konstrukcijos savojo svyravimo dažniai pradeda artėti prie lėktuvo kaip standaus kūno judėjimo erdvėje dažnių. Pagal Yen'o ir Swaim'o 1977 metų tyrimą, konstrukcijos žemo dažnio svyravimai ir jų sąveikavimas su kieto kūno judėjimo režimu labai neigiamai paveikia lėktuvo valdumą [10]. Bandytas vykdytas skrydžių simulatoriuje, naudojant bombonešio B-1 modelį. Skirtingiems atvejams tirti modelyje buvo nustatomi skirtingi lėktuvo pirmųjų dviejų tampriųjų svyravimų režimų savieji dažniai. Buvo nustatyta, kad žeminant svyravimų dažnį, ties tam tikra riba lėktuvas tampa dinamiškai nestabilus fugoido režime (pilotai įvertino kaip šiek tiek sunkesnę valdymo atvejį), o dažniui toliau mažėjant lėktuvas tampa nestabilus be svyravimų (ši atvejį pilotai įvertino kaip labai sudėtingą valdyti). Pirmo režimo dažnis turėjo žymiai didesnę įtaką valdumui, nei antro. Šis tyrimas parodė, kad žemo dažnio lėktuvo struktūros tamprieji svyravimai labai neigiamai paveikia jo valdumą.

## **1.2. Aeroservotamprumo analizės taikymas**

Šiame poskyryje bus apžvelgiami darbai, kuriuose aeroservotamprumo analizė taikoma skrydžio valdymo algoritmų kūrimui ir tobulinimui ir aktyvaus flaterio slopinimo tobulinimui. Darbai manevringumo srityje daugiausia sutelkti į aukšto manevringumo karinius orlaivius, bet taip pat yra pritaikymų ir lėtesniems mažo standumo orlaiviams su žemo dažnio struktūros svyravimais. Karinių orlaivių tyrimuose pagrinde atsižvelgiama į stabilumo išlaikymą ekstremaliomis skrydžio sąlygomis ir struktūros apkrovimo mažinimą, nekenkiant orlaivio manevringumui. Mažo standumo orlaiviams siekiama sumažinti žemo dažnio svyravimų ryšį su lėktuvo judėjimu erdvėje, neprastinant valdumo. Flaterio slopinimo tyrimai daugiausiai fundamentiniai, sutelkti į naujų algoritmų kūrimą ir esamų tobulinimą, taikant juos labai paprastam dviejų dimensijų aeroservotamprumo modeliui. Kai kurie darbai siekia taikyti flaterio slopinimą sudėtingesniems atvejams, atitinkantiems visą lėktuvą.

### **1.2.1. Analizės taikymas manevravimo apkrovų mažinimui**

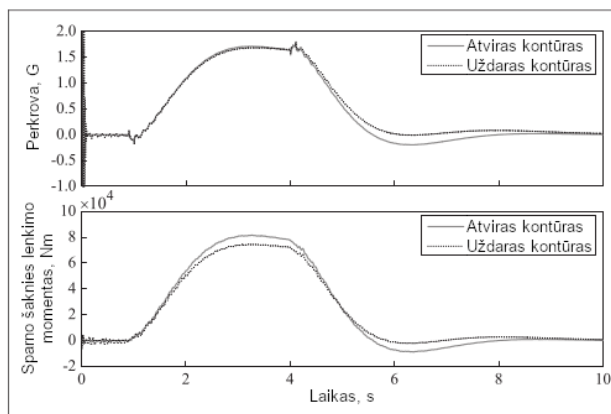
Vienas iš aerotamprumo analizės taikymo būdų yra siekiant sumažinti manevravimo metu patiriamas apkrovas (ilustruota 2 pav.). Dažniausiai tai taikoma kariniams orlaiviams, kaip pavyzdžiui Brenner'io 1993 m. techniniame memorandume, kuriame autorius aprašė aeroservotamprumo analizės taikymą karinių orlaivių valdymo algoritmų stabilumui prie didelių atakos kampų ir mažų greičių nustatyti [11]. Buvo taikyti žemos ir aukštos eilės aerodinaminiai ir orlaivio sistemų modeliai ir lyginant su eksperimentiniais duomenimis nustatyta tikslumo priklausomybė nuo naudojamo modelio detalumo. Autorius numato, kad tokiose skrydžio sąlygose įprastinis žemos eilės valdymo sistemos modeliavimas gali būti nebepakankamai tikslus, nes prie mažesnių greičių esant silpnesnėms aerodinaminėms jėgoms, valdiklių veikimas turi sąlyginai didesnę įtaką. Buvo nustatyta, kad žemos eilės valdymo sistemos modelis gali būti netinkamas ne tik didelių atakos kampų ir mažų greičių sąlygomis, bet gali duoti nepatikimus stabilumo parodymus bet kokiomis skrydžio sąlygomis. Priklausomybė nuo aerodinaminio modelio eilės buvo mažesnė, išskyrus kelias didelio nuokrypio anomalijas.



**2 pav.** Lėktuvo sparno apkrovos sumažinimas, išlaikant keliamąją jėgą (sumažinta apkrova pažymėta punktyrine linija) [12]

Kitą panašią analizę atliko Li ir kt. savo 2019 m. straipsnyje siekia sukurti patikimą manevravimo apkrovų sumažinimo sistemą kariniam naikintuvui, efektyviai veikiančią plačiame ruože skrydžio sąlygų [13]. Tokia problema iškeliama, nes ankstesnės tokios sistemos veikė tik siaurame Macho skaičių ruože arba reikalavo didelių modifikacijų platesniam veikimui. Kad sumažinant manevravimo apkrovas nenukentėtų lėktuvo manevringumas, kartu kuriama ir rekursiniais neuroniniais tinklais paremta skrydžio valdymo sistema. Aerodinaminių apkrovų mažinimo sistema kuriama naudojant H-begalybės metodą.

Aprašant lėktuvo aeroservotamprųjį modelį, klasikiniu būdu sukuriama keli modeliai prie skirtingų diskrečių Macho skaičiaus verčių, tada interpoliuojant tarp jų sukuriama galutinis tolydus modulis. Taip pat pasitelkiamos tiesinės dalinės transformacijos, taip siekiant kuriamus valdiklius padaryti atsparius aeroservotampriojo modelio neužtikrintumams. Sukurti valdikliai buvo išbandyti naudojant tą patį modelį, koks buvo naudotas ir jų sukūrimui. Bandymo rezultatuose, pateiktuose 3 pav., galima matyti, kad naudojant autorių sukurtus valdiklius, perkrova manevro pradžioje beveik nepakinta (t.y. lėktuvo manevringumas nesumažėja), bet lėčiau grįžta į nusistovėjusią po manevro, taigi išėjimas iš manevro gali būti lėtesnis. Maksimali sparno lenkimo apkrova sumažėja nuo maždaug 80 kNm iki maždaug 70 kNm. Panašūs rezultatai matomi ir prie kitų Macho skaičių.



**3 pav.** Perkrovos manevro metu ir sparno lenkimo apkrovos palyginimas taikant ir netaikant apkrovos sumažinimo ir skrydžio valdymo valdiklius, esant  $M = 0,4$  [13]

Susijusi aeroservotamprumo analizės taikymo sritis yra lėktuvų manevringumo didinimui. Livet ir kt. savo 1994 m. straipsnyje nagrinėja ne karinių orlaivių, bet bendrai mažo standumo lėktuvams kylančią problemą, kai dėl mažėjančio lėktuvų standumo jų struktūrinio svyravimo dažniai artėja prie lėktuvo judesių erdvėje dažnių, dėl to taikant klasikinę virpesių slopinimo metodiką gali prastėti lėktuvo valdymas [14]. Autoriai siekia sukurti lėktuvui valdymo sistemą, leidžiančią išsaugoti lėktuvo valdumą ir slopinančią elastinį lėktuvo nestabilumą. Tam pasitelkiamas tiesinio-kvadratinio regulatoriaus valdymo modelis, optimizuojamas iteraciškai. Tuo metu ši metodika jau buvo taikoma lėktuvų valdymo algoritmams kurti, atsižvelgiant į valdymo vėlinimą ir efektyvumą, bet dar nebuvo pritaikyta atsižvelgiant į struktūros elastingumą. Sukurtas valdymo algoritmas efektyviai slopino elastinį lėktuvo nestabilumą, nepakenkdamas valdymo charakteristikoms.

Verta paminėti aeroservotamprumo analizės taikymą ir kosminių transporto priemonių valdymui. Mao ir kt. 2018 straipsnyje aprašo siekį sukurti daugkartinio panaudojimo kosminės raketos valdymo sistemą, skirtą palaikyti kryptingai grįžtant į atmosferą [15]. Tokiai valdymo sistemai sukurti autoriai taiko aeroservotamprumo modelį, darydami prielaidą, kad erdvėlavis patiria tik išilgines lenkimo deformacijas.

Valdymo sistema sudaroma iš kelių dalių. Pagrindines valdymo komandas sukuria du slenkančiojo režimo valdikliai ("sliding mode controller"), iš kurių vienas valdo kampinę padėtį, o kitas kampinį greitį. Tokio tipo valdikliai naudojami dėl jų gero pritaikomumo plačiame skrydžio sąlygų ruože. Šių valdiklių išvestis papildomai yra apdorojama antrojo tipo neaiškios logikos sistemomis, siekiant geriau įvertinti sistemos netiesiškumą. Taip pat sistemoje yra įdiegtas raketos aeroservotamprumo modelis, kuriuo siekiamas nuspėti tolesnis jos elgesys ir kompensuojami vėlavimai. Sistema optimizuojama kaip tikslą naudojant kuo mažesnę nukrypimą nuo reikiamos kampinės padėties erdvėje.

Palyginus su įprastine valdymo sistema, paremta slenkančio režimo valdikliais, autorių sukurta sistema grįžimo į atmosferą metu užtikrino mažesnę erdvėlavio maksimalų kampinį greitį (48 °/s vietoje 82 °/s). Taip pat buvo pasiektas tikslesnis reikiamos kampinės padėties išlaikymas, naudojant silpnesnius padėties koregavimo signalus.

Aeroservotamprumo analizė būtina ir norint pritaikyti kai kuriuos visiškai naujus kontrolės metodus. *Fly-by-feel* koncepcija nusako lėktuvo paviršiuje paskirstytų jutiklių, leidžiančių nustatyti bendrą srauto pobūdį apie orlaivį naudojimą skrydžio valdymui, vietoj dabar įprastos sistemos, kai naudojami keli taškiniai jutikliai [16]. Savo 2014 metų straipsnyje Mangalam'as ir Brenner'is aprašo *Fly-by-feel* koncepcinę versiją, jos veikimą demonstruodami aerodinaminio vamzdžio eksperimentais [17]. Tyrimas unikalus tuo, kad aerodinaminėms jėgoms, veikiančioms sparną apskaičiuoti, kaip jutiklių duomenis naudoja tik priekinės briaunos stagnacijos taško padėtį ir jo judėjimo greitį.

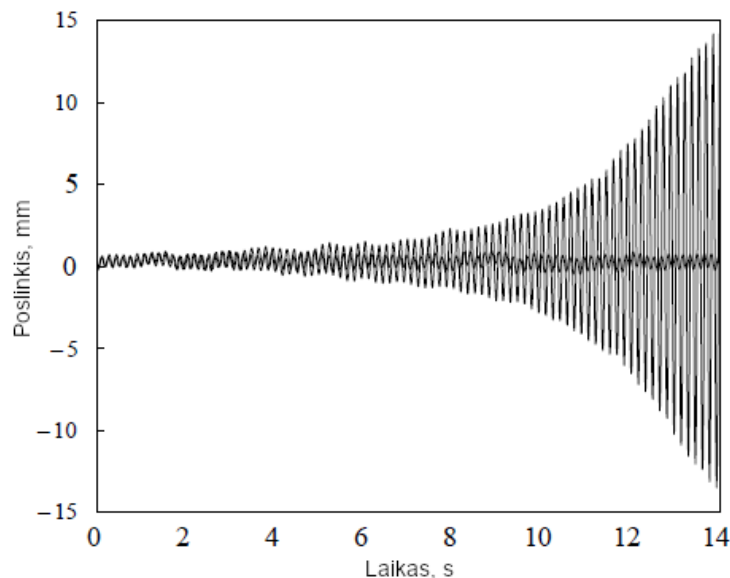
### **1.2.2. Aktyvaus flaterio slopinimo optimizavimas**

Didžioji dalis mokslinių darbų apie aeroservotamprumą ieško būdų jį panaudoti efektyvesniam flaterio slopinimui ir vėjo gūsių apkrovų mažinimui. Dalis autorių atlieka labiau fundamentinius tyrimus, kurdami algoritmus dvimačiam profiliui su vienu arba dviem valdymo paviršiais, kitų darbas yra labiau taikomasis, tiriantis flaterio slopinimo pritaikymą sparnui ar net visam orlaiviui. Dauguma autorių labai didelį dėmesį skiria valdymo algoritmo atsparumui ("robustness"), tai yra jo galimybei veikti sąlygomis, kurios skiriasi nuo tų, kurioms jis buvo sukurtas. Tai yra svarbus parametras, nes

skrydžio sąlygos ne visada atitinka scenarijus, kuriais veikė modelis, kai buvo kuriamas valdymo algoritmas, ir labai svarbu kad tokiomis nenumatytomis sąlygomis algoritmas duotų bent jau nuspėjamą ir patenkinamą, jei ir ne optimalų, rezultatą.

Borglund'as ir Kuttenukeuler'is 2002 m. aprašė aktyvaus flaterio slopinimo optimizavimo bandymą erdviniam sparnui [18]. Slopinimui naudotas sparno galinėje briaunoje esantis valdymo paviršius, einantis per dalį sparno ilgio. Sparnui buvo sudarytos judesio lygtys ir dažniais apibrėžtas dvimatis nenusistovėjęs srauto aerodinamikos modelis. Dvimačio modelio pasirinkimas pagrįstas tuo, kad sparnas yra didelio proilgio, dėl to trimačio srauto efektai sąlyginai maži. Eleroną valdo servo motoras, aprašytas atsižvelgiant į judesio slopinimą ir signalo vėlinimą. Sparno padėties fiksavimui ir grįžtamojo ryšio sukūrimui naudoti lazeriniai atstumo jutikliai.

Aerodinaminiam vamzdyje buvo atlikti savųjų svyravimų, flaterio nevaldant elerono ir elerono valdymo fiksuotu dažniu atsako bandymai. Pagal šių bandymų rezultatus pakoregavus aeroservotampumo modelį, juo naudojantis buvo sudarytas ir optimizuotas valdymo algoritmas, turėjęs slopinti pirmąjį flaterio režimą. Šio algoritmo išbandymas oro vamzdyje buvo sėkmingas – be jo buvo flateris pasireiškė ties 15,4 m/s, pradėjus taikyti algoritmą jis buvo atidėtas iki 23,0 m/s. Sparno galo poslinkio palyginimas taikant ir netaikant aktyvaus flaterio slopinimo esant 18,0 m/s srauto greičiui pateiktas 4 pav.. Galima matyti, kad ties tokiu greičiu nenaudojant aktyvaus slopinimo flateris pasireiškia maždaug per 5 s, o aktyvaus flaterio slopinimo panaudojimas leidžia beveik visiškai jį sustabdyti. Autoriai atkreipia dėmesį į tai, kad toks efektyvus slopinimas greičiausiai buvo galimas dėl palyginti silpno žemiausio flaterio režimo. Taip pat verta turėti omenyje, kad eksperimento metu naudotas lazerinis sparno padėties nustatymo būdas nėra pritaikytinas realiame orlaivyje.



**4 pav.** Sparno galo poslinkis su ir be aktyvaus flaterio slopinimo prie 18,0 m/s srauto greičio [18]

Zhang'as ir Ye savo 2007 m. straipsnyje išskiria problemą, kad klasikiniai tiesiniai aerotampumo modeliai nėra taikytini transgarsiniuose skrydžio režimuose, todėl flaterio slopinimo algoritmų veikimas yra apribotas ikigarsiniam skrydžiui [19]. Darbe buvo pasinaudota tuo metu naujomis kompiuterių galimybėmis ir naudotas tiesioginis Eulerio lygtimis paremtas CFD aeroelastinis modelis. Bandymo metu flaterio slopinimo algoritmai buvo optimizuojami prie skirtingų Macho skaičių. Buvo nustatyta,

kad ikigarsiniuose greičiuose flaterio slopinimo algoritmo optimalūs koeficientai yra silpnai priklausomi nuo Macho skaičiaus, bet transgarsiniame ruože (šiam straipsnyje nuo  $M = 0,8$ ) optimalūs koeficientai labai stipriai priklauso nuo Macho skaičiaus, dėl didelės vibracijų įtakos smūginių bangų susidarymui. Pasak autorių, tuo parodoma atskiro optimizavimo transgarsiniame ruože svarba. Taikytas slopinimo algoritmas greitį, prie kurio vyksta flateris, padidino apie 20%.

Fan'as ir kt. savo 2010 m. straipsnyje aprašo flaterio slopinimo ir orlaivio valdymo pagerinimo algoritmo kūrimą, taikytiną dideliems kroviniams lėktuvams [20]. Iškeliama problema, kad lėktuvų su didelio proilgio sparnais aerotamprumo žemų dažnių režimai susisieja su lėktuvo judėjimu erdvėje, ir vienas kitą veikia. Tai daro poveikį lėktuvo valdymui ir skrydžio savybėms.

Dėl šios priežasties atlikdami aeroservotamprumo analizę autoriai atsižvelgė į lėktuvo judėjimo erdvėje ir aerotamprumo tarpusavio poveikį. Kuriamas valdymo algoritmas paremtas tiesiniu grįžtamuju ryšiu ir neaiškia logika (angl. *fuzzy logic*), taip siekiant jį padaryti labiau pritaikomą skirtingoms skrydžio sąlygoms. Sukurtas algoritmas sėkmingai slopina žemo dažnio svyravimus.

Savo 2012 m. straipsnyje Gang'as ir kt. aprašo bandymą sudaryti lėktuvo flaterio slopinimo sistemą naudojant tikrinio statmenojo išskirstymo (angl. *proper orthogonal decomposition* – POD) redukuotos eilės modelyje (ROM) metodą [21]. Toks metodas didesnio detalumo, nei klasikiniai aeroservotamprumo analizei taikomi tiesiniai aerodinamikos modeliai. Kadangi įprastiniai POD laiko-domeno modeliai yra per aukštos eilės, taikomas balansuoto apribojimo algoritmas, taip gaunamas POD-BT/ROM modelis.

Sukurtas modelis buvo validuotas tiriant aerodinaminį profilį ir du skirtingus sparnus. Profilio atveju nustatyta, kad 80-os eilės POD/ROM modeliu buvo išgautas toks tikslumas, kaip taikant netiesinį CFD, kai tuo tarpu taikant POD-BT/ROM tam užteko tik 5-os eilės modelio. Tiriant pirmą sparną buvo atitinkamai nustatyta, kad 300-os eilės POD/ROM modelis ir 40-os eilės POD-BT/ROM modelis atitinka CFD rezultatus. CFD analizė užtruko 6 val., kai tuo tarpu POD/ROM analizė truko tik 2 min. Antram sparnui buvo bandoma rasti flaterio pradžios greitį naudojant CFD ir 40-os eilės POD-BT/ROM modelį. CFD skaičiavimai truko apie 20 val., o POD-BT/ROM truko kelias minutes, abiem būdais gauti rezultatai sutapo.

Savo 2015 m. straipsnyje Brillante ir Mannarino aprašo bandymą naudoti neuroninius tinklus aktyviam flaterio slopinimui transgarsiniuose skrydžio režimuose [22]. Iškeliama idėja, kad tokiu pagrindu sukurtus slopinimo algoritmus bus greičiau ir paprasčiau pritaikyti veikti prie skirtingų skrydžių sąlygų. Tiriamo sparno aerodinaminiai duomenys surinkti oro vamzdžio bandymais ir pasitelkiant kompiuterinę aerodinaminę analizę. Grįžtamajam ryšiui gauti yra naudojami du akselerometrai – vienas ant sparno priekinės briaunos, o kitas ties eleronu vyriu. Neuroninio tinklo treniravimui naudojami du veikimo scenarijai – atsako į pradines sąlygas ir atsako į valdymo impulsus. Vėlesniam valdymo algoritmo įvertinimui pasitelkiami papildomi scenarijai – atsitiktinio srauto greičio kitimo, mažų variacijų modelyje, pavėluoto slopinimo aktyvavimo. Nors buvo treniruotas tik naudojant du pirmuosius scenarijus, algoritmo rezultatai buvo labai geri ir papildomuose scenarijuose. Bandymų metu nustatyta kad sparno flaterio greitis padidėjo nuo 108,58 m/s iki 134 m/s.

Fonte ir kt. savo 2015 m. straipsnyje aprašo statinės išvesties grįžtamojo ryšio (SOF) tipo valdiklio kūrimą, kurio paskirtis yra gūsių apkrovos mažinimas [23]. Tokio tipo valdiklis priima seriją įvesties



parametrų, jų vektorių padaugina iš stiprinimo matricos, ir gautą rezultatą perduoda kaip išvestį. Valdiklio kūrimo esmė yra geriausių stiprinimo matricos verčių parinkimas. Tokio tipo valdikliai yra labai paprasti, ir dėl to tinkami sudėtingoms sistemoms su daug įvesčių ir išvesčių, taip pat kaip atsarginiai valdikliai, perimantys valdymą pagrindiniam nustojus veikti. Numatomas valdiklio pritaikymas yra naujos kartos regioniniam transportiniam lėktuvui.

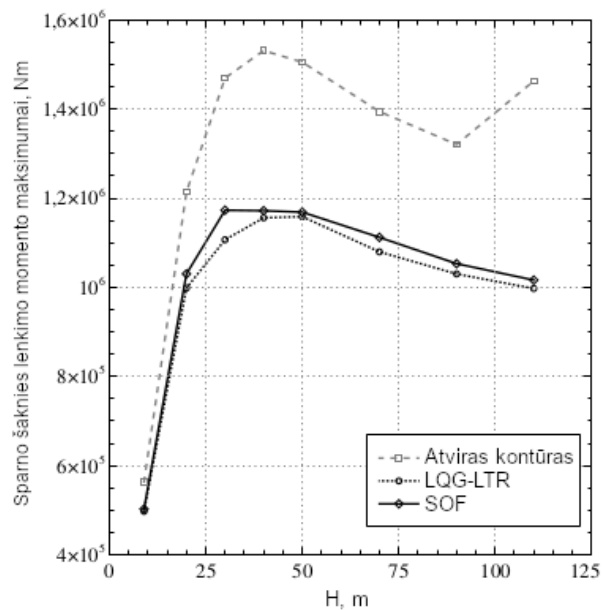
Pradinei aerotampriajai lėktuvo analizei buvo taikytas plokštelinis aerodinamikos ir strypinis struktūrinis modeliai. Kuriant valdiklį valdymo paviršių veikimas modeliuojamas pirmos eilės perdavimo funkcija, o analizuojant jo veikimą naudojama antros eilės funkcija. Vėjo gūsių simuliacijai naudojama EASA rekomenduojama  $1 - \cos$  formulė:

$$v_g = \begin{cases} \frac{U_g}{2} [1 - \cos(\frac{\pi s}{H})] & \text{kai } 0 < s \leq 2H, \\ 0 & \text{kai } s > 2H, \end{cases} \quad (1.1)$$

čia  $v_g$  – gūsio lokalus greitis,  $U_g$  – gūsio maksimalus greitis,  $s$  – gūsio prasiskverbimo atstumas,  $H$  – gūsio gradientas.

Kaip valdiklio optimizavimo tikslai parenkamos apkrovos lėktuvo struktūroje, didžiausią įtaką suteikiant apkrovai sparno šaknyje. Kaip papildomi tikslai nustatomi ir skrydžio komfortas, lėktuvo valdymo charakteristikos. Duomenys valdikliui paimami iš keturių akselerometrų, išdėstytų sparne, dviejų ant horizontalaus stabilizatoriaus ir trijų lėktuvo liemenyje. Taip pat naudojami duomenys apie valdymo paviršių atsilenkimą ir bendrą lėktuvo padėtį erdvėje.

5 pav. pateiktas straipsnyje sukurto valdiklio veiksmingumo palyginimas su sudėtingesniu tiesiniu-kvadratinu Gauso/uždaro ciklo perdavimo atstatymo (LQG/LTR) valdikliu, įprastai taikomu ir gerai veikiančiu tokiais atvejais.



**5 pav.** Lenkimo momentas sparno šaknyje netaikant aktyvaus apkrovos mažinimo (punktirinė linija), taikant statinės išvesties grįžtamojo ryšio metodą (ištisinė linija) ir taikant tiesinį-kvadratinį Gauso/uždaro ciklo perdavimo atstatymo metodą (taškinė linija) [23]

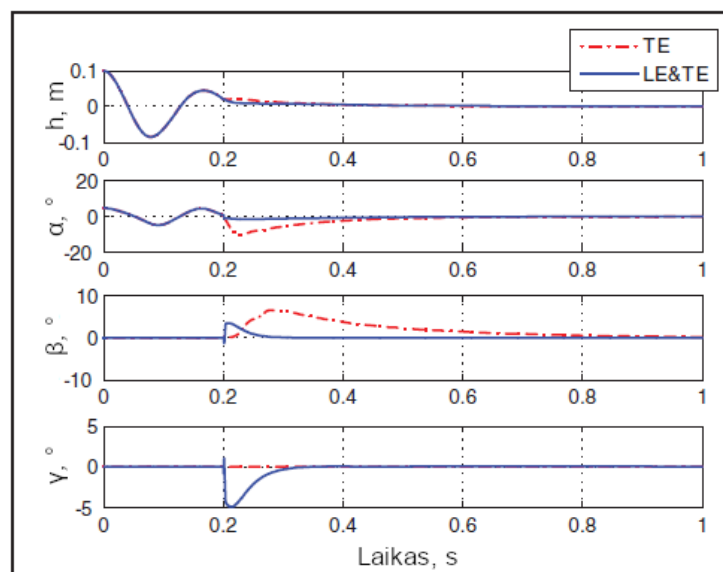
5 pav. matyti, kad lenkimo momentas taikant SOF yra didesnis, nei taikant LQG/LTR, bet mažesnis nei netaikant jokio aktyvaus apkrovos mažinimo. Abiejų valdiklių efektyvumas yra panašus, straipsnio autoriai tokį rezultatą laiko teigiamu, nes SOF valdiklis yra žymiai paprastesnis už LQG/LTR.

Zhang'as ir Behal'is savo 2016 m. straipsnyje aprašo bandymą sukurti flaterio slopinimo algoritmą, veikiantį esant nepastoviam srautui su iš anksto nežinomais trukdžiais [24]. Algoritmas kuriamas dvimačio srauto aplink aerodinaminį profilį, turintį du judėjimo laisvės laipsnius (polinkis ir vertikali padėtis) atvejui. Slopinimui naudojamas vienas valdymo paviršius, esantis ant galinės sparno briaunos. Kaip vienintelė valdymo algoritmo įvestis yra naudojamas sukimo momentas, susidarantis valdymo paviršiaus vryje.

Į aeroservotamprumo modelį buvo įvesti trys ribotos neapibrėžties dėmenys, aprašantys tris eiles ieškomo polinkio neatitikimo su realiu. Pagal šį modelį buvo suprojektuotas valdymo algoritmas. Vėliau jis buvo išbandytas skirtingiems atvejams, mažesniuose ir didesniuose nei flaterio greičiuose, esant įvairiems vėjo gūsių profiliams. Bandymai parodo valdymo sistemos gebėjimą visiškai slopinti polinkio svyravimus, bet ne vertikalios padėties svyravimus. Autoriai numato, kad tam reikalingas slopinimas dviem valdymo paviršiais.

Zhang'as ir kt. 2017 m. aprašė atliktą flaterio slopinimo algoritmo optimizaciją, slopinimui naudojant valdymo paviršius esančius ant priekinės ir galinės sparno briaunos [25]. Analizuotas dvimatis modelis, turintis keturis laisvės laipsnius – vertikalus poslinkis, pasisukimo kampas, priekinės ir galinės briaunos valdymo paviršių atlenkimo kampai. Šiuo atveju valdymo sistema buvo "kelių įvadų – kelių išvadų" (MIMO – multiple-input multiple-output) tipo. Valdymo algoritmo tikslas yra sistemą asimptotiškai artinti prie pusiausvyros būsenos.

Sukurtos valdymo sistemos efektyvumas buvo išbandytas naudojant kompiuterinį modelį. 6 pav. pavaizduotas atsako skirtumas tarp sistemos su vienu valdymo paviršiumi galinėje briaunoje ir sistemos su dvejais valdymo paviršiais, viršijant flaterio greitį.



**6 pav.** Keturių sparno laisvės laipsnių kitimas naudojant vieną valdymo paviršių (TE) ir du valdymo paviršius (LE&TE).  $h$  – vertikalus poslinkis,  $\alpha$  – kampinis poslinkis,  $\beta$  – galinio paviršiaus atsilenkimas,  $\gamma$  – priekinio paviršiaus atsilenkimas [25]

6 pav. galima matyti, kad sistema su dviem valdymo paviršiais žymiai greičiau sugrįžta į pradinę padėtį, ir kad reikalingas mažesnis (bet staigesnis) galinio paviršiaus atsilenkimas tam sugrįžimui pasiekti. Tai parodo efektyvesnę stabilizaciją, naudojant du valdymo paviršius.

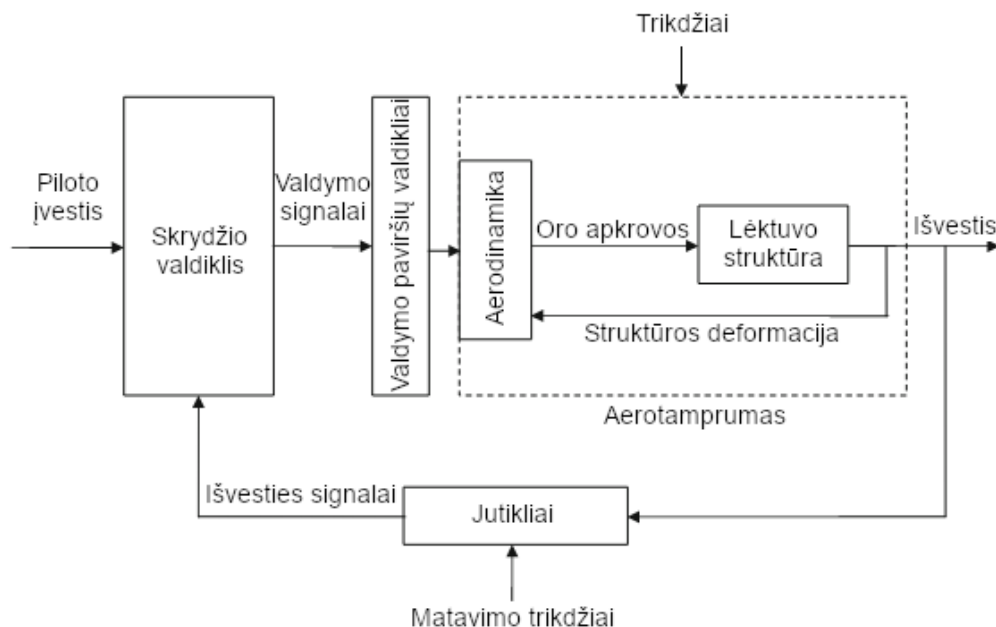
### 1.2.3. Aeroservotamprumo analizės taikymas flaterio elgesiui optimizuoti

Tai yra turbūt labiausiai klasikinė aeroservotamprumo analizės taikymo sritis. Kadangi ji netiesiogiai siejasi su baigiamojo darbo tematika, čia ji bus apžvelgiama tik labai trumpai. Vienas iš tokių pavyzdžių yra Patil'o 2004 m. straipsnis, kuriame nagrinėjamos dėl flaterio padidėjusio pasipriešinimo pasekmės [26]. Autorius pademonstruoja, kad dėl flaterio padidėjęs pasipriešinimas susieja aeroservotamprumą su orlaivio greičiu, ir dėl to vykstantis flateris sukelia orlaivio greičio svyravimus.

Arena ir kt. 2021 m. išleido straipsnį, kuriame aprašomas priekinės briaunos prietaiso skirto mažinti krovinių lėktuvų tūpimo triukšmui projektavimas [27]. Jame pasitelkiama aeroservotamprumo analizė, siekiant įsitikinti kad prietaisas neįeis į flaterį.

### 1.3. Aeroservotamprumo analizės metodai

Šiame skyriuje apžvelgiami skirtingi metodai aeroservotamprumo analizei atlikti. Pradžioje apžvelgiami pamatiniai šios srities darbai, taip pat aprašomi ir naujausi pasiekimai aeroservotamprumo srityje. Aeroservotamprumas aprašo sistemą, sudarytą iš abipusiai susietų deformuojamos orlaivio geometrijos ir aerodinaminių apkrovų, taip pat atsižvelgiant į valdymo paviršių įtaką, valdymo signalus (7 pav.).



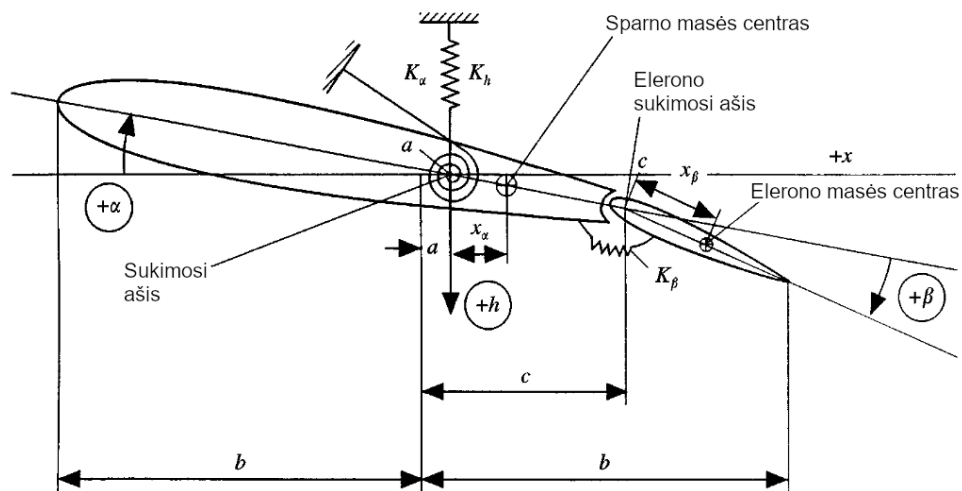
7 pav. Tipinės aeroservotamprumo sistemos blokinė schema [28]

Kadangi net ir esant standžiai lėktuvo struktūrai, flateris gali įvykti dėl nesubalansuotų valdymo paviršių, lėktuvų aeroservotamprumas analizuojamas jau nuo XX a. pradžios [29]. Vienas iš pamatinių darbų šioje srityje yra 1934 m. Theodorsen'o išleistas straipsnis, kuriame analitiškai

išsprendžiamas profilio su užsparniu flaterio uždavinys [30]. Nagrinėjama sistema turi tris laisvės laipsnius – vertikalią profilio padėtį, atakos kampą ir užsparnio atlenkimo kampą. Tokia sistema (pavaizduota 8 pav.) yra standartinė, nagrinėjama daugumoje fundamentinių aeroservotamprumo tyrimų ir dabar. Autorius aerodinaminį profilį supaprastino, darydamas prielaidą kad jo aerodinaminė forma yra apskritimas, jį veikiančias jėgas ir momentus skaičiavo integruodamas slėgį per apskritimo kreivę. Profilio ir užsparnio judėjimą autorius aprašė diferencialinėmis lygtimis, vėliau jas išsprendęs sudarė formulę greičiui, prie kurio vyksta flateris apskaičiuoti:

$$v = F \frac{b\omega_r r_r}{\sqrt{\kappa}}, \quad (1.2)$$

čia  $b$  – pusė profilio stygos;  $\kappa$  – santykis tarp masės oro cilindro, kurio skersmuo lygus sparno stygai, ir sparno masės;  $\omega_r$  – savasis svyravimo dažnis (reikia nustatyti kuris iš trijų svarbiausias);  $r_r$  – savųjų svyravimų spindulys (reikia nustatyti kuris iš trijų svarbiausias);  $F$  – flaterio faktorius, randamas sprendžiant tiesinę lygčių sistemą [30]. Prieš Antrąjį pasaulinį karą paprastos aeroservotamprumo problemos jau buvo gerai suprastos ir reguliariai sprendžiamos projektuojant lėktuvus [29].



8 pav. Standartinė schema aeroservotamprumui nagrinėti [31]

Kartu su fundamentiniais tyrimais, buvo atliekamas ir programinės įrangos, skirtos analizuoti aeroservotamprumo reiškinius, tobulinimas. Vienas iš pavyzdžių yra 1989 m. Gupta'os ir kt. išleistas straipsnis, kuriame aprašomas NASA naudotos struktūrinės analizės programinės įrangos STARS išplėtimas, kad ja būtų galima atlikti aeroservotamprumo analizę [32]. Šis programos plėtinys pademonstruotas atliekant eksperimentinio kompozitinio lėktuvo su neigiamu strėliškimu X-29A analizę, taip dar kartą parodomas aeroservotamprumo tyrimų kompozitiniams lėktuvams aktualumas.

Keli būdai atlikti aeroservotamprumo analizę buvo apžvelgti Zimmermann'o [33] jo 1991 straipsnyje. Jis nusakė būtinybę į skaičiavimus įtraukti lėktuvo valdymo sistemos elgesį tais atvejais, kai jos negalima aprašyti kaip paprastos spyruoklės-slopintuvo sistemos. Jis išskiria du pagrindinius aktyvių valdymo sistemų tipus:

1. aktyvios skrydžio valdymo sistemos, naudojamos valdyti "atlaisvinto stabilumo" lėktuvą, kad būtų išgautas didesnis manevringumas;
2. aktyvios valdymo sistemos, skirtos slopinti flaterį, sumažinti gūsinės ir manevravimo apkrovas,

pagerinti skrydžio komfortą.

Straipsnyje aprašomi trys skirtingi būdai aeroservotamprumui analizuoti. Pirmasis iš jų remiasi klasikiais flaterio aprašymo metodais. Lėktuvas visų pirma aprašomas kaip energetiškai uždara sistema. Kad tokia sistema susidarytų, aktyvios skrydžio valdymo sistemos dalys atjungiamos, pakeičiamos visiškai standžiomis arba pakeičiamos fiksuoto standumo elementais. Tokio elemento standumas turi atitikti pasyvaus valdiklio standumą. Taip modifikavus lėktuvą, jam gali būti atliekami žemės vibraciniai testai, kurių metu nustatomi savojo svyravimo režimų dažniai. Vėliau tokia sistema aprašoma matematiškai ir praplečiama nariais, aprašančiais lėktuvo skrydžio valdymo sistemos poveikį. Bendra aeroservotamprumo lygtis įgyja formą [33]

$$\left( s^2 M + sD + K + \frac{1}{2} \rho v^2 Q(Im(s)) + \Phi_P^t H(s) \Phi_M \right) q = 0, \quad (1.3)$$

kurios pirmieji keturi nariai yra iš neišplėtos aerotamprumo lygties,  $M$ ,  $D$ ,  $K$ ,  $Q$  yra atitinkamai masių, slopinimų, standumų ir nepastovių aerodinaminių apkrovų matricos.  $\Phi_P$  aprašo skrydžio valdymo sistemos valdiklių jėgas,  $\Phi_M$  – skrydžio valdymo sistemos jutiklių duomenis, o  $H(s)$  yra skrydžio valdymo sistemos perdavimo funkcija.  $q$  nusako bendras sistemos koordinatas.

Antrasis analizės metodas yra išvestas iš pirmojo, perrašius jį iš būsenos formos į būsenos-erdvės formą. Tokioje formoje negalima aprašyti nepastovių aerodinaminių jėgų, bet vietoj to aprašomas visas lėktuvas kaip bendra dinaminė sistema. Šis metodas aprašomas lygtimi

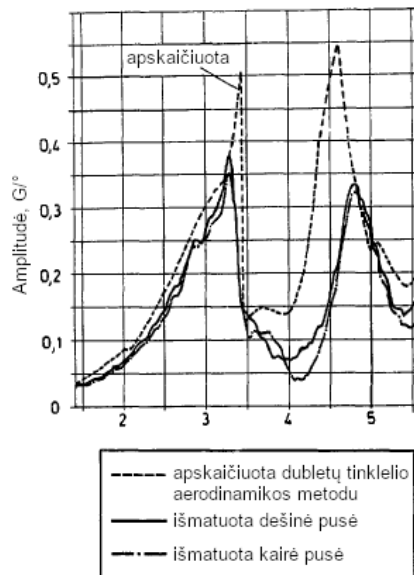
$$\begin{bmatrix} sq \\ s^2 q \end{bmatrix} = T \begin{bmatrix} \lambda_F & 0 \\ 0 & \lambda_{F^*} \end{bmatrix} T^{-1} \begin{bmatrix} q \\ sq \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1} \Phi_P^t \end{bmatrix} F. \quad (1.4)$$

Šios lygties dešinės pusės pirmasis narys pakeičia 1.3 lygties pirmuosius keturis narius. Palyginus su praeita, ši lygtis pranašesnė tuo, kad leidžia atsižvelgti į skrydžio sistemos valdiklių nestabilumus.

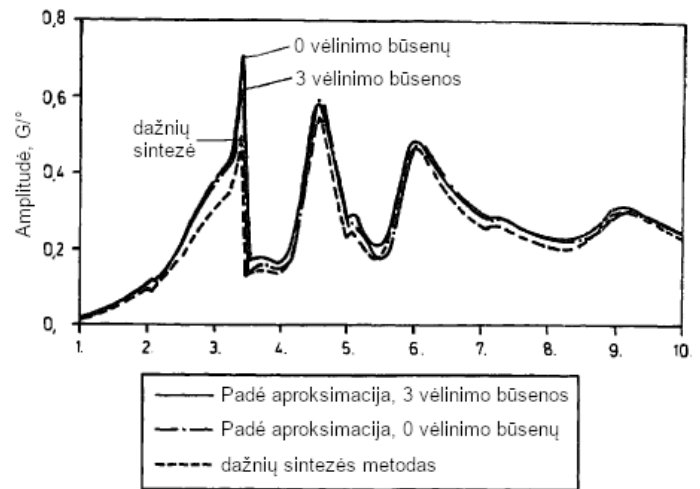
Trečias Zimmermann'o aprašomas būdas yra vadinamieji Padé'o polinomiali, aprašomi formule

$$\frac{1}{2} \rho v^2 Q = \left( \tilde{A}_0 + \tilde{A}_1 \left( \frac{cs}{v} \right) + \tilde{A}_2 \left( \frac{cs}{v} \right)^2 + \sum_{m=1}^L \frac{\tilde{B}_m s}{s + (v/c) \beta m} \right). \quad (1.5)$$

Paskutiniai šios lygties nariai (aprašyti suma) yra vadinami vėlavimo būsenomis. Norint gauti gerą aproksimaciją naudojantis šiuo metodu, reikia naudoti didelį kiekį vėlavimo būsenų. Zimmermann atliko bandymą su lėktuvo sparnų galiukuose pritvirtintomis besisukančiomis aerodinaminėmis plokštumomis, sukeliančiomis sparno virpesius. 9 pav. pateiktas šio bandymo rezultatų palyginimas su dažninės sintezės metodo rezultatais (pagal 1.3 lygtį), o 10 pav. pateiktas dažninės sintezės metodo palyginimas su Padé polinomų analize, kai naudojamos 0 ir 3 vėlavimo būsenos. Galima matyti, kad, ypač prie pirmojo rezonansinio dažnio, Padé analizė smarkiai pervertina svyravimų amplitudę.



**9 pav.** Išmatuotos svyravimo amplitudės palyginimas su apskaičiuotos dažninės sintezės metodu [33]



**10 pav.** Dažninės sintezės metodu apskaičiuotos amplitudės palyginimas su apskaičiuotos Padé'o polinomų metodu [33]

1990 m. straipsnyje Noll'as apžvelgia tuo metu naujus NASA pasiekimus aeroservotamprumo srityje [34]. Autorius aprašo tiek analizės metodus, tiek naudojamų modelių tobulinimą, taip pat aprašo modelių validavimą ir aeroservotamprumo taikymo sritis.

Savo 1999 m. straipsnyje Karpel'as aprašo kelių tuo metu naujų sumažintos eilės aeroservotamprumo modelių apjungimą, siekiant sukurti programą, optimizuojančią visus aeroservotamprumo aspektus kartu [35]. Jo optimizavimo algoritmas atsižvelgė į flaterį, valdymo ribas, atsaką į gūsius, aerotamprumo efektyvumą, apkrovas ir įtempius.

Weishäupl'as ir Laschka savo 2004 m. straipsnyje aprašo Eulerio formulotės srauto analizės pritaikymą aeroservotamprumo uždaviniams skaičiuoti [36]. Tam buvo atmesta pastovi jėgos dedamoji ir nagrinėta tik maža kintanti jėgos dedamoji, dėl kurios pasireiškia aeroservotamprumo efektai. Autoriai parodo, kad tokio metodo tikslumas palyginti su įprastiniu Eulerio metodu yra labai geras, o reikalingas skaičiavimo laikas sumažėja nuo 3 iki 12 kartų.

Munteanu ir kt. 2005 m. išleido straipsnį, kuriame aprašo aerodinaminių ir struktūrinių netiesiškumų įvedimą į sumažintos eilės aeroservotamprumo modelį [37]. Autoriai teigia, kad toks modelis veikia greitai, palyginus su esamais metodais.

Savo 2005 m. straipsnyje Mor'as ir Livne aprašo minimalios būsenos metodo tobulinimą [38]. Įprastai šis metodas leidžia sutaupyti daug skaičiavimo laiko atliekant aeroservotamprumo skaičiavimus, bet tik tais atvejais, kai bendra konfigūracija yra nekintanti. Autoriai pasiūlo patobulinimą, kuris leistų gauti labai greitą veikimą, esant kintančiai lėktuvo konfigūracijai, ir pademonstruoja jo veikimą analizuodami atvejį su stipriai kintančia lėktuvo sparno plano forma.

Moulin'as savo 2006 m. straipsnyje aprašo sudarymą modelio, leidžiančio atlikti atsparią (angl. *robust*) aeroservotamprumo analizę laiko arba dažnių domene [39]. Tokia analizė atsižvelgia į sistemoje

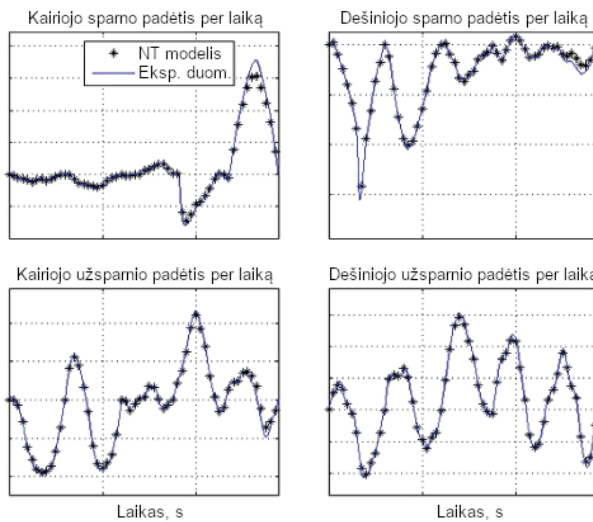
egzistuojančius neapibrėžtumus, ir užtikrina, kad nukrypus nuo idealiai tikslaus atvejo, modelis vis tiek bus validus.

2006 m. straipsnyje Thomas'as ir kt. aprašo būdą įvesti pataisas į sumažintos eilės aeroservotamprumo modelį [40]. Autoriai parodo, kad jų metodas palyginus su standartiniais tuo metu taikytais metodais pasiekia žymiai tikslesnį atsakymą per tą patį iteracijos laiką.

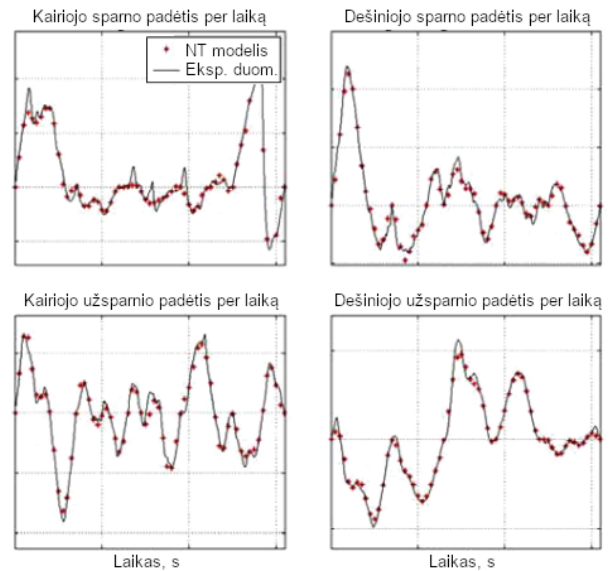
Boėly ir Botez'as savo 2010 m. straipsnyje aprašo bandymą sukurti neuroninį tinklą, skirtą F/A-18 aeroservotamprumui ir flateriui modeliuoti [41]. Tinklo apmokymui buvo naudoti skrydžio flaterio bandymų duomenys prie penkių skirtingų skrydžio aukščio ir Macho skaičiaus kombinacijų. Dar 11 bandymų duomenys buvo naudojami sukurto modelio validavimui. Duomenys buvo surenkami skrydžio metu judinant eleronus įvairiais dažniais (tam naudota speciali įranga) ir matuojant lėktuvo struktūros deformacijas.

Mokomam neuroniniam tinklui kaip įvesties duomenys buvo pateikiami visų valdymo paviršių atsilenkimo padėties, o kaip išvesties buvo reikalaujama abiejų sparnų ir užsparnių padėčių. Tai reiškia kad neuroninio tinklo įvesties sluoksnis turi turėti šešis neuronus, o išvesties keturis neuronus. Autoriai nusprendė naudoti keturių sluoksnių neuroninį tinklą, kuris turi penkis paslėptus neuronus abiejuose paslėptuose sluoksniuose (iš viso 20 neuronų, iš kurių 10 paslėpti).

Validuojant apmokytą neuroninį tinklą, iš 11 tam naudotų atvejų blogiausia atitiktis buvo 89,29%, šis atvejis yra parodytas 11 pav.. Taip pat buvo atlikti modelio atsparumo nuokrypiams bandymai, kai modeliui pateikiami šiek tiek modifikuoti duomenys ir tikrinama, ar jo rezultatai nenukrypsta. Blogiausias iš šių atvejų pavaizduotas 12 pav., jo atitiktis buvo 76,78%.



**11 pav.** Blogiausią atitiktį turintis modelio validavimo atvejis ( $M = 1, 2, H = 10000\text{ft}$ ) [41]



**12 pav.** Blogiausią atitiktį turintis modelio atsparumo testavimo atvejis ( $M = 1, 3, H = 25000\text{ft}$ ) [41]

Nustatyta, kad neuroniniu tinklu paremto modelio efektyvumas ir atsparumas yra labai geras, taip pat modelis pasižymi labai greitu veikimu. Pagrindinis tokio modelio trūkumas yra palyginti sudėtingas taikymas stabilumo analizei.

## 2. Aeroservotamprumo modelio aprašymas

Remiantis atlikta literatūros analize, buvo sudarytas fizikinis lėktuvo skrydžio modelis. Pagal jį buvo parašytas kompiuterinis skrydžio modelis. Šiame skyriuje bus aprašomi sudaryti fizikinis ir kompiuterinis skrydžio modeliai.

### 2.1. Fizikinis modelis

Kuriamas modelis skirtas analizuoti mažo standumo orlaiviams, kurių deformacijos skrydžio metu ženkliai paveikia jų aerodinamines savybes. Dėl to skrydžio modeliui sudaryti naudojami ne tik kinematinis ir aerodinaminis (pakankami aprašant kieto kūno judėjimą erdvėje dėl aerodinaminių jėgų poveikių), bet ir struktūrinis modelis. Tai leidžia aprašyti deformuojamo kūno judėjimą erdvėje veikiant aerodinaminėms jėgoms.

Tiriamas modelis padalinamas į idealiai standžius elementus, tarpusavyje sujungtus kampinėmis spyruoklėmis su slopinimu (išsamiau žr. 2.1.2 poskyryje).

Orlaivį veikiančioms aerodinaminėms jėgoms apskaičiuoti buvo nuspręsta naudoti sukurių tinklelio metodą (angl. *vortex lattice method*), toliau STM. Šis metodas pasirinktas dėl nedidelės reikalingos skaičiuojamosios galios ir gero tikslumo skaičiuojant keliamąją jėgą esant ikikritiniams atakos kampams (išsamiau žr. 2.1.3 poskyryje).

Lėktuvo judėjimo lygtys yra aprašomos laiko atžvilgiu, naudojant tradicines judėjimo erdvėje veikiant išorinėms jėgoms lygtis. Kiekvienas orlaivį sudarantis elementas turi tiesinę inerciją, aprašomą masės, ir sukimosi inerciją, aprašomą inercijos momentų matricos (išsamiau žr. 2.1.1).

#### 2.1.1. Kinematinis modelis

Kiekvieno elemento slenkamasis judėjimas aprašomas lygtimis

$$A = \frac{F}{m}, \quad (2.6)$$

$$V = V_0 + \frac{A + A_0}{2} \cdot dt, \quad (2.7)$$

$$X = X_0 + \frac{V + V_0}{2} \cdot dt, \quad (2.8)$$

čia  $X$  – padėties vektorius,  $V$  – greičio vektorius,  $A$  – pagreičio vektorius,  $F$  – jėgos vektorius,  $m$  – elemento masė,  $dt$  – diskretinis laiko žingsnis. Indeksu 0 žymimi dydžiai, apskaičiuoti praeito laiko žingsnio metu. Kiekviename laiko žingsnyje diskretinei integracijai naudojami vidutinės per praeitą laiko žingsnį buvusios greičio ir pagreičio vertės.

Sukamasis judėjimas ir kampinė padėtis analogiškai aprašoma

$$\varepsilon = \frac{M}{I}, \quad (2.9)$$

$$\omega = \omega_0 + \varepsilon \cdot dt, \quad (2.10)$$

$$\theta = \theta_0 + \omega \cdot dt, \quad (2.11)$$



čia  $\theta$  – kampinės padėties vektorius,  $\omega$  – kampinio greičio vektorius,  $\varepsilon$  – kampinio pagreičio vektorius,  $M$  – momento vektorius,  $I$  – elemento inercijos matrica.

Kad būtų kompensuotos diskretiškumo paklaidos ir išlaikytas atstumas tarp elementų (elementai sujungti visiškai standžiai slenkamajam judėjimui), kiekvieną laiko žingsnį atliekama normalizacija, kurios metu elementų padėtis perskaičiuojama pagal jų dydį ir kampinę padėtį centrinio elemento atžvilgiu.

### 2.1.2. Struktūrinis modelis

Daroma prielaida, kad lėktuvą sudarantys elementai yra idealiai standūs. Elementai tarpusavyje sujungti kampinėmis spyruoklėmis su slopintuvais, kurių standumas aprašytas standumo matrica (standumo matavimo vienetas Nm/rad), o slopinimas aprašytas slopinimo matrica (slopinimo matavimo vienetas Nm/(rad/s)). Jungtys tarp elementų yra idealiai standžios poslinkiui. Toks modelis tinka vertinti sukimo ir lenkimo deformacijoms, kurios vyrauja orlaiviuose, bet neatsižvelgia į tempimo ir šlyties deformacijas.

Standumo momentas, veikiantis bet kurį elementą, aprašomas formule

$$M_{k,i} = - \sum_{j=1}^n (K_{ij} \times \Delta\theta_{ij}) , \quad (2.12)$$

čia  $M_{k,i}$  –  $i$ -tąjį elementą veikiantis standumo momentas;  $n$  – elementų, su kuriais sujungtas  $i$ -tasis elementas, skaičius;  $K_{ij}$  – jungties tarp  $i$ -tojo ir  $j$ -tojo elementų standumo matrica;  $\Delta\theta_{ij}$  – kampinės padėties tarp  $i$ -tojo ir  $j$ -tojo elementų skirtumas.

Slopinimo momentas aprašomas panašia formule

$$M_{c,i} = - \sum_{j=1}^n (C_{ij} \times \Delta\omega_{ij}) , \quad (2.13)$$

čia  $M_{c,i}$  –  $i$ -tąjį elementą veikiantis slopinimo momentas;  $C_{ij}$  – jungties tarp  $i$ -tojo ir  $j$ -tojo elementų slopinimo matrica;  $\Delta\omega_{ij}$  – kampinio greičio tarp  $i$ -tojo ir  $j$ -tojo elementų skirtumas.

Sudarant konkretaus lėktuvo skrydžio modelį, standumo ir slopinimo matricių vertės gali būti nustatomos eksperimentiniu būdu (jei sudaromas realaus lėktuvo modelis) arba apskaičiuojamos atliekant tikslesnę struktūrinę lėktuvo analizę.

### 2.1.3. Aerodinaminis modelis

Lėktuvą veikiančioms aerodinaminėms jėgoms apskaičiuoti naudojamas sūkurių tinklelio metodas (toliau STM). Šis metodas geru tikslumu apskaičiuoja keliamąją jėgą, veikiančią aerodinaminis paviršius esant ikikritiniams atakos kampams, ir pasižymi sąlyginai greitu veikimu. Kitas STM pranašumas yra tai, kad jis efektyviai skaičiuoja atlenktų valdymo paviršių sukuriamas jėgas.

Įprastas STM neatsižvelgia į oro klampumą, juo apskaičiuojama pasipriešinimo jėga nėra tiksli (įvertinamas tik indukuotas, bet ne profiline trinties pasipriešinimas). Kad būtų gaunama tikslesnė pasipriešinimo jėga, prie sparną veikiančių jėgų pridedamas profiline pasipriešinimas, interpoliuojamas pagal vietinį keliamosios jėgos koeficientą iš profilio aerodinaminių duomenų.

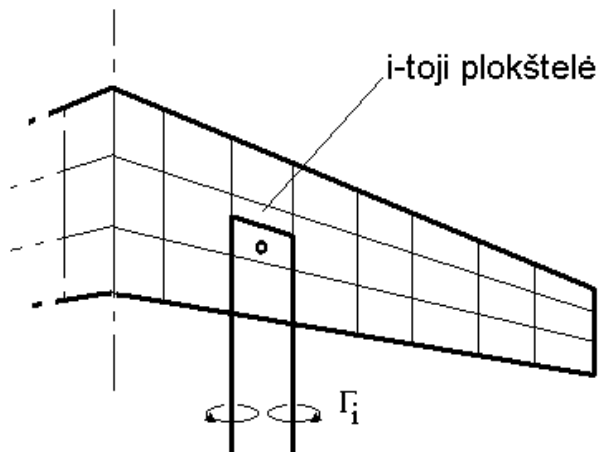
STM nėra tinkamas skaičiuoti aerodinamines jėgas, veikiančias didelio storio objektus, pavyzdžiui lėktuvo liemenį. Daroma prielaida, kad didžioji dalis lėktuvą veikiančios aerodinaminės jėgos bus sukuriama sparno ir uodegos paviršių, dėl to į liemenį veikiančias jėgas galimas neatsižvelgti.

Kitas STM trūkumas yra tai, kad šis aerodinaminis modelis yra kvazi-statinis, t.y. visada daroma prielaida, kad srautas yra nusistovėjęs. Kadangi iš tiesų skaičiuojamas srautas kintantis, tai sukelia tam tikrų skaičiavimo netikslumų. Taip pat STM nėra tinkamas skaičiuoti aerodinaminėms jėgoms kai srautas atitrūkęs, todėl kuriamas modelis bus tinkamas tik skaičiuojant skrydį esant ikikritiniams atakos kampams.

STM skaičiavimas paremtas aerodinaminio paviršiaus padalinimu į plokšteles. Šiuo konkrečiu pritaikymo atveju, į plokšteles padalinami lėktuvo modelio elementai, atitinkantys aerodinaminis paviršius. Valdymo paviršiams skiriama tam tikra dalis aerodinaminių paviršių, šių plokštelių padėtis keičiasi priklausomai nuo valdymo paviršiaus atlenkimo.

Taikant superpozicijos principą yra daroma prielaida, kad aplink sparną susidarantis sūkurys gali būti padalintas į atskirus sūkurius (vienas sūkurys kiekvienai plokštelei), nepakeičiant sūkurio poveikio. plokštelės sūkurio pavyzdys pateiktas 13 pav.. Metodas taip pat paremtas tuo, kad oro srauto greitis statmenai plokštelės visada yra lygus nuliui [42].

Kiekvienos plokštelės sūkurys sudarytas iš trijų dalių – prisijungusio sūkurio, sutampančio su plokštelės ketvirčio stygos linija, ir dviejų numestų sūkurių, einančių lygiagrečiai laisvam srautui į begalybę. Taip pat kiekvienoje plokštelėje yra parenkamas integracijos taškas (įprastai trijų ketvirčių stygos linijos centre), kuriame sumuojamas kitų plokštelių sūkurių poveikis.



13 pav. STM vienos plokštelės sūkurio pavyzdys [42]

Pagrindinė STM skaičiavimo metu sprendžiama lygtis yra [42]

$$w_i = \sum_{j=1}^N A_{ij} \Gamma_j, \quad (2.14)$$

čia  $w_i$  – sparno indukuoto srauto greičio ties  $i$ -tąja plokštele komponentė, statmena tai pačiai plokštelei;  $N$  – plokštelių skaičius;  $A_{ij}$  – koeficientas nusakantis  $w_i$  priklausomybę nuo  $\Gamma_j$ ;  $\Gamma_j$  –  $j$ -tosios plokštelės sūkurio stiprumas. Ši lygtis nusako tai, kad indukuotas srauto greitis ties bet kuria plokštele yra lygus visų plokštelių sūkurių indukuotų tame taške greičių sumai.

Kadangi srautas statmenai plokštelių neteka,  $w_i$  yra lygus laisvojo srauto komponentei, statmenai  $i$ -tajai plokštei, tik priešingos krypties, t.y. sparno indukuojamas srautas kiekviename taške yra toks, kad panaikinama laisvojo srauto komponentė, statmena sparnui tame taške. Jei  $\alpha_i$  yra  $i$ -tosios plokštelės atakos kampas, o  $V$  – laisvojo srauto greitis, šis dėsnis aprašomas lygtimi [42]

$$w_i = -V \sin(\alpha_i). \quad (2.15)$$

$A_{ij}$  apskaičiuojamas pagal Bio-Savaro dėsnį. Šis dėsnis teigia, kad  $\Gamma$  stiprumo sūkurys, taške, esančiame  $r$  atstumu nuo sūkurio linijos indukuos srauto greitį [43]

$$w = \frac{\Gamma}{4\pi r} (\cos \alpha_1 - \cos \alpha_2), \quad (2.16)$$

kai  $\alpha_1$  – kampas tarp sūkurio linijos ir linijos, jungiančios tašką su sūkurio pradžia, ir  $\alpha_2$  – kampas tarp sūkurio linijos ir linijos, jungiančios tašką su sūkurio pabaiga. Pagal 2.16 lygtį,  $A_{ij}$  apskaičiuojamas

$$A_{ij} = \frac{-1 - \cos \alpha_1}{4\pi r_{ns1}} + \frac{\cos \alpha_1 - \cos \alpha_2}{4\pi r_{ps}} + \frac{\cos \alpha_2 + 1}{4\pi r_{ns2}}, \quad (2.17)$$

šioje lygtyje pirmas narys aprašo pirmojo numesto sūkurio įtaką, antras sūkurys prisijungusio sūkurio įtaką, o trečias narys antrojo numesto sūkurio įtaką. Atstumai  $r$  matuojami nuo  $i$ -tosios plokštelės integracinio taško iki atitinkamo  $j$ -tosios plokštelės sūkurio linijos.

Turint koeficientų matricą  $A$  ir kairės lygties pusės vektorių  $w$ , sprendžiama tiesinė lygčių sistema ir apskaičiuojamas nežinomųjų vektorius  $\Gamma$ , nurodantis kiekvienos plokštelės sūkurio stiprumą. Tada  $i$ -tają plokštelę veikianti keliamoji jėga apskaičiuojama pagal formulę

$$L_i = 2\rho V \Gamma_i k_i, \quad (2.18)$$

čia  $\rho$  – oro tankis;  $k_i$  – plokštelės plotis. Bendra elementą veikianti keliamoji jėga apskaičiuojama susumavus visų jį sudarančių plokštelių jėgas.

## 2.2. Standaus lėktuvo modelis

Palyginimui su aeroservotamprumo modeliu taip pat atliekamas skrydžio modeliavimas, neatsižvelgiant į struktūrinės deformacijas. Naudojami aerodinaminis ir kinematinis modeliai tokie patys, kaip ir aeroservotamprumo modelyje, bet naudojamas visiškai standus struktūrinis modelis. Tam pagal atskirų elementų parametrus apskaičiuojama viso lėktuvo masė ir inercijos momentai ir priskiriami pirmajam lėktuvo modelio elementui. Vykdam skaičiavimus, visų jėgų ir momentų poveikis perduodamas pirmajam elementui ir tik jam integruojamas judesys pagal kinematinės lygtis. Visi kiti elementai išlaiko tą pačią kampinę padėtį kaip ir pirmasis. Visų elementų linijinė padėtis apskaičiuojama pagal kampines padėtis, išlaikant fiksuotą atstumą nuo pirmojo elemento.

## 2.3. Modelio programinis įgyvendinimas

Šiame poskyryje bus apžvelgtas programinis sukurto fizikinio modelio įgyvendinimas, jo panaudojimas skrydžio skaičiavimams.

### 2.3.1. Kompiuterinio modelio aprašas

Fizikinis modelis buvo įgyvendintas programiškai naudojant C programavimo kalbą. Ši kalba pasirinkta dėl greito veikimo. Lėktuvo struktūrą sudarantys elementai ir jungtys aprašomi tekstiniam failui JSON formatu, šiuos duomenis sukurta programa nuskaityti ir pagal juos sudaro lėktuvo modelį. Tekstiniuose įvesties failuose pateikiami jungtys ir elementus aprašantys kintamieji išvardinti atitinkamai 2 ir 3 lentelėse.

**2 lentelė.** Lėktuvo modelio jungtį aprašantys kintamieji

Kintamojo pavadinimas	Kintamojo tipas	Vektoriaus arba matricos dydis	Matavimo vienetas
Pradinė padėtis erdvėje	Realiųjų skaičių vektorius	3	mm
Standumo koef. matrica	Realiųjų skaičių matrica	3×3	Nmm/rad
Slopinimo koef. matrica	Realiųjų skaičių matrica	3×3	Nmm/(rad/ms)
Jungiamų elementų numeriai	Sveikųjų skaičių vektorius	2	–
Profilio stygos ilgis	Realusis skaičius	–	mm
Priekinės briaunos santykinė padėtis	Realiųjų skaičių vektorius	3	mm

Jungties tarp elementų pradinė padėtis erdvėje aprašoma trijų realiųjų skaičių vektoriumi, nurodančiu jungties padėtį globalioje koordinačių sistemoje. Standumo ir slopinimo koeficientų matricos aprašo jungties kampinės spyruoklės savybes. Jungiamų elementų numerių vektoriuje įvedami elementų, kuriuos jungia aprašoma jungtis, eilės numeriai. Viena jungtis gali jungti tik du elementus, ir visada pirmas elementas turi būti įvedamas tas, kuris elementų medyje yra arčiau pirmojo elemento. Jei jungtis naudojama tik aerodinaminiam paviršiui aprašyti ir nejungia dviejų elementų, elementų vektoriuje du kartus įvedamas elemento, kuriam priklauso jungtis, eilės numeris. Profilio stygos ilgis nurodo sparno stygos ilgį ties aprašoma jungtimi, o priekinės briaunos santykinė padėtis nurodo aerodinaminio profilio priekinės briaunos padėtį jungties koordinačių sistemoje.

**3 lentelė.** Lėktuvo modelio elementą aprašantys kintamieji

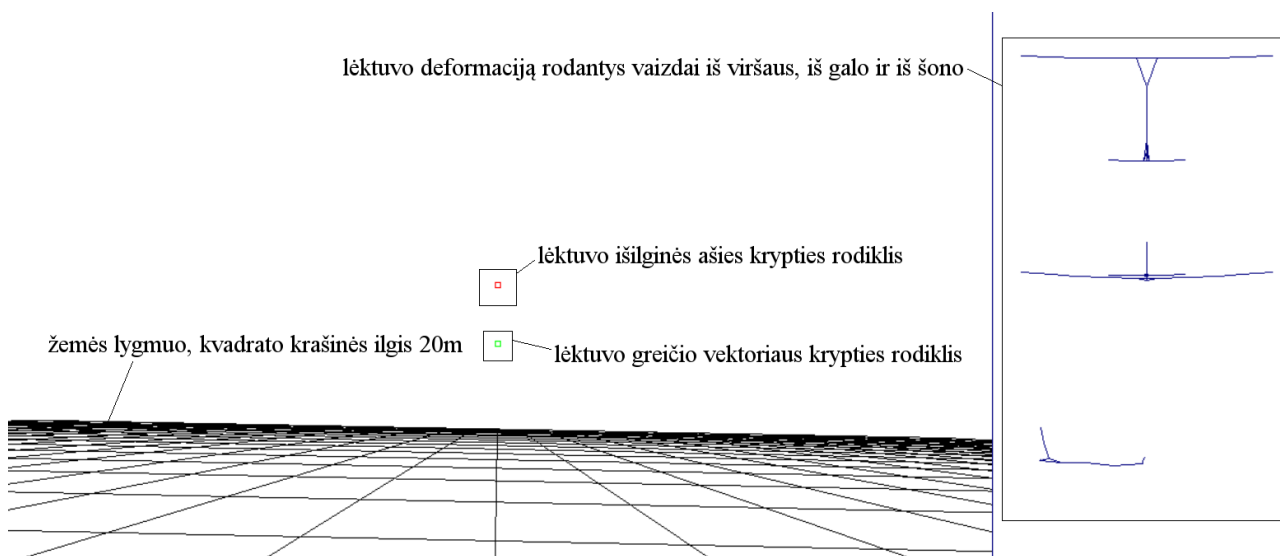
Kintamojo pavadinimas	Kintamojo tipas	Vektoriaus arba matricos dydis	Matavimo vienetas
Masė	Realusis skaičius	–	g
Pradinė padėtis erdvėje	Realiųjų skaičių vektorius	3	mm
Sukimo inercijos matrica	Realiųjų skaičių matrica	3×3	g·mm <sup>2</sup>
Aerod. plokštelės išilgai mosto	Sveikasis skaičius	–	–
Aerod. plokštelės išilgai stygos	Sveikasis skaičius	–	–
Vald. pav. plokštelės išilgai stygos	Sveikasis skaičius	–	–
Santykinė vald. pav. styga	Realusis skaičius	–	–

Visi modelį sudarantys elementai aprašomi mase, masės centro pradine padėtimi globalioje koordinačių sistemoje ir sukimo inercijos matrica. Elementai, kurie sudaro aerodinaminį paviršius, taip pat aprašomi STM plokštelių skaičiumi, į kurį padalintas elementas išilgai mosto ir išilgai stygos. Jei šie kintamieji lygūs nuliui, elementas praleidžiamas sudarinėjant STM skaičiuojamąjį tinklą. Jei aerodinaminis paviršius turi ir valdymo paviršių, papildomai nurodomas jį sudarančių panelių skaičius išilgai stygos ir santykinis stygos ilgis.

Bendru atveju kiekvienas elementas gali turėti neribotą kiekį jungčių, bet aerodinaminis paviršius aprašantys elementai gali turėti tik dvi jungtis. STM skaičiuojamasis tinklelis sudaromas kiekvienam elementui atskirai. Abiejose elemento turimose jungtyse sudaromos stygos linijos. Linija pradama nuo jungties duomenyse nurodyto priekinės briaunos taško, o jos kryptis nustatoma pagal kampinę abiejų elementų, kuriuos jungia jungtis, padėtį. Jei elemento aerodinaminis paviršius atlenktas, atitinkama stygos linijos dalis nulenkiama reikiamu kampu. Po to stygos linijos padalinamos į trumpesnes atkarpas, kurių skaičius atitinka įvestyje nurodytą aerodinaminių plokštelių išilgai stygos skaičių. Abiems stygos linijoms priklausančių atkarpų galai tarpusavyje sujungiami išilgai mosto einančiomis linijomis. Šios išilgai mosto einančios linijos taip pat padalinamos į atkarpas pagal įvestyje nurodytą plokštelių skaičių, jungiant šių atkarpų galus užbaigiamas elemento STM skaičiuojamasis tinklelis.

Toliau naudojant įgyvendintą fizikinį modelį atliekami skaičiavimai. STM pagrindinė lygtis ir elementų pagreičio skaičiavimo lygtis yra didelės tiesinių lygčių sistemos, joms spręsti naudojama *LAPACK* biblioteka [46]. Skaičiuojant taip pat reikia atlikti daug paprastesnių veiksmų su matricomis ir vektoriais, tam naudojama *BLAS* biblioteka [45].

Kad modeliui veikiant būtų galima geriau suprasti skrydžio pobūdį ir modeliuojamo lėktuvo judėjimą erdvėje, programos sukuriame lange (parodytame 14 pav.) gyvai atvaizduojama lėktuvo padėtis ir struktūros deformacijos. Šiam atvaizdavimui naudojama *SDL2* grafinė biblioteka [44].



14 pav. Sukurtos programos lango vaizdas

Išanalizavus programos veikimą buvo nustatyta, kad didžiąją dalį skaičiavimo laiko užima STM pagrindinės lygties koeficientų matricos sudarymas, dėl to ši programos dalis buvo paralelizuota naudojant standartinę C programavimo kalbos *pthread* biblioteką.

Dėl didelės apimties visas programos kodas darbe nepateikiamas, 1 priedas pateiktos funkcijos aerodinaminėms jėgoms skaičiuoti naudojant STM.

### 3. Modelio validavimo metodika

Šiame skyriuje aprašyta aeroservotamprumo modelio eksperimentinio validavimo metodika. Validavimo procesą galima išskirstyti į šiuos etapus:

1. skrydžio bandymų atlikimas,
2. skrydžio modeliavimas, naudojant 2 skyriuje aprašytą aeroservotamprumo modelį,
3. skrydžio modeliavimas, naudojant 2.2 poskyryje aprašytą standaus lėktuvo modelį,
4. abiejų modelių rezultatų palyginimas tarpusavyje ir su skrydžio bandymų rezultatais, nuokrypių apskaičiavimas.

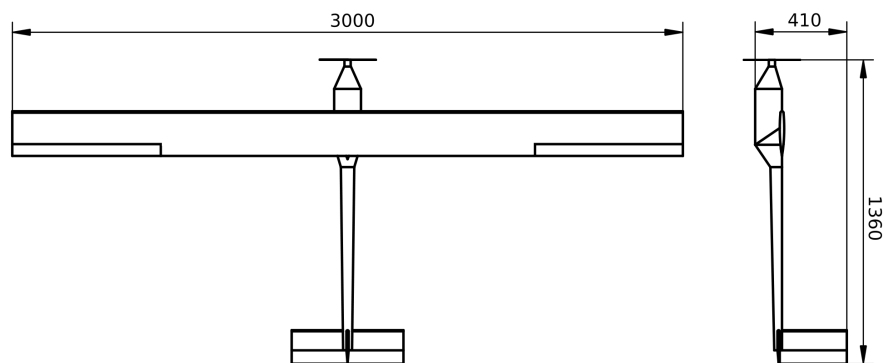
Toliau kiekvienas iš etapų nagrinėjamas išsamiau.

#### 3.1. Skrydžio bandymų metodika

Skrydžio bandymai atliekami su specialiai tam sukonstruotu nuotoliniu būdu valdomu lėktuvu (plačiau aprašytas 3.1.1 skyrelyje). Skrydžio metu lėktuvas filmuojamas, kad būtų užfiksuotas bendras jo skrydžio pobūdis, taip pat lėktuve įmontuotas mikrovaldiklis renka duomenis apie skrydį. Toliau bus išsamiau apžvelgtos skrydžio bandymų dedamosios.

##### 3.1.1. Bandymų orlaivis

Bandymams naudojamo orlaivio du vaizdai ir gabaritiniai matmenys pateikti 15 pav.. Lėktuvas yra klasikinės konfigūracijos. Pagrindinė konstrukcinė medžiaga polistireno putos, dengta lipnia juosta dėl standumo ir geresnės paviršiaus kokybės. Lėktuvo sparnas papildomai sustiprintas mediniu lonžeronu. Po sparnu esančioje dėžutėje sumontuoti lėktuvo elektronikos komponentai.



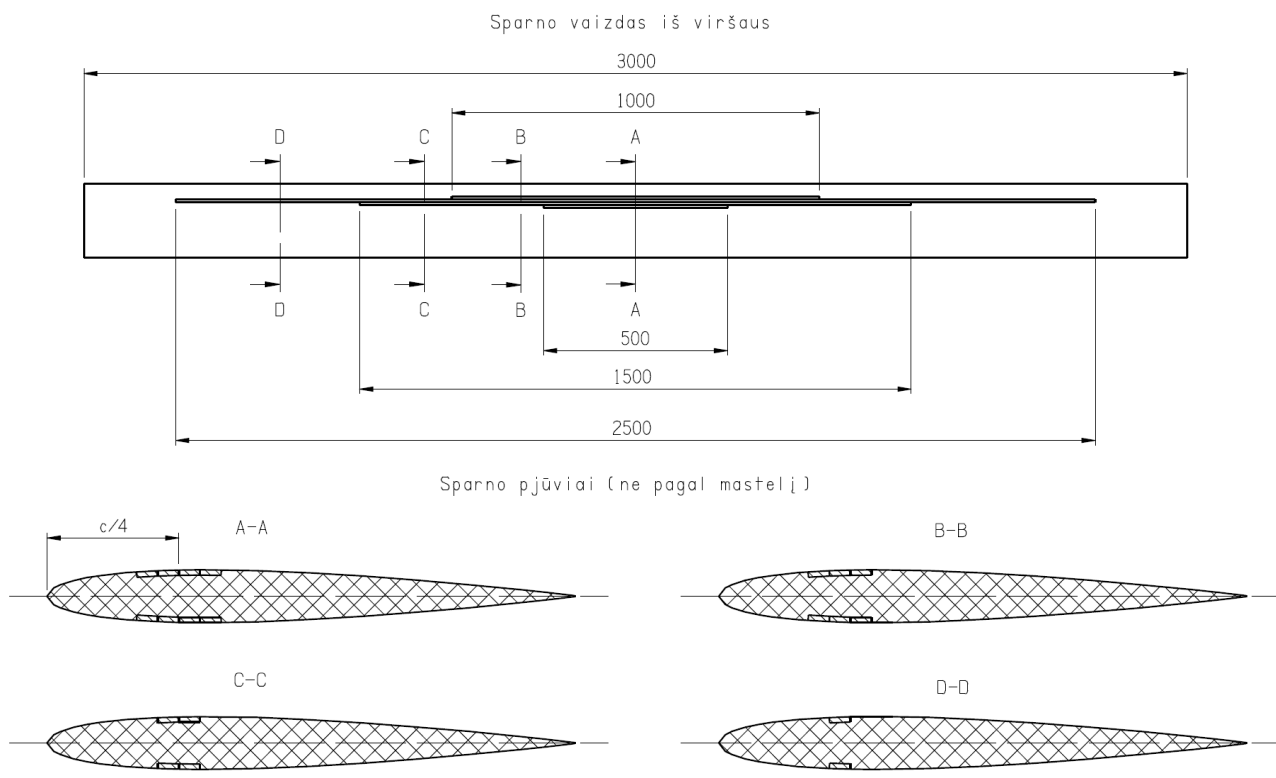
15 pav. Bandymams naudojamo lėktuvo gabaritiniai matmenys

Lėktuvo dalių masės aprašytos 4 lentelėje. bendra lėktuvo masė 1,660 kg, o masės centro padėtis sparno priekinės briaunos atžvilgiu 0,153 m. Lėktuvo centruotė yra 76,5 %. Nors tokia centruotė labiau galinė nei įprasta, lėktuvas yra statiškai ir dinamiškai stabilus.

**4 lentelė.** Lėktuvo atskirų dalių masės ir jų padėtys (atstumai pateikti nuo sparno priekinės briaunos)

Komponentas	Masė, kg	Išilginė masės centro padėtis, m
Liemuo	1,030	0,115
Sparnas	0,550	0,090
Stabilizatorius	0,050	1,075
Kilis	0,030	1,075
<b>Visas lėktuvas</b>	<b>1,660</b>	<b>0,153</b>

Lėktuvo sparno mostas 3,0 m, styga 0,2 m, plotas 0,6 m<sup>2</sup>. Sparnui naudotas simetrinis NACA 0010 aerodinaminis profilis. Lėktuvo sparno viršuje ir apačioje įklijuotos lonžerono lentynos, kurių išdėstymo schema pateikta 16 pav.. Lonžeronas pagamintas iš pušinių 2×8 mm tašelių.



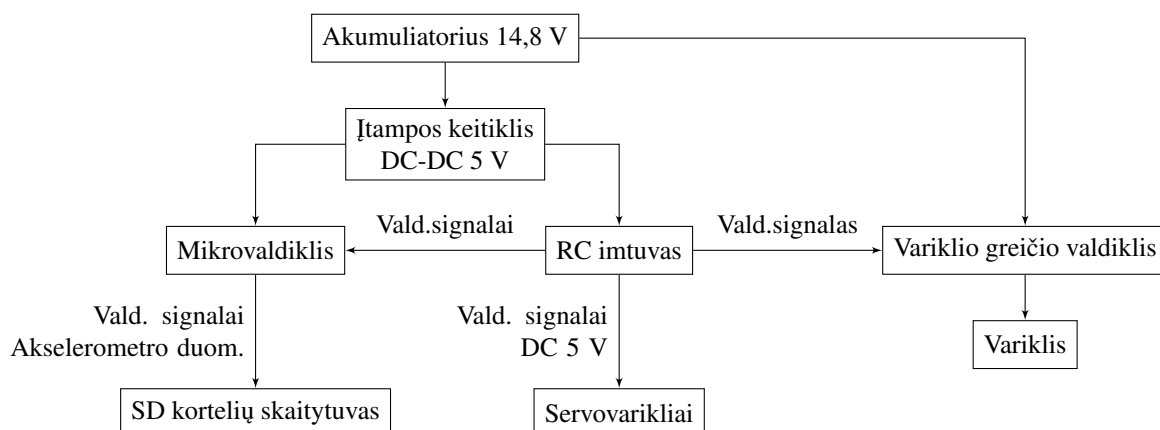
**16 pav.** Bandymams naudoto lėktuvo sparno lonžerono schema

Viršutinę ir apatinę lonžerono lentynas sudaro po 4 tašelius, iš kurių trumpiausias yra 0,5 m ilgio, o ilgiausias 2,5 m ilgio. Lonžerono lentynų centras yra ties ketvirčio stygos linija (0,05 m nuo sparno priekinės briaunos). Sparnų galuose yra 0,75 m ilgio eleronai, jų stygos ilgis sudaro 25 % sparno stygos ilgio.

Lėktuvo uodegos paviršiams naudotas plokščias 10 mm storio (6,7 % santykinio storio) profilis. Abiejų paviršių stygos ilgis 0,15 m. Stabilizatoriaus mostas 0,5 m, plotas 0,075 m<sup>2</sup>. Kilio aukštis 0,3 m, plotas 0,045 m<sup>2</sup>. Aukštumos ir posūkio vairai įrengti per visą uodegos paviršių ilgį ir sudaro 40 % uodegos paviršių stygos.

17 pav. pateikta lėktuvo elektrinės sistemos schema. Orlaivio traukos sistemą sudaro bešepetėlinis variklis „PROPDRIVE v2 2830 800KV“, variklio greičio valdiklis „ElectriFly Silver Series 35A“ ir 4 celių ličio polimerų akumuliatorius „Turnigy Graphene 4000mAh 4S 65C“. Nuotolinis valdymas

atliekamas naudojant siųstuvą „RadioLink AT9s Pro“, lėktuve valdymo signalams priimti naudotas imtuvas „RadioLink R9DS“. Valdymo plokštumoms valdyti naudojami servovarikliai „Turnigy TGY-9025MG“.



17 pav. Bandymams naudoto lėktuvo elektrinės sistemos blokinė diagrama

Į servovariklius imtuvas siunčia valdymo signalą, kurį sudaro 1,09–1,90 ms ilgio pulsas (pulso ilgiai nustatyti matuojant mikrovaldikliu), pasikartojantis kas 25 ms. Reikiama servovariklio padėtis nustatoma pagal siunčiamo pulso ilgį. Fiksuojant šiuos pulsų ilgius, galima nustatyti valdymo paviršių padėties kitimą einant laikui.

Pulsų ilgius fiksuoja mikrovaldiklis „Arduino Uno WiFi Rev.2“, taip pat naudojamas fiksuoti ir akcelerometro duomenis. Šis mikrovaldiklio modelis pasirinktas dėl integruoto akcelerometro-giroskopo ir turimo pakankamo kiekio įvesties kanalų, tinkamų pulso ilgiams matuoti. Prie mikrovaldiklio prijungtas SD kortelių skaitytuvas, saugantis akcelerometro, giroskopo ir valdymo signalų duomenis SD kortelėje. Matavimo sistemos išgaunamas diskretizacijos dažnis apytiksliai lygus 12 Hz.

### 3.1.2. Bandymo sąlygos

Skrydžio bandymai atliekami lauke, nes naudojamas lėktuvas yra per didelis ir per greitas, kad būtų galima skrydžius atlikti uždaroje patalpose. Kad būtų mažesnė vėjo įtaka, bandymai turėtų būti atliekami esant mažam vėjo greičiui, nors esant mažam vėjo greičiui sudėtingesnis orlaivio paleidimas (jis leidžiamas iš rankos). Bandymų atlikti prie mažo vėjo sąlygų nepavyko (atliekant bandymus vėjo greitis buvo apie 6 m/s), todėl atliekant skrydžio modeliavimą privaloma atsižvelgti į vėjo įtaką lėktuvo skrydžiui. Į gūsių poveikį skrydžiui atsižvelgti neįmanoma, nes nėra turima pakankamai duomenų, bet analizuojami tik kelių sekundžių ilgio skrydžio segmentai, todėl vėjo gūsių sukeltos paklaidos yra ribotos. Bandymai atlikti esant sausam orui, saugant bandymų orlaivio elektroniką ir siekiant išvengti įtakos, sukeltos vandens lašų ant aerodinaminių paviršių. Skrydžio bandymai atliekami mažame aukštyje, todėl į oro tankio pokyčius galima neatsižvelgti.

### 3.1.3. Duomenų rinkimas

Skrydžio bandymo metu duomenys fiksuojami dviem skirtingais būdais. Mikrovaldiklis, esantis lėktuve, fiksuoja lėktuvo patiriamus pagreičius ir kampinius greičius, taip pat valdymo signalus, iš imtuvo perduodamus servovarikliams. Papildomi duomenys renkami filmuojant skrydį dviem skaitmeniniais fotoaparatais, taip gaunant erdvinį vaizdą, leidžiantį nustatyti bendrą lėktuvo padėtį erdvėje.



5 lentelėje apibendrinami renkami duomenys, jiems rinkti naudojami prietaisai ir pagrindiniai jų parametrai.

**5 lentelė.** Skrydžių bandymo metu renkami duomenys ir rinkimui naudojami prietaisai

Duomenų tipas	Prietaisas	Diskretizacijos dažnis	Matavimo ribos	Padalos dydis	Paklaida
Lėktuvo padėtis erdvėje	Skaitmeniniai fotoaparatai „Nikon D7100“ ir „Sony α6000“ (pastatyti 0,77 m atstumu)	30 Hz	–	–	–
Pagreitis	Akselerometras-giroskopas LSM6DS3TR (integruotas mikrovaldiklyje) [47]	12 Hz	±16 G	0,488 mG	±40 mG
Kampinis greitis	Akselerometras-giroskopas LSM6DS3TR (integruotas mikrovaldiklyje) [47]	12 Hz	±2000 °/s	0,07 °/s	±10 °/s
Valdymo signalai servovarikliams	Mikrovaldiklis „Arduino Uno WiFi Rev.2“ [48]	12 Hz	0–25 ms	1 μs	–

Kadangi skrydis filmuojamas dviem skirtingais fotoaparatais (prie dviejų vienodų fotoaparatus nebuvo prieigos), turinčiais skirtingus objektyvus, abiejų vaizdų iškreipimai ir matymo kampai bus šiek tiek skirtingi, dėl to atstumo nustatymo tikslumas nebus didelis. Matymo kampų skirtumą galima kompensuoti vieną vaizdą padidinant, bet skirtingo vaizdų iškreipimo kompensuoti neįmanoma, neatlikus išsamaus objektyvų optinių savybių tyrimo, neįveikiant šio darbo apimtį.

Lėktuvo trijose ašyse patiriamas pagreitis ir kampinis greitis fiksuojami mikrovaldiklyje integruoto akselerometro-giroskopo. Mikrovaldiklis taip pat fiksuoja servovariklių valdymo signalų pulso ilgus. Nuo mikrovaldiklio veikimo pradžios praėjęs laikas, akselerometro-giroskopo ir valdymo duomenys yra įrašomi į SD kortelėje esantį tekstinį failą, kurį vėliau galima atidaryti kompiuteryje ir atlikti surinktų duomenų analizę. Trys pavyzdinės failo, kuriame saugomi skrydžio duomenys, eilutės pateiktos 18 pav..

```
t | ax | ay | az | gx | gy | gz | lail | rail | elev | rudd | thrt
1361 | 0.11 | -0.03 | 1.01 | -0.55 | 0.49 | -2.26 | 1363 | 1598 | 1648 | 1481 | 1079 |
1452 | 0.11 | -0.03 | 0.99 | -1.28 | 0.61 | -2.32 | 1378 | 1621 | 1655 | 1496 | 1079 |
```

**18 pav.** Skrydžio duomenų failo trys pirmos eilutės

Pirmojoje eilutėje pateikti trumpiniai nurodo kokius duomenis įrašyti kiekviename stulpelyje:

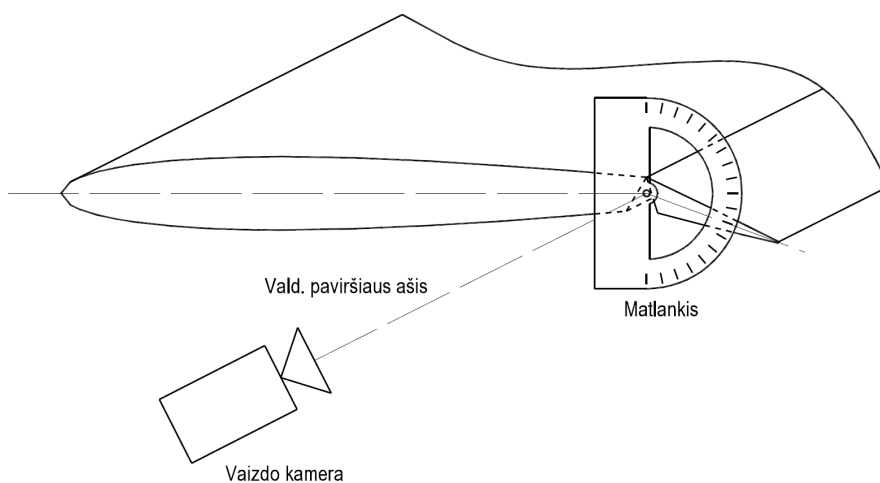
- t – laikas nuo sistemos paleidimo, ms;
- ax, ay ir az – pagreitis trijose ašyse, G;
- gx, gy ir gz – kampinis greitis aplink tris ašis, °/s;
- lail, rail, elev, rudd ir thrt – atitinkamai kairiojo elerono, dešiniojo elerono, aukštumos vairo, posūkio vairo ir greičio valdiklio valdymo signalų pulso ilgiai, μs.

Toliau kiekvienoje failo eilutėje pateikiama po vieną įrašomų duomenų rinkinį.

### 3.2. Valdymo paviršių kalibravimo metodika

Skrydžio metu mikrovaldiklis fiksuoja tik valdymo paviršių servo variklių gaunamus valdymo signalus (atlenkimo kampas valdomas pagal servo variklio gaunamo signalo impulso trukmę), o kompiuteriniam modeliui valdyti reikalingi paviršių atlenkimo kampai. Kad būtų galima apskaičiuoti paviršių atlenkimo kampus pagal užfiksuotas impulsų trukmes, atliekamas valdymo paviršių kalibravimas.

Kalibravimo metu prie aerodinaminio paviršiaus galo pritvirtinamas matlankis, kurio  $90^\circ$  žymė sutampa su neutralia valdymo paviršiaus padėtimi. Skaitmeninis fotoaparatas pastatomas ant trikojo taip, kad jo optinė ašis sutaptų su valdymo paviršiaus sukimosi ašimi, kadre būtų matomas matlankis ir valdymo paviršius. Matavimo schema pateikta 19 pav.. Signalo impulsų trukmės fiksuojamos lėktuve esančiu mikrovaldikliu, kaip aprašyta 3.1.3 skyrelyje.



**19 pav.** Aerodinaminių paviršių atlenkimo priklausomybės nuo valdymo signalo trukmės priklausomybės kalibravimo schema

Pradedant matavimą lėktuvo maitinimas įjungiamas, pradedamas filmavimas skaitmeniniu fotoaparatu. Naudojant valdymo pultą, valdymo paviršius pastatomas į nulinę padėtį, tada  $10^\circ$  žingsniais didinamas atlenkimo kampas, kol pasiekama maksimali atlenkimo riba. Ties kiekvienu žingsniu sujodinamas kuris nors kitas valdymo paviršius (kuris tuo metu nematuojamas), kad būtų lengviau sinchronizuoti mikrovaldiklio įrašytus duomenis su nufilmuota medžiaga. Pasiekus maksimalią paviršiaus atlenkimo ribą, grįžtama į pradinę padėtį ir ta pati procedūra kartojama į kitą paviršiaus atlenkimo pusę. Taip surenkami duomenys iš visų valdymo paviršių.

Paviršių valdymo duomenis ir filmuotą medžiagą sulyginant kompiuteryje, valdymo signalo kitimo grafike rankiniu būdu išskiriami laiko tarpai, kuriuose žinomas paviršiaus atlenkimo kampas. Apskaičiuojant signalo trukmės vidurkį kiekviename iš šių laiko tarpų, gaunama signalo trukmė, reikalinga pasiekti kiekvienam atlenkimo kampui. Šie duomenys vėliau naudojami modeliavimo metu išgauti paviršių atlenkimo kampus pagal skrydžio metu surinktus valdymo duomenis.

### 3.3. Skrydžio modeliavimo metodika

Sudaromas skrydžio bandymų lėktuvo modelis. Modeliui sudaryti panaudojami realaus lėktuvo inercijos ir standumo duomenys, taip pat naudoto aerodinaminio profilio poliarės duomenys.

Sudarant modelį pirmiausia nusprendžiama, kaip jis bus padalintas į elementus. Dėl to, kaip sukurtame aeroservotamprumo modelyje aprašomi aerodinaminiai paviršiai, viename elemente negali būti daugiau nei vienas aerodinaminis paviršius. Aerodinaminių paviršių dalys, kuriose yra valdymo paviršiai, taip pat aprašomos kaip atskiri elementai. Norint pasiekti gerą veikimo tikslumą, į atskirus elementus reikia išskirti lėktuvo dalis, kurių lenkimo ir/arba sukimo standumas yra mažas.

Elementų masės nustatomos atskirai pasvėrus kiekvieną lėktuvo dalį. Inercijos momentai apskaičiuojami pagal mases ir lėktuvo matmenis, darant prielaidą, kad lėktuvas sudarytas iš paprastų geometrinių formų. Daroma prielaida, kad aerodinaminės plokštumos yra plokštumos formos, o liemuo yra stačiakampio gretasienio formos.

Jungčių tarp lėktuvo modelio elementų standumas nustatomas apkraunant elementus lenkimo jėga ir matuojant elemento galo poslinkį. Pagal poslinkį apskaičiuojamas elemento pasisukimo kampas, o pagal žinomą jėgą apskaičiuojamas lenkimo momentas. Jungties standumas yra lenkimo momento ir pasisukimo kampo santykis. Taip pat, tik apkraunant sukimo momentu ir matuojant elemento susisukimo kampą, apskaičiuojamas jungties standumas sukimui.

Jungčių tarp elementų slopinimo koeficientai įvertinami apytiksliai. Nustačius savąjį elemento svyravimo dažnį ir svyravimo periodų skaičių prieš jam nuslopstant, parenkamas apytikslis slopinimo faktorius, ir pagal jį apskaičiuojamas jungties slopinimo koeficientas.

Aerodinaminiai paviršiai modelyje aprašomi pagal realaus lėktuvo aerodinaminių paviršių geometriją, darant prielaidą kad paviršius yra nulinio storio. Paviršių aerodinaminių profilių poliarės apskaičiuojamos naudojant programinę įrangą XFLR5 [49].

Žiūrint skrydžio bandymų vaizdo įrašus ir peržiūrint skrydžio metu įrašytus duomenis, parenkamas modeliuojamas skrydžio fragmentas. Modeliuojamo fragmento metu lėktuvo variklis turi būti išjungtas, jo pradžioje lėktuvas turi skristi kuo tiesesne linija (bet nebūtinai horizontaliai), nesisukti. Pasirinkus analizuojamą fragmentą, pradinė lėktuvo padėtis ir greitis nustatoma pagal vaizdo įrašą.

Sudarius lėktuvo modelį ir parinkus modeliuojamą skrydžio fragmentą, aeroservotamprumo analizė atliekama pagal 2 skyriuje aprašytą metodiką.

Modeliuojamo lėktuvo valdymo paviršių atlenkimas valdomas pagal skrydžio metu išsaugotus valdymo signalus. Valdymo signalo trukmė perskaičiuojama į paviršiaus atlenkimo kampą pagal 3.2 poskyryje aprašytu metodu surinktus valdymo paviršių kalibravimo duomenis.

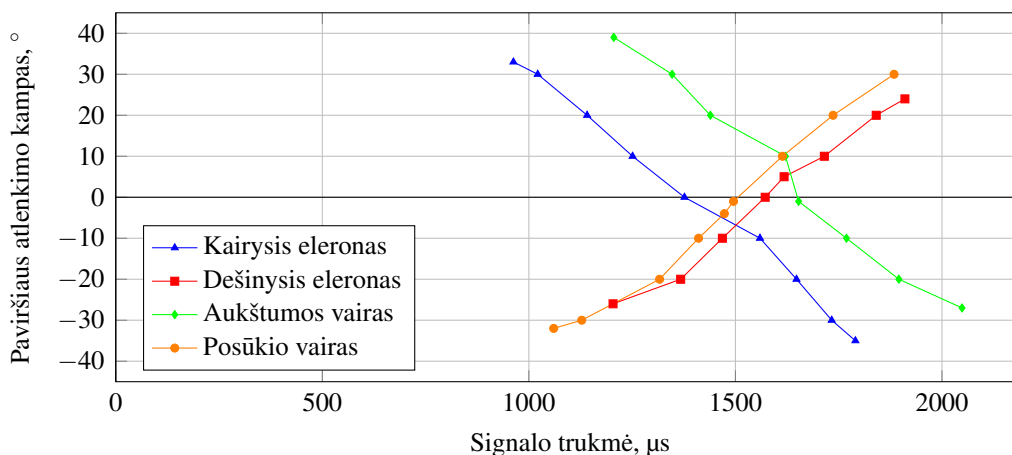
Modeliuojant į teksto failą išsaugoma lėktuvo padėtis erdvėje, jo kampiniai ir linijiniai greičiai ir pagreičiai.

Naudojant visus tuos pačius parametrus, taip pat atliekamas ir standus skrydžio modeliavimas, pagal metodiką aprašytą 2.2 poskyryje.

## 4. Tyrimo rezultatai

### 4.1. Valdymo paviršių kalibravimas

Pagal 3.2 poskyryje aprašytą metodiką buvo atliktas valdymo paviršių kalibravimas, jo rezultatai grafiškai pateikiami 20 pav.. Visų servo variklių priimamų signalų ilgis yra tarp 963  $\mu$ s ir 2049  $\mu$ s. Kairiojo eleronu atlenkimo ribos nuo  $-35^\circ$  iki  $32^\circ$ , o dešiniojo nuo  $-26^\circ$  iki  $24^\circ$ . Tokia eleronų asimetrija egzistuoja dėl mažų tiriamojo modelio gamybinių tikslumų. Aukštumos vairo atlenkimo ribos nuo  $-28^\circ$  iki  $39^\circ$ , nors jo atlenkimo asimetrija didelė, tai neigiamos įtakos lėktuvo valdymui neturi. Posūkio vairo atlenkimo kampo ribos nuo  $-32^\circ$  iki  $30^\circ$ , taigi posūkio vairo asimetrija yra nedidelė. Priklausomybės yra artimos tiesinėms, didžiausi netiesiškumai yra ties  $0^\circ$  aukštumos vairo ir kairiojo eleronu atvejais, taip pat ties atlenkimo kampo riba. Ties  $0^\circ$  esančio netiesiškumo tikėtina priežastis yra valdymo paviršiaus sujungimo su servovarikliu laisvumas. Ties atlenkimo kampo riba esantys netiesiškumai gali būti geometriniai (dėl nebegaliojančios mažų kampų prielaidos) arba dėl padidėjusių jėgų valdymo jungtyse (dėl valdymo paviršiaus standumo).



20 pav. Valdymo paviršių atlenkimo kampo priklausomybė nuo servovariklio gaunamos signalo trukmės

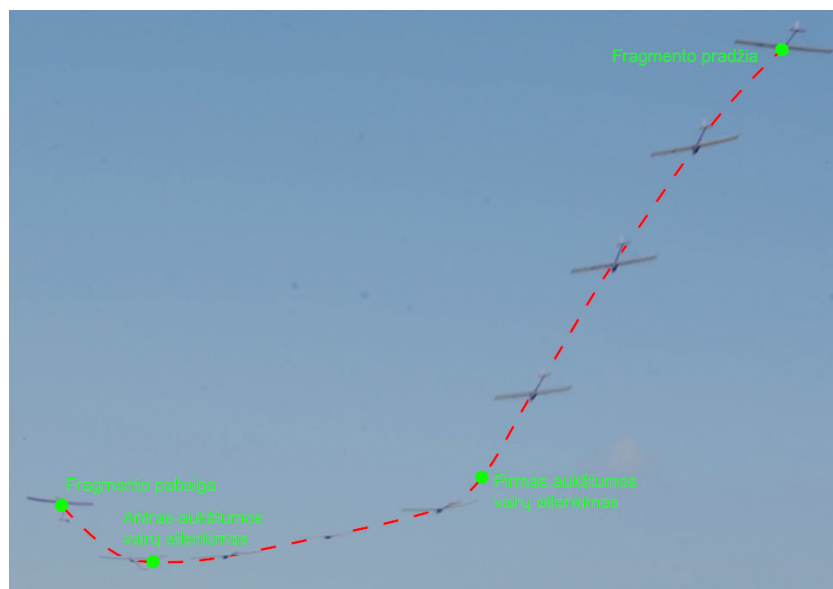
Kalibravimo metu surinkti duomenys įrašyti į tekstinį failą. Modeliavimo metu jie nuskaitomi programos ir naudojami perskaičiuoti skrydžio metu užfiksuotas valdymo signalų trukmes į valdymo paviršių atlenkimo kampus.

### 4.2. Atlikti skrydžio bandymai

Buvo sėkmingai atlikta viena skrydžio bandymų sesija, jos metu iš viso atlikti keturi skrydžiai. Peržiūrėjus skrydžio vaizdo įrašus ir mikrovaldiklio surinktus duomenis, buvo atrinkti du skrydžio fragmentai, tinkami palyginamajai analizei. Kad skrydžio fragmentas būtų tinkamas palyginimui, jo metu turi būti išjungtas variklis, taip pat lėktuvas turi patekti į stebinčių vaizdo kamerų kadra.

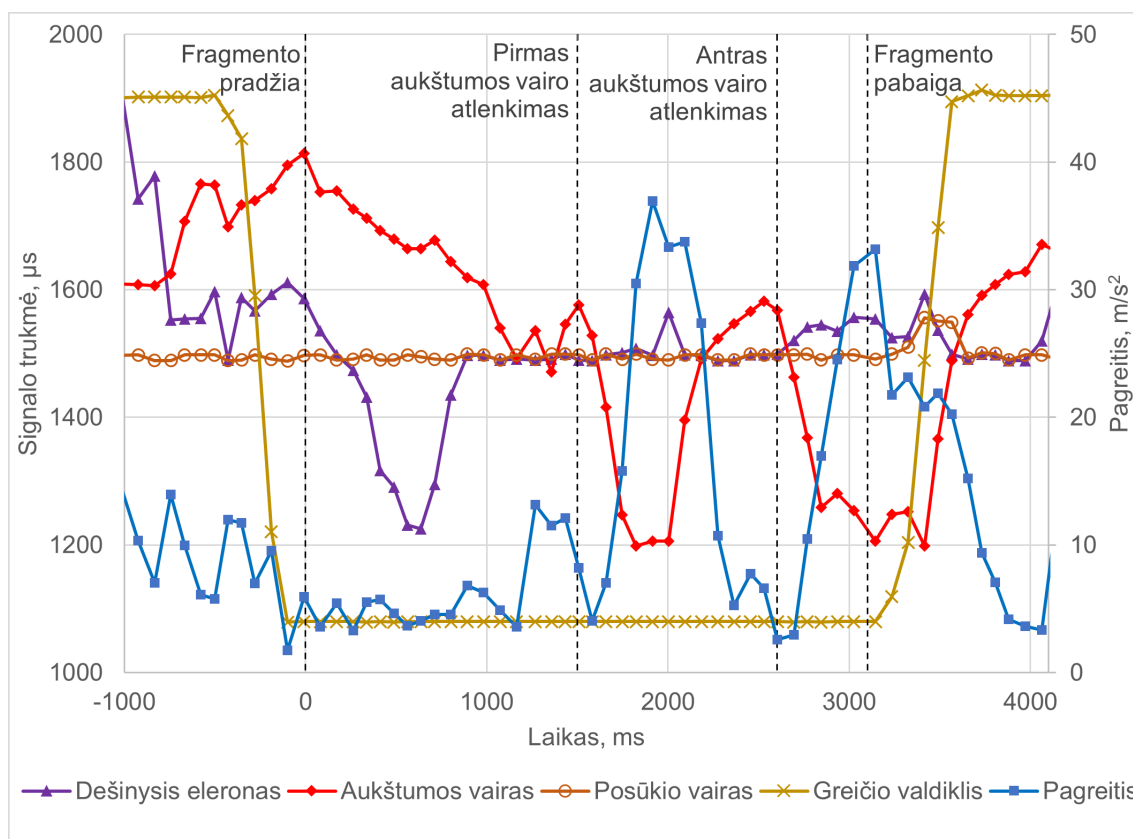
#### 4.2.1. Pirmasis analizuojamas fragmentas

Pirmojo analizei atrinkto fragmento bendras vaizdas pateiktas 21 pav., lėktuvo trajektorija pažymėta raudona punktyrine linija, svarbiausi etapai atskirti žaliais taškais. Galima matyti, kad fragmento pradžioje lėktuvas yra nuožulnioje žemėjimo trajektorijoje. Po žemėjimo pilnai atlenkiamas aukštumos vairo ir lėktuvas pereina į horizontalų sklendimą. Horizontalaus sklendimo gale vėl atlenkiamas aukštumos vairo ir lėktuvas pradeda aukštėti.



**21 pav.** Pirmojo analizuojamo skrydžio fragmento vaizdas (pateikta lėktuvo padėtis kas 0,5 s)

22 pav. pateikti šio skrydžio fragmento metu surinkti duomenys. Vertikaliomis punktyrinėmis linijomis grafike pažymėti pagrindiniai skrydžio etapai. Fragmentas prasideda iš karto išsijungus varikliui (Greičio valdiklio signalo trukmei sumažėjus iki minimalios). Jo pradžioje aukštumos vairas atlenktas žemyn, lėktuvas pikiruoja, bendras akselerometro fiksuojamas pagreitis lygus  $4,5 \text{ m/s}^2$ .



**22 pav.** Pirmojo analizuojamo skrydžio bandymo fragmento duomenys

Prieš atlenkiant aukštumos vairą aukštyn, trumpam atlenkiami eleronai, lėktuvas pasvyra kairėn. Prieš pat pilną aukštumos vairo atlenkimą yra trumpas pagreičio padidėjimas iki  $11,5 \text{ m/s}^2$ . Pirmą kartą pil-

nai atlenkus aukštumos vairą, pagreičio vertė staigiai pakyla iki  $37,0 \text{ m/s}^2$ , vairui grįžus į neutralią padėtį sumažėja iki  $7,7 \text{ m/s}^2$ . Antrą kartą atlenkus vairus pagreitis padidėja iki  $31,9 \text{ m/s}^2$ , tada fragmentas baigiasi, nes įjungiamas lėktuvo variklis.

#### 4.2.2. Antrasis analizuojamas fragmentas

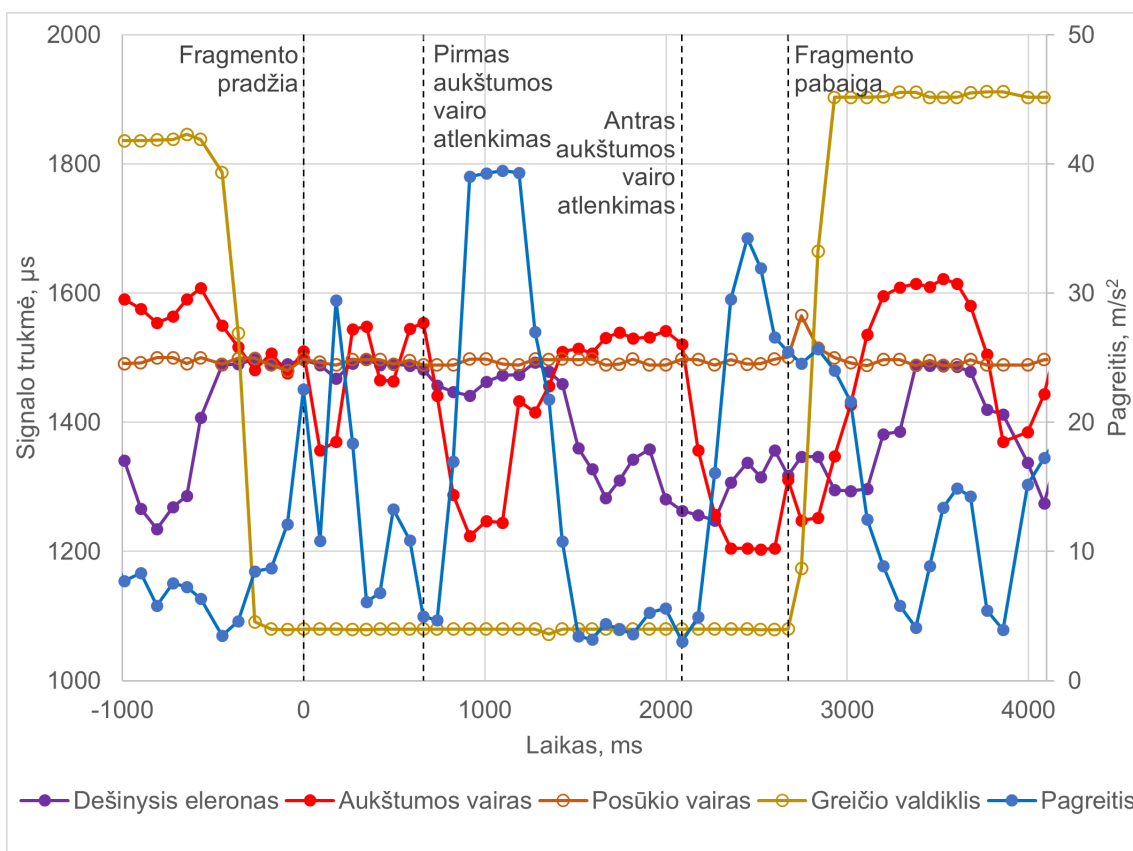
Antrojo analizei atrinkto fragmento bendras vaizdas pateiktas 23 pav.. Lėktuvo trajektorija pažymėta raudona punktyrine linija, svarbiausi manevro etapai atskirti žaliais taškais.



23 pav. Antrojo analizuojamo skrydžio fragmento vaizdas (pateikta lėktuvo padėtis kas 0,25 s)

Matyti, kad fragmento pradžioje lėktuvas stačiai žemėja, tada atlenkus aukštumos vairus pradeda šiek tiek aukštėti, tada lėtai vėl pereina į žemėjančią trajektoriją. Toliau vėl atlenkiami aukštumos vairo, bet prieš tai lėktuvas išskrenda iš kadro, todėl šis manevras neužfiksuotas.

24 pav. pateikti šio skrydžio fragmento metu surinkti duomenys. Vertikaliomis punktyrinėmis linijomis grafike pažymėti pagrindiniai skrydžio etapai.



24 pav. Antrojo analizuojamo skrydžio bandymo fragmento duomenys

Fragmentas prasideda praėjus apie 200 ms po variklio išsijungimo. Fragmento pradžioje visi lėktuvo valdymo paviršiai yra neutralioje padėtyje, bet iš karto fragmentui prasidėjus trumpam šiek tiek atlenkiamas aukštumos vairas, tai sukelia  $29,4 \text{ m/s}^2$  pagreitį. Po to vairams sugrįžus į neutralią padėtį pagreitis vėl sumažėja. Ties maždaug 700 ms stipriai atlenkiamas aukštumos vairas ir lėktuvas per eina iš nėrimo į horizontalų skrydį, manevro metu pasiekiamas  $39,5 \text{ m/s}^2$  pagreitis. Tada aukštumos vairas gražinamas į neutralią padėtį, pagreitis sumažėja iki  $3,2 \text{ m/s}^2$ . Lėktuvo padėtis šiek tiek pakoreguojama eleronais. Antrą kartą aukštumos vairas atlenkiamas ties 2100 ms, lėktuvas pradeda aukštėti, manevro metu pasiekia  $34,3 \text{ m/s}^2$  pagreitį. Lėktuvui lėtėjant pagreitis mažėja, tada įjungiamas variklis ir analizuojamas skrydžio fragmentas baigiasi.

### 4.3. Modeliavimo rezultatai

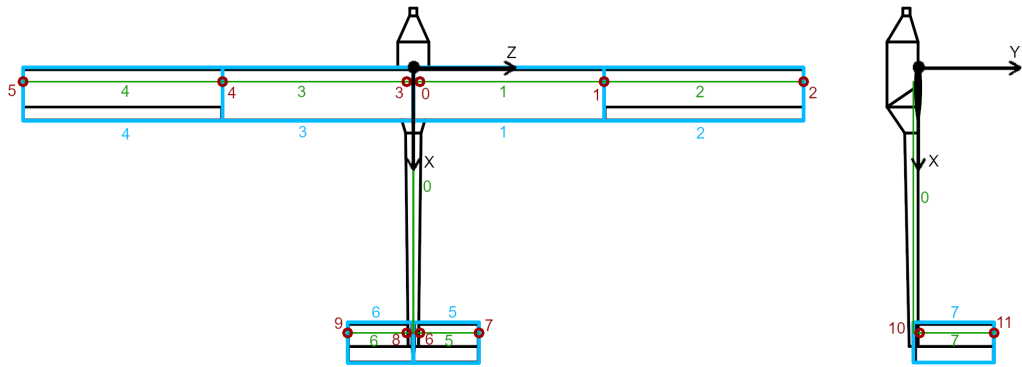
#### 4.3.1. Sudarytas lėktuvo modelis

Nuspręsta analizuojamą lėktuvą padalinti į 8 elementus. Kiekvieno elemento savybės aprašytos 6 lentelėje (plačiau savybės apibūdinamos 2.3.1 poskyryje). Kadangi visi nepagrindiniai elementų inercijos momentai yra nuliniai, lentelėje pateikiami tik pagrindiniai (sudarantys inercijos matricos įstrižainę). 0-nis elementas yra lėktuvo liemu, 1-as ir 2-as elementai dešinysis sparnas, 3-as ir 4-as kairysis sparnas, 5-as ir 6-as sudaro stabilizatorių, o 7-as kilj. Visi elementai, išskyrus liemens, yra aprašyti kaip aerodinaminiai paviršiai. 2-as, 4-as, 5-as, 6-as ir 7-as elementai yra aerodinaminiai paviršiai su valdymo paviršiais.

**6 lentelė.** Lėktuvo modelio elementus aprašantys kintamieji

El. Nr.	Masė, g	Padėtis, mm	Pagr. inercijos momentai, $10^6 \text{ g} \cdot \text{mm}^2$	Aerod. plokšt. išilgai mosto	Aerod. plokšt. išilgai stygos	Užsp. plokšt.	Santykinis užsp. stygos ilgis
0	1030	[115, -50, 0]	[1,9, 24,7, 24,7]	0	0	0	0
1	160	[90, 0, -425]	[7,5, 8,0, 0,53]	6	5	0	0
2	115	[90, 0, -1125]	[5,4, 5,8, 0,38]	6	5	2	0,2
3	160	[90, 0, 425]	[7,5, 8,0, 0,53]	6	5	0	0
4	115	[90, 0, 1125]	[5,4, 5,8, 0,38]	6	5	2	0,2
5	25	[1075, 1, -125]	[0,13, 0,18, 0,05]	5	5	2	0,4
6	25	[1075, 1, 125]	[0,13, 0,18, 0,05]	5	5	2	0,4
7	30	[1075, 150, 0]	[0,13, 0,18, 0,05]	5	5	2	0,4

Lentelė sudaryta atlikus lėktuvo matavimus pagal 3.3 poskyryje pateiktą metodiką. Bendras lėktuvo modelio elementų ir jungčių išsidėstymas pateiktas 25 pav., žaliai pažymėti elementai, mėlynai aerodinaminiai paviršiai, o raudonai jungtys.



25 pav. Elementų ir jungčių schema modeliuojamame lėktuve

Taip pat pagal 2.2 poskyryje aprašytą metodiką sudarytas standus lėktuvo modelis. Jo pirmojo elemento masė 1660 g, masės centro padėtis aprašoma vektoriumi [153, -28, 0] mm, o inercijos momentai aprašomi matrica

$$\begin{bmatrix} 380 & -6,2 & 0 \\ -6,2 & 474 & 0 \\ 0 & 0 & 100 \end{bmatrix} \cdot 10^6 \text{ g} \cdot \text{mm}^2.$$

Kadangi visas lėktuvas nėra paprasta geometrinė forma, ne visi jo šalutiniai inercijos momentai lygūs nuliui.

#### 4.3.2. Pirmojo atvejo modeliavimas

Pradinė lėktuvo kampinė padėtis nustatyta pagal skrydžio bandymų vaizdo įrašus. Kadangi modelyje nėra galimybės keisti pradinę lėktuvo kampinę padėtį, buvo pakeista laisvojo kritimo pagreičio kryptis, kad atitiktų modeliuojamą situaciją. Lėktuvo pradinis linijinis ir kampinis greičiai buvo nustatyti pagal vaizdo įrašą. Nustatytos pradinės sąlygos pateiktos 7 lentelėje.

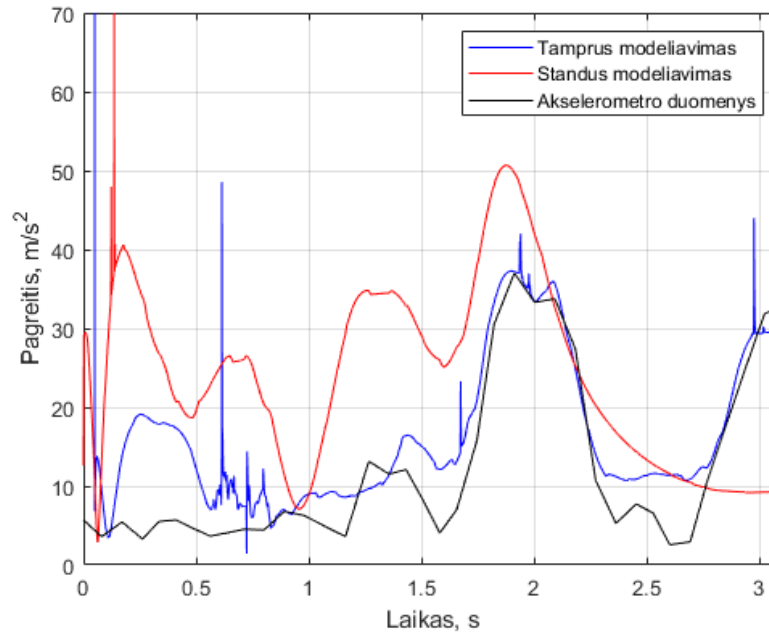
7 lentelė. Pirmojo atvejo modeliavimo sąlygos

Kintamasis	X vertė	Y vertė	Z vertė
Laisvojo kritimo pagreitis, m/s <sup>2</sup>	-1,67	-9,67	0,00
Pradinis greitis, m/s	-25,0	-2,4	0,0
Pradinis kampinis greitis, rad/s	0,8	-0,3	1,0

Pagal analizuojamo fragmento išsaugotus valdymo duomenis nustatytas pradinis laikas 482000 ms, skaičiavimo pabaigos laikas 485100 ms. Nustatytas laiko žingsnis 0,05 ms parenkant maksimalų laiko žingsnį, prie kurio atliekami skaičiavimai stabilūs. Patikrinimui atlikus skaičiavimus su dvigubai trumpesniu laiko žingsniu rezultatai nepakito, todėl galima teigti, kad žingsnis parinktas gerai.

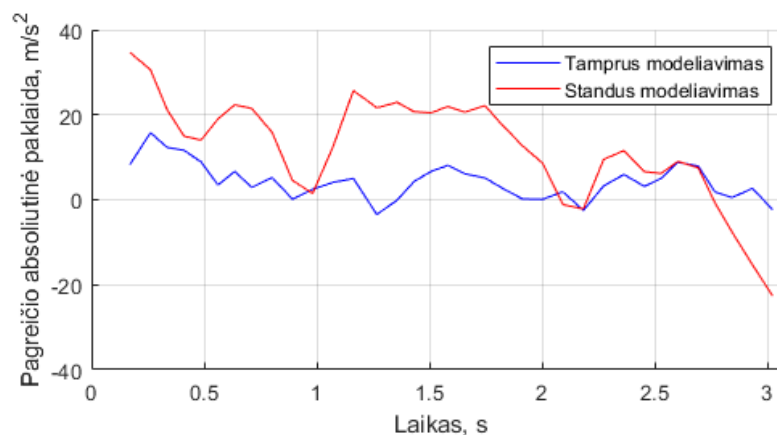
26 pav. pateikti pirmojo atvejo modeliavimo metu gauti pagreičiai. Juoda linija pažymėti skrydžio bandymų metu surinkti akselerometro duomenys (išsamiau eksperimentinė pagreičio kreivė apžvelgta 4.2.1 skyrelyje), mėlyna linija aeroservotamprumo modeliavimo metodu gauti pagreičiai, o raudona linija standaus modeliavimo metodu gauti pagreičiai.





**26 pav.** Pirmojo atvejo modeliavimo pagreičiai

Galima matyti, kad aeroservotamprumo modeliavimo metodu gauta kreivė žymiai artimesnė realiems pagreičiams, nei standaus modeliavimo metodu. Modeliuojant tamprų lėktuvą dėl atlenkto aukštumos vairo susisuka stabilizatorius, tai sumažina aukštumos vairo efektyvumą. Taip pat dėl deformacijų mažėja sparno kuriama keliamoji jėga. Modeliuojant standų lėktuvą aukštumos vairs veikia pilnu efektyvumu, ir sparnas kuria didesnę keliamąją jėgą, dėl to manevruojant patiriami didesni pagreičiai prie tų pačių vairų atlenkimo kampų. Dėl to analizuoto fragmento pradžioje standaus modeliavimo pagreičiai labai (iki  $34,7 \text{ m/s}^2$ ) viršija eksperimentinius. Dėl didelio indukcinio pasipriešinimo standžiai modeliuojamas lėktuvas praranda daug greičio ir fragmento pabaigoje pagreičiai yra žymiai mažesni nei eksperimentiniai. Tmpriai modeliuoto lėktuvo pagreičiai esant didelėms perkrovoms artimi (arba šiek tiek didesni) už eksperimentiškai nustatytus, o esant mažoms perkrovoms iki  $15,8 \text{ m/s}^2$  viršija eksperimentinius. 27 pav. pateiktos absoliutinės abiem modeliais apskaičiuoto pagreičio paklaidos.

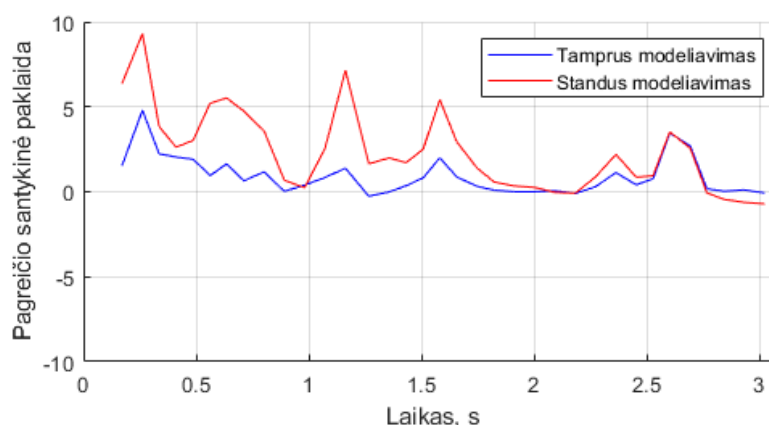


**27 pav.** Pirmojo atvejo modeliavimo pagreičių absoliutinė paklaida

Galima matyti, kad standaus modelio paklaidos per visą analizuojamą fragmentą yra didesnės nei tampraus modelio, išskyrus labai trumpus tarpus ties 1,0 s ir 2,1 s ir tarpą ties 2,8 s, kur standaus

modeliavimo pagreičio paklaida iš teigiamos pereina į neigiamą. Didžiausia aeroservotamprumo modelio absoliutinė pagreičio paklaida lygi  $15,8 \text{ m/s}^2$ , o standaus modelio didžiausia paklaida lygi  $34,7 \text{ m/s}^2$ , taigi aeroservotamprumo modelio maksimali paklaida 2,2 karto mažesnė. Standus modelis apskaičiuoja žymiai per didelį pagreitį skaičiavimų pradžioje ir žymiai per mažą skaičiavimų pabaigoje, praradus greitį. Aeroservotamprumo modelis beveik visada apskaičiuoja per didelį pagreitį, o absoliutinė paklaida žymiai didesnė esant mažiems pagreičiams. Vidutinis nuokrypis tamprinio modeliavimo atveju  $4,7 \text{ m/s}^2$ , o standžiojo modeliavimo atveju  $15,1 \text{ m/s}$ , taigi tamprinio modeliavimo atveju vidutinis nuokrypis 3,2 karto mažesnis.

28 pav. pateiktos santykinės abiem modeliais apskaičiuoto pagreičio paklaidos.



28 pav. Pirmojo atvejo modeliavimo pagreičių absoliutinė paklaida

Matyti, kad didžiausia santykinė standaus modelio paklaida žymiai didesnė už aeroservotamprumo modelio. Didžiausia santykinė standaus modelio paklaida 930,3 %, o didžiausia aeroservotamprumo modelio paklaida 479,6 %. Vidutinis santykinis aeroservotamprumo modelio nuokrypis 91,8 %, o standaus modelio 242,4 %, taigi aeroservotamprumo modelio vidutinis nuokrypis 2,6 karto mažesnis.

#### 4.3.3. Antrojo atvejo modeliavimas

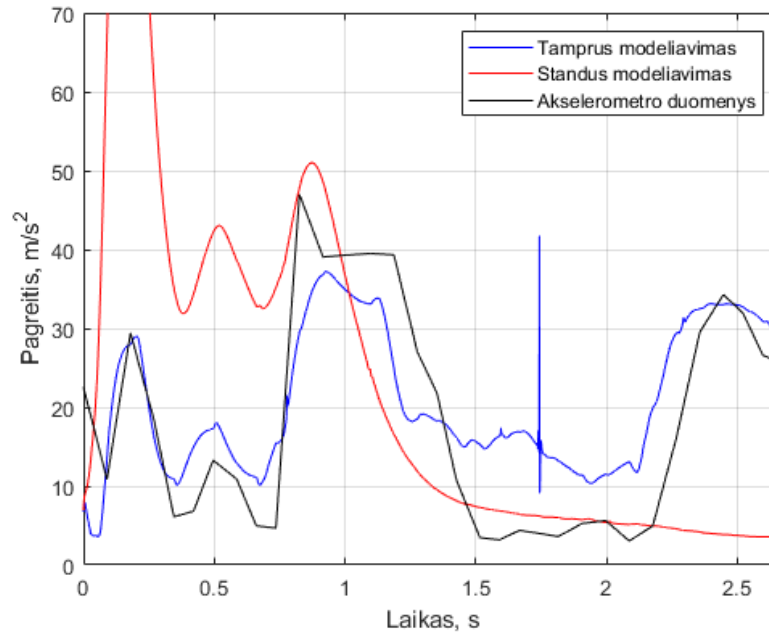
Pradinė lėktuvo padėtis ir greičiai nustatyti kaip ir pirmuoju modeliavimo atveju. Nustatytos pradinės sąlygos pateiktos 8 lentelėje.

8 lentelė. Antrojo atvejo modeliavimo sąlygos

Kintamasis	X vertė	Y vertė	Z vertė
Laisvojo kritimo pagreitis, $\text{m/s}^2$	-6,31	-7,51	0,00
Pradinis greitis, $\text{m/s}$	-30,0	0,0	0,0
Pradinis kampinis greitis, $\text{rad/s}$	0,0	0,0	0,0

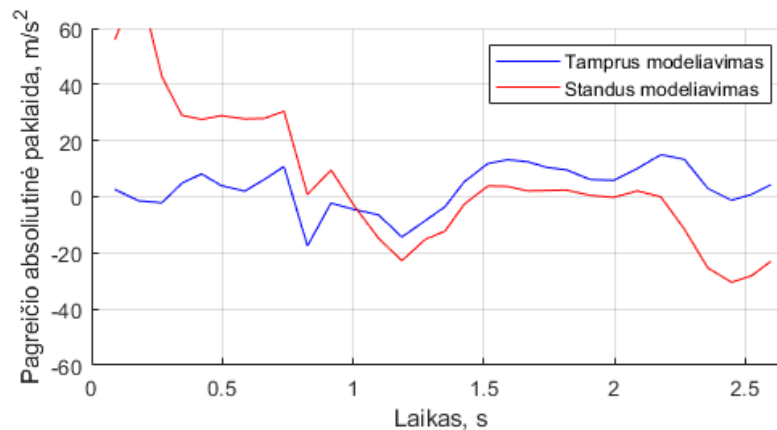
Nustatytas fragmento pradžios laikas 503100 ms, skaičiavimo pabaigos laikas 505773 ms. Nustatytas laiko žingsnis 0,05 ms kaip ir analizuojant pirmąjį fragmentą.

29 pav. pateikti antrojo atvejo modeliavimo metu gauti pagreičiai. Kaip ir pirmuoju atveju, juoda linija pažymėti skrydžio bandymų metu surinkti akselerometro duomenys (išsamiau eksperimentinė pagreičio kreivė apžvelgta 4.2.2 skyrelyje), mėlyna linija aeroservotamprumo modeliavimo metodu gauti pagreičiai, o raudona linija standaus modeliavimo metodu gauti pagreičiai.



**29 pav.** Antrojo atvejo modeliavimo pagreičiai

Kaip ir pirmuoju atveju, aeroservotamprumo modeliavimo kreivė žymiai artimesnė realiems pagreičiams, nei standaus modeliavimo. Standžiai modeliuoto lėktuvo pagreičiai labai dideli fragmento pradžioje ir labai sumažėja toliau, lėktuvui praradus greitį, taip pat kaip ir skaičiuojant pirmąjį atvejį. Tampriojo modeliavimo metodu gauti pagreičiai antruoju atveju šiek tiek mažesni nei eksperimentiniai ties dideliais pagreičiais, bet didesni esant mažiems pagreičiams. 30 pav. pateiktos absoliutinės abiem modeliais apskaičiuoto pagreičio paklaidos.

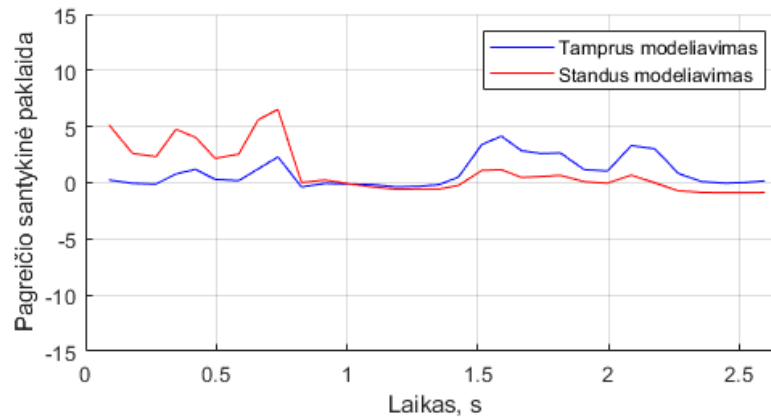


**30 pav.** Antrojo atvejo modeliavimo pagreičių absoliutinė paklaida

Galima matyti, kad standaus modelio paklaidos yra mažesnės ruože nuo 1,4 s iki 2,2 s. Tai yra ruožas, kuriame eksperimentiniai pagreičiai maži. Standaus modelio atveju pagreičiai taip pat maži dėl sumažėjusio lėktuvo greičio, o tampriojo modelio atveju lėktuvas šiame ruože dar išlaikęs greitį ir dėl modelio netikslumų patiria didesnius pagreičius nei eksperimentiniai. Didžiausia aeroservotamprumo modelio absoliutinė pagreičio paklaida  $15,0 \text{ m/s}^2$ , o standaus modelio didžiausia paklaida  $76,9 \text{ m/s}^2$ , taigi aeroservotamprumo modelio maksimali paklaida 5,1 karto mažesnė. Standus modelis apskaičiuoja žymiai per didelį pagreitį skaičiavimų pradžioje, vėliau, kai realiam skrydyje pagreičiai

maži dėl mažo aukštumos vairo atlenkimo, standžiajame modelyje pagreičiai maži dėl mažo lėktuvo greičio, ir paklaidos yra mažos. Kai fragmento gale vėl atlenkiami aukštumos vairai ir patiriami dideli pagreičiai, standaus modelio pagreičiai lieka žymiai per maži. Aeroservotamprumo modelis apskaičiuoja per didelį pagreitį esant mažiems pagreičiams ir per mažą pagreitį esant dideliems pagreičiams. Vidutinis nuokrypis tampriojo modeliavimo atveju  $7,2 \text{ m/s}^2$ , o standžiojo modeliavimo atveju  $18,1 \text{ m/s}$ , taigi tampriojo modeliavimo atveju vidutinis nuokrypis 2,5 karto mažesnis.

31 pav. pateiktos santykinės abiem modeliais apskaičiuoto pagreičio paklaidos.



31 pav. Antrojo atvejo modeliavimo pagreičių absoliutinė paklaida

Matyti, kad didžiausia santykinė standaus modelio paklaida žymiai didesnė už aeroservotamprumo modelio. Didžiausia santykinė standaus modelio paklaida  $652,2 \%$ , o didžiausia aeroservotamprumo modelio paklaida  $414,5 \%$ . Vidutinis santykinis aeroservotamprumo modelio nuokrypis  $108,0 \%$ , o standaus modelio  $150,0 \%$ , taigi aeroservotamprumo modelio vidutinis nuokrypis 1,4 karto mažesnis.

#### 4.4. Modelio efektyvumas

Analizuojant du eksperimentinio skrydžio fragmentus sukurtu aeroservotamprumo modeliu, gauti vidutiniai santykiniai pagreičio nuokrypiai lygūs  $91,8 \%$  ir  $108,0 \%$ . Analizuojant tuos pačius fragmentus standaus lėktuvo modeliu, gauti vidutiniai santykiniai pagreičio nuokrypiai  $242,4 \%$  ir  $150,0 \%$ . Naudojant tampraus lėktuvo modelį gauti atitinkamai 2,6 ir 1,4 karto mažesni santykiniai nuokrypiai, todėl galima daryti išvadą, kad atliekant mažo standumo lėktuvų skrydžio modeliavimą būtina atsižvelgti į lėktuvo struktūros deformacijas.

Nors naudojant sukurtą aeroservotamprumo modelį gautos mažesnės paklaidos, nei naudojant standaus lėktuvo modelį, jos vis tiek yra per didelės, kad modelis būtų naudingas realiai skrydžio analizei atlikti. Galimi didelių paklaidų šaltiniai yra naudojamas kvazi-statinis aerodinamikos modelis, netikslus pradinis skrydžio sąlygų nustatymas, vėjo gūsių įtaka, neatsižvelgimas į valdymo paviršių deformaciją dėl aerodinaminių apkrovų.

Toliau vystant sukurtą modelį, būtų ieškoma būdų kaip padidinti jo tikslumą, kad jis būtų tinkamas realaus skrydžio analizei. Tam būtų galima naudoti pažangesnius dinaminis aerodinamikos modelius, tiksliau nustatyti pradinę lėktuvo padėtį erdvėje, griežčiau kontroliuoti eksperimento sąlygas ir sudaryti detalesnę valdymo paviršių dinamiką aprašantį modelį.

## Išvados

1. Remiantis atlikta literatūros analize sudarytas fizikinis aeroservotamprumo modelis, kuris buvo įgyvendintas naudojant C programavimo kalbą. Aeroservotamprumo modelį nuspręsta paremti STM aerodinamikos modeliu ir standžių elementų sujungtų spyruoklėmis struktūriniu modeliu. Geras STM tikslumas palyginus su tradiciniais metodais ir maža reikalinga skaičiuojamoji galia leistų sukurtą aeroservotamprumo modelį naudoti daug iteracijų reikalaujančioms užduotims, tokioms kaip skrydžio valdymo algoritmų optimizavimas.
2. Modelio validavimo eksperimentiniams skrydžio bandymams vykdyti sukonstruotas bepilotis orlaivis, kurio masė 1,66 kg, sparnų mostas 3,0 m.
3. Atliktas bandymų orlaivio valdymo paviršių kalibravimas, kad būtų susieti servovariklių valdymo signalai su valdymo paviršių atlenkimo kampais. Gautos beveik tiesinės atlenkimo kampo priklausomybės nuo signalo trukmės, su netiesiškais ties neutralia ir kraštinėmis padėtimis.
4. Atlikti skrydžio bandymai, iš viso įvykdyti 4 sėkmingi skrydžiai, analizuojant jų duomenys išskirti 2 skrydžio epizodai, tinkami palyginamajai analizei. Skrydžiai vykdyti vėjuotą dieną, į tai atsižvelgiama modelyje nustačius laisvojo srauto tėkmės greitį.
5. Pagal bepilotį orlaivį sudarytas skaičiuojamasis lėktuvo modelis, padalintas į 8 elementus, iš kurių 7 yra aerodinaminiai paviršiai. Taip pat palyginimui sudarytas standus lėktuvo modelis, turintis tas pačias aerodinamines savybes kaip ir tamprus.
6. Atliktas skrydžio modeliavimas naudojant aeroservotamprumo modelį ir palyginamąjį standų modelį. Aeroservotamprumo modelio maksimali absoliutinė pagreičio paklaida palyginus su eksperimentiniais duomenimis  $11,9 \text{ m/s}^2$ . Standaus modelio maksimali absoliutinė paklaida  $44,4 \text{ m/s}^2$ , taigi maksimali aeroservotamprumo modelio paklaida 3,73 karto mažesnė.
7. Aeroservotamprumo modelio maksimali santykinė paklaida 444,7 %, o vidutinė santykinė paklaida 102,1 %. Standaus modelio maksimali santykinė paklaida 1233,5 %, o vidutinė santykinė paklaida 199,9 %. Aeroservotamprumo modelio vidutinė santykinė paklaida 1,96 karto mažesnė.
8. Analizuotas aeroservotamprumo modelis žymiai tikslesnis nei palyginimui naudotas standus skrydžio modelis. Nepaisant to, aeroservotamprumo modelio paklaidos vis tiek yra labai didelės, todėl modelį reikia toliau tobulinti, kad jis būtų naudingas. Numatomos pagrindinės didelių paklaidų priežastys yra naudojamas kvazi-statinis aerodinamikos modelis ir netikslus pradinės lėktuvo padėties nustatymas, tolimesnių tyrimų metu būtų galima ištaisyti šiuos modelio trūkumus.

## **Darbu sąrašas**

1. Publikacija „Development of Gust-Aleviation Algorithms Using Aeroservoelastic Analysis Methods“ į 2021-10-06–2021-10-08 vykusią tarptautinę konferenciją „Transport Means 2021“. Publikacijos kopija pateikta 2 priedas.

## Literatūra

1. RIJN, L. Design of Sailplanes Using Composite Materials. In *Comprehensive Composite Materials*. 2000, p. 243–277. ISBN 978-0-08-042993-9. doi: 10.1016/B0-08-042993-9/00114-5.
2. SLAYTON, R. ir G. SPINARDI. Radical Innovation in Scaling up: Boeing's Dreamliner and the Challenge of Socio-Technical Transitions. *Technovation*. 2015, vol. **47**. doi: 10.1016/j.technovation.2015.08.004.
3. *Aluminum 6061-T6; 6061-T651* [interaktyvus]. [žiūrėta 2021-06-11]. Prieiga per: <https://www.matweb.com/search/DataSheet.aspx?MatGUID=b8d536e0b9b54bd7b69e4124d8f1d20a&ckck=1>.
4. *Aluminum 7075-T6; 7075-T651* [interaktyvus]. [žiūrėta 2021-06-11]. Prieiga per: <https://www.matweb.com/search/DataSheet.aspx?MatGUID=4f19a42be94546b686bbf43f79c51b7d>.
5. *Carbon Fabric 160 g/M<sup>2</sup> (Style 442 Aero, Twill Weave) 100 Cm* [interaktyvus]. [žiūrėta 2021-06-11]. Prieiga per: <https://shop1.r-g.de/en/art/190225>.
6. *Carbon Fabric 200 g/M<sup>2</sup> (Style 450-5 Aero, Plain Weave) 100 Cm* [interaktyvus]. [žiūrėta 2021-06-11]. Prieiga per: <https://shop1.r-g.de/en/art/190229>.
7. *Glass Fabric 163 g/M<sup>2</sup> (Interglas 92110, Aero, Finish FK 144, TW) 100 Cm* [interaktyvus]. [žiūrėta 2021-06-11]. Prieiga per: Prieiga%20per%20internet%C4%85:%20https://shop1.r-g.de/en/art/190120.
8. *Glass Fabric 280 g/M<sup>2</sup> (Interglas 92125, Aero, Finish FK 144, TW) 100 Cm* [interaktyvus]. [žiūrėta 2021-06-11]. Prieiga per: <https://shop1.r-g.de/en/art/190138>.
9. WASZAK, M. R. ir D. K. SCHMIDT. Flight Dynamics of Aeroelastic Vehicles. *Journal of Aircraft* [interaktyvus]. American Institute of Aeronautics and Astronautics, 1988, vol. **25**(6), 563–571 [žiūrėta 2021-06-11]. doi: 10.2514/3.45623.
10. YEN, W. Y. ir R. L. SWAIM. *Effects of Dynamic Aeroelasticity on Handling Qualities and Pilot Rating - NASA-CR-155339* [interaktyvus]. Bendradarb. su PURDUE UNIV. 1977. [žiūrėta 2021-03-18]. Prieiga per: <http://hdl.handle.net/2060/19780005127>.
11. BRENNER, M. J. *Actuator and Aerodynamic Modeling for High-Angle-of-Attack Aeroservoelasticity - NASA-TM-4493* [interaktyvus]. Bendradarb. su NASA DRYDEN FLIGHT RESEARCH FACILITY. Sponsoring Organization: NASA Dryden Flight Research Facility, 1993 [žiūrėta 2021-02-22]. Prieiga per: <http://hdl.handle.net/2060/19940008782>.
12. MUKHOPADHYAY, V. ir E. LIVNE. Aeroservoelasticity. In *Encyclopedia of Aerospace Engineering* [interaktyvus]. American Cancer Society, 2010 [žiūrėta 2021-06-17]. ISBN 978-0-470-68665-2. doi: 10.1002/9780470686652.eae154.
13. LI, H., R. HUANG, Y. ZHAO ir H. HU. Maneuver Load Alleviation for High Performance Aircraft Robust to Flight Condition Variations. *Journal of Vibration & Control*. Sage Publications, Ltd, 2019, vol. **25**(5), 1044–1058. ISSN 1077-5463. doi: 10.1177/1077546318810033.
14. LIVET, T., P. IVANOV, F. KUBICA ir J. F. MAGNI. Robust Flight Control Desing With Respect to Delays and Control Efficiencies. *IFAC Proceedings Volumes*. 1994, vol. **27**(13), 273–278. ISSN 1474-6670. doi: 10.1016/S1474-6670(17)45812-5.

15. MAO, Q., L. DOU, B. TIAN ir Q. ZONG. Reentry Attitude Control for a Reusable Launch Vehicle with Aeroservoelastic Model Using Type-2 Adaptive Fuzzy Sliding Mode Control. *International Journal of Robust & Nonlinear Control*. John Wiley & Sons, Inc, 2018, vol. **28**(18), 5858–5876. ISSN 1049-8923. doi: 10.1002/rnc.4349.
16. KOPSAFTOPOULOS, F. Data-Driven Stochastic Identification for Fly-by-feel Aerospace Structures: Critical Assessment of Non-parametric and Parametric Approaches. In *AIAA Scitech 2019 Forum* [interaktyvus]. American Institute of Aeronautics and Astronautics, [s. a.] [žiūrėta 2021-06-17]. doi: 10.2514/6.2019-1534.
17. MANGALAM, A. S. ir M. J. BRENNER. Fly-by-Feel Sensing and Control: Aeroservoelasticity. In *AIAA Atmospheric Flight Mechanics Conference* [interaktyvus]. American Institute of Aeronautics and Astronautics, [s. a.] [žiūrėta 2021-06-17]. doi: 10.2514/6.2014-2189.
18. BORGLUND, D. ir J. KUTTENKEULER. Active Wing Flutter Suppression Using A Trailing Edge Flap. *Journal of Fluids and Structures* [interaktyvus]. 2002, vol. **16**(3), 271–294 [žiūrėta 2021-03-18]. ISSN 0889-9746. doi: 10.1006/jfls.2001.0426.
19. ZHANG, W. ir Z. YE. Control Law Design for Transonic Aeroservoelasticity. *Aerospace science and technology*. Elsevier SAS, 2007, vol. **11**(2), 136–145. ISSN 1270-9638. doi: 10.1016/j.ast.2006.12.004.
20. FAN, Y., J. ZHU, C. HU, X. MENG, X. WANG ir A. GE. Aeroservoelastic Model Based Active Control for Large Civil Aircraft. *Science China Technological Sciences*. Heidelberg: SP Science China Press, 2010, vol. **53**(4), 1126–1137. ISSN 1674-7321. doi: 10.1007/s11431-010-0001-z.
21. GANG, C., S. JIAN ir L. YUEMING. Active Flutter Suppression Control Law Design Method Based on Balanced Proper Orthogonal Decomposition Reduced Order Model. *Nonlinear Dynamics*. Springer Nature, 2012, vol. **70**(1), 1–13. ISSN 0924-090X. doi: 10.1007/s11071-012-0392-4.
22. BRILLANTE, C. ir A. MANNARINO. Improvement of Aeroelastic Vehicles Performance through Recurrent Neural Network Controllers. *Nonlinear Dynamics* [interaktyvus]. 2016, vol. **84**(3), 1479–1495 [žiūrėta 2021-03-18]. ISSN 1573-269X. doi: 10.1007/s11071-015-2583-2.
23. FONTE, F., S. RICCI ir P. MANTEGAZZA. Gust Load Alleviation for a Regional Aircraft Through a Static Output Feedback. *Journal of Aircraft*. American Institute of Aeronautics & Astronautics, 2015, vol. **52**(5), 1559–1575. ISSN 0021-8669. doi: 10.2514/1.C032995.
24. ZHANG, K. ir A. BEHAL. Continuous Robust Control for Aeroelastic Vibration Control of a 2-D Airfoil under Unsteady Flow. *Journal of vibration and control*. London, England: SAGE Publications, 2016, vol. **22**(12), 2841–2860. ISSN 1077-5463. doi: 10.1177/1077546314554821.
25. ZHANG, K., P. MARZOCCA ir A. BEHAL. Adaptive Aeroelastic Control of Nonlinear Airfoil with Multiple Flaps under Unsteady Flow. *Journal of vibration and control*. London, England: SAGE Publications, 2017, vol. **23**(10), 1593–1606. ISSN 1077-5463. doi: 10.1177/1077546315597180.
26. PATIL, M. J. Limit-Cycle Oscillations of Aircraft Caused by Flutter-Induced Drag. *Journal of Aircraft* [interaktyvus]. American Institute of Aeronautics and Astronautics, 2004, vol. **41**(3), 571–576 [žiūrėta 2021-06-17]. doi: 10.2514/1.11482.



27. ARENA, G., R. M. J. GROH, A. PIRRERA, T. TURNER, W. SCHOLTEN ir D. HARTL. Design of Shape-Adaptive Deployable Slat-Cove Filler for Airframe Noise Reduction. *Journal of Aircraft* [interaktyvus]. American Institute of Aeronautics and Astronautics, 2021, 1–17 [žiūrėta 2021-06-17]. doi: 10.2514/1.C036070.
28. TEWARI, A. Aeroservoelasticity. In Ashish TEWARI (sud.). *Aeroservoelasticity: Modeling and Control* [interaktyvus]. New York, NY: Springer, 2015, p. 1–11 [žiūrėta 2021-06-17]. Control Engineering. ISBN 978-1-4939-2368-7. doi: 10.1007/978-1-4939-2368-7\_1.
29. FUNG, Y. C. *An Introduction to the Theory of Aeroelasticity*. Courier Dover Publications, 2008. ISBN 978-0-486-46936-2.
30. THEODORSEN, T. *General Theory of Aerodynamic Instability and the Mechanism of Flutter*. NASA. Ames Res. Center Classical Aerodyn. Theory; p 291-311, 1934-05-02. Nr. NACA-TR-496.
31. TANG, D., E. H. DOWELL ir L. N. VIRGIN. LIMIT CYCLE BEHAVIOR OF AN AIRFOIL WITH A CONTROL SURFACE. *Journal of Fluids and Structures* [interaktyvus]. 1998, vol. **12**(7), 839–858 [žiūrėta 2021-06-15]. ISSN 0889-9746. doi: 10.1006/jfls.1998.0174.
32. GUPTA, K. K., M. J. BRENNER ir L. S. VOELKER. Integrated Aeroservoelastic Analysis Capability with X-29A Comparisons. *Journal of Aircraft* [interaktyvus]. American Institute of Aeronautics and Astronautics, 1989, vol. **26**(1), 84–90 [žiūrėta 2021-06-15]. doi: 10.2514/3.45726.
33. ZIMMERMANN, H. Aeroservoelasticity. *Computer Methods in Applied Mechanics and Engineering* [interaktyvus]. 1991, vol. **90**(1), 719–735 [žiūrėta 2021-03-12]. ISSN 0045-7825. doi: 10.1016/0045-7825(91)90181-5.
34. NOLL, T. E. *Aeroservoelasticity - NASA-TM-102620* [interaktyvus]. Bendradarb. su NASA LANGLEY RESEARCH CENTER. Sponsoring Organization: NASA Langley Research Center, 1990 [žiūrėta 2021-02-22]. Prieiga per: <http://hdl.handle.net/2060/19900009911>.
35. KARPEL, M. Reduced-Order Models for Integrated Aeroservoelastic Optimization. *Journal of Aircraft* [interaktyvus]. American Institute of Aeronautics and Astronautics, 1999, vol. **36**(1), 146–155 [žiūrėta 2021-06-15]. doi: 10.2514/2.2420.
36. WEISHAUPL, C. ir B. LASCHKA. Small Disturbance Euler Simulations for Delta Wing Unsteady Flows Due to Harmonic Oscillations. *Journal of Aircraft* [interaktyvus]. American Institute of Aeronautics and Astronautics, 2004, vol. **41**(4), 782–789 [žiūrėta 2021-06-17]. doi: 10.2514/1.12601.
37. MUNTEANU, S., J. RAJADAS, C. NAM ir A. CHATTOPADHYAY. Reduced-Order-Model Approach for Aeroelastic Analysis Involving Aerodynamic and Structural Nonlinearities. *AIAA Journal* [interaktyvus]. American Institute of Aeronautics and Astronautics, 2005, vol. **43**(3), 560–571 [žiūrėta 2021-06-17]. ISSN 0001-1452. doi: 10.2514/1.10971.
38. MOR, M. ir E. LIVNE. Minimum-State Unsteady Aerodynamics for Aeroservoelastic Configuration Shape Optimization of Flight Vehicles. *AIAA Journal* [interaktyvus]. American Institute of Aeronautics and Astronautics, 2005, vol. **43**(11), 2299–2308 [žiūrėta 2021-06-17]. ISSN 0001-1452. doi: 10.2514/1.10005.

39. MOULIN, B. Modeling of Aeroservoelastic Systems with Structural and Aerodynamic Variations. *AIAA Journal* [interaktyvus]. American Institute of Aeronautics and Astronautics, 2005, vol. **43**(12), 2503–2513 [žiūrėta 2021-06-17]. ISSN 0001-1452. doi: 10.2514/1.15023.
40. THOMAS, J. P., E. H. DOWELL ir K. C. HALL. Static/Dynamic Correction Approach for Reduced-Order Modeling of Unsteady Aerodynamics. *Journal of Aircraft* [interaktyvus]. American Institute of Aeronautics and Astronautics, 2006, vol. **43**(4), 865–878 [žiūrėta 2021-06-17]. doi: 10.2514/1.12349.
41. BOËLY, N. ir R. M. BOTEZ. New Approach for the Identification and Validation of a Nonlinear F/A-18 Model by Use of Neural Networks. *IEEE Transactions on Neural Networks*. IEEE, 2010, vol. **21**(11), 1759–1765. ISSN 1045-9227. doi: 10.1109/TNN.2010.2071398.
42. *Subsonic Aerofoil and Wing Theory | Aerodynamics for Students* [interaktyvus]. [žiūrėta 2021-08-30]. Prieiga per: <http://www.aerodynamics4students.com/subsonic-aerofoil-and-wing-theory/3d-vortex-lattice-method.php>.
43. MAXWELL, James Clerk. *On Physical Lines of Force*. London, 1861.
44. *Simple DirectMedia Layer - Homepage* [interaktyvus]. [žiūrėta 2021-08-31]. Prieiga per: <https://www.libsdl.org/>.
45. *BLAS (Basic Linear Algebra Subprograms)* [interaktyvus]. [žiūrėta 2021-08-31]. Prieiga per: <https://www.netlib.org/blas/>.
46. *LAPACK—Linear Algebra PACKage* [interaktyvus]. [žiūrėta 2021-08-31]. Prieiga per: <https://www.netlib.org/lapack/>.
47. *LSM6DS3TR STMicroelectronics | Mouser* [interaktyvus]. Mouser Electronics. [žiūrėta 2022-05-08]. Prieiga per: <https://eu.mouser.com/ProductDetail/511-LSM6DS3TR>.
48. *ARDUINO UNO WiFi REV2* [interaktyvus]. Arduino Official Store. [žiūrėta 2022-05-08]. Prieiga per: <http://store.arduino.cc/products/arduino-uno-wifi-rev2>.
49. *XFLR5* [interaktyvus]. [žiūrėta 2021-08-31]. Prieiga per: <http://www.xflr5.tech/xflr5.htm>.

## Priedai

### 1 priedas. Aerodinaminių jėgų skaičiavimo kodas

```
1
2 //VLM panel struct declaration
3 struct vlm_panel{
4     int pb;
5     double k, w, A;
6     double normal[3], v1p[3], v2p[3], ip[3], v[3], BV[3], SV1[3], SV2[3], kv[3], wv[3];
7 };
8
9 //structure to pass arguments to parallelized VLM thread
10 struct vlm_thread_st{
11     struct beam *b;
12     struct vlm_panel *vlmp;
13     double air_density;
14     int panel_count;
15     int beam_count;
16     double *A;
17     int i0, in;
18 };
19
20 //VLM mesher function
21 int vlm_mesh(struct node *n, struct beam *b, struct vlm_panel *vlm_panelm, int bn, int pnn, int
panel_count){
22     int cnt = panel_count;
23     int tnn = 0;
24     int tbn = 0;
25     if(n[pnn].beams[0] == n[pnn].beams[1] && pnn != -1){
26         return cnt;
27     }
28     for(int i = 0; i < b[bn].node_number; i+=1){
29         tnn = b[bn].nodes[i];
30         tbn = n[tnn].beams[0];
31         if(pnn != tnn){
32             //paneliu sudarymas
33             if(pnn >= 0){
34                 if(n[pnn].aero && n[tnn].aero){
35                     //sparno segmento krastiniu vektoriu apskaiciavimas
36                     double dx1[3], dx2[3], dz1[3], dz2[3], dx10[3], dx20[3], dx1f[3], dx2f[3], wa
[3], fr[3], wr[3], frq[4], wrq[4], frqi[4], wrqi[4];
37
38                     //sparno krastiniu vektoriai
39                     memcpy(dx1, vrot(n[pnn].rquat, n[pnn].lcl, n[pnn].irquat), 3*sizeof(double));
40                     memcpy(dx2, vrot(n[tnn].rquat, n[tnn].lcl, n[tnn].irquat), 3*sizeof(double));
41                     memcpy(dz1, n[tnn].gcpos, 3*sizeof(double));
42                     cblas_daxpy(3, -1., n[pnn].gcpos, 1, dz1, 1);
43                     memcpy(dz2, n[tnn].gcpos, 3*sizeof(double));
44                     cblas_daxpy(3, 1., dx2, 1, dz2, 1);
45                     cblas_daxpy(3, -1., n[pnn].gcpos, 1, dz2, 1);
46                     cblas_daxpy(3, -1., dx1, 1, dz2, 1);
47
48                     //sparno asis
49                     cblas_dcopy(3, zero_v, 1, wa, 1);
50                     cblas_daxpy(3, 0.5/cblas_dnrm2(3, dz1, 1), dz1, 1, wa, 1);
51                     cblas_daxpy(3, 0.5/cblas_dnrm2(3, dz2, 1), dz2, 1, wa, 1);
52
53                     cblas_dcopy(3, wa, 1, wr, 1);
54                     cblas_dscal(3, b[tbn].wa, wr, 1);
55                     cblas_dcopy(4, rotq(wr), 1, wrq, 1);
56                     cblas_dcopy(4, invq(wrq), 1, wrqi, 1);
57
58                     cblas_dcopy(3, vrot(wrq, dx1, wrqi), 1, dx1, 1);
59                     cblas_dcopy(3, vrot(wrq, dx2, wrqi), 1, dx2, 1);
60
```

```

61 //uzsparnio sudarymo vektoriai
62 cblas_dcopy(3, dx1, 1, dx10, 1);
63 cblas_dscal(3, 1.-b[tnb].cf, dx10, 1);
64 cblas_dcopy(3, dx2, 1, dx20, 1);
65 cblas_dscal(3, 1.-b[tnb].cf, dx20, 1);
66
67 cblas_dcopy(3, wa, 1, fr, 1);
68 cblas_dscal(3, b[tnb].fa, fr, 1);
69 cblas_dcopy(4, rotq(fr), 1, frq, 1);
70 cblas_dcopy(4, invq(fr), 1, frqi, 1);
71
72 cblas_dcopy(3, vrot(fr, dx1, frqi), 1, dx1f, 1);
73 cblas_dscal(3, b[tnb].cf, dx1f, 1);
74 cblas_dcopy(3, vrot(fr, dx2, frqi), 1, dx2f, 1);
75 cblas_dscal(3, b[tnb].cf, dx2f, 1);
76
77 //paneles ploto apskaiciavimas
78 double e1[3], e2[3];
79 double A0, Af;
80 //fiksuota panele
81 A0 = 0.;
82 cblas_dcopy(3, dx10, 1, e1, 1);
83 cblas_dscal(3, 1/(double)b[tnb].xpan0, e1, 1);
84 cblas_dcopy(3, dz1, 1, e2, 1);
85 cblas_dscal(3, 1/(double)b[tnb].zpan, e2, 1);
86 A0 += cblas_dnrm2(3, vcross(e1, e2), 1)/2.;
87 cblas_dcopy(3, dx20, 1, e1, 1);
88 cblas_dscal(3, 1/(double)b[tnb].xpan0, e1, 1);
89 cblas_dcopy(3, dz2, 1, e2, 1);
90 cblas_dscal(3, 1/(double)b[tnb].zpan, e2, 1);
91 A0 += cblas_dnrm2(3, vcross(e1, e2), 1)/2.;
92 //printf("PLOTAS %f ", A0);
93 //uzsparnio panele
94 Af = 0.;
95 cblas_dcopy(3, dx1f, 1, e1, 1);
96 cblas_dscal(3, 1/(double)b[tnb].xpanf, e1, 1);
97 cblas_dcopy(3, dz1, 1, e2, 1);
98 cblas_dscal(3, 1/(double)b[tnb].zpan, e2, 1);
99 Af += cblas_dnrm2(3, vcross(e1, e2), 1)/2.;
100 cblas_dcopy(3, dx2f, 1, e1, 1);
101 cblas_dscal(3, 1/(double)b[tnb].xpanf, e1, 1);
102 cblas_dcopy(3, dz2, 1, e2, 1);
103 cblas_dscal(3, 1/(double)b[tnb].zpan, e2, 1);
104 Af += cblas_dnrm2(3, vcross(e1, e2), 1)/2.;
105
106 for(int k = 0; k < b[tnb].zpan; k += 1){
107     //paneles fiksuotoje sparne dalyje
108     for(int j = 0; j < b[tnb].xpan0; j += 1){
109         //2 krastines vektoriaus itakos koeficientai
110         double wxv1, wxv2, wxi, wzv1, wzv2, wzi;
111         //krastines vektoriaus daugiklis
112         double axv1, axv2, axi, azv1, azv2, azi;
113         wxv1 = (double)k/(double)b[tnb].zpan;
114         wxv2 = ((double)k + 1.0)/(double)b[tnb].zpan;
115         wxi = ((double)k + 0.5)/(double)b[tnb].zpan;
116         wzv1 = ((double)j + 0.25)/(double)b[tnb].xpan0;
117         wzv2 = wzv1;
118         wzi = ((double)j + 0.75)/(double)b[tnb].xpan0;
119         axv1 = wzv1;
120         axv2 = wzv2;
121         axi = wzi;
122         azv1 = wxv1;
123         azv2 = wxv2;
124         azi = wxi;
125
126         memcpy(vlm_panelm[cnt].v1p, n[pnn].pos, 3*sizeof(double));

```

```

127 memcopy(vlm_panelm[cnt].v2p, n[pnn].pos, 3*sizeof(double));
128 memcopy(vlm_panelm[cnt].ip, n[pnn].pos, 3*sizeof(double));
129
130 //pirmo sukurio prisijungimo tasko apskaiciavimas
131 cblas_daxpy(3, (1.0-wxv1)*axv1, dx10, 1, vlm_panelm[cnt].v1p, 1);
132 cblas_daxpy(3, (wxv1)*axv1, dx20, 1, vlm_panelm[cnt].v1p, 1);
133 cblas_daxpy(3, (1.0-wzv1)*azv1, dz1, 1, vlm_panelm[cnt].v1p, 1);
134 cblas_daxpy(3, (wzv1)*azv1, dz2, 1, vlm_panelm[cnt].v1p, 1);
135
136 //antro sukurio prisijungimo tasko apskaiciavimas
137 cblas_daxpy(3, (1.0-wxv2)*axv2, dx10, 1, vlm_panelm[cnt].v2p, 1);
138 cblas_daxpy(3, (wxv2)*axv2, dx20, 1, vlm_panelm[cnt].v2p, 1);
139 cblas_daxpy(3, (1.0-wzv2)*azv2, dz1, 1, vlm_panelm[cnt].v2p, 1);
140 cblas_daxpy(3, (wzv2)*azv2, dz2, 1, vlm_panelm[cnt].v2p, 1);
141
142 //integracijos tasko apskaiciavimas
143 cblas_daxpy(3, (1.0-wxi)*axi, dx10, 1, vlm_panelm[cnt].ip, 1);
144 cblas_daxpy(3, (wxi)*axi, dx20, 1, vlm_panelm[cnt].ip, 1);
145 cblas_daxpy(3, (1.0-wzi)*azi, dz1, 1, vlm_panelm[cnt].ip, 1);
146 cblas_daxpy(3, (wzi)*azi, dz2, 1, vlm_panelm[cnt].ip, 1);
147
148 //paneles normales apskaiciavimas
149 double R1[3], R2[3];
150 cblas_dcopy(3, vlm_panelm[cnt].v1p, 1, R1, 1);
151 cblas_daxpy(3, -1., vlm_panelm[cnt].ip, 1, R1, 1);
152 cblas_dcopy(3, vlm_panelm[cnt].v2p, 1, R2, 1);
153 cblas_daxpy(3, -1., vlm_panelm[cnt].ip, 1, R2, 1);
154 cblas_dcopy(3, vcross(R2, R1), 1, vlm_panelm[cnt].normal, 1);
155 cblas_dscal(3, 1./cblas_dnrm2(3, vlm_panelm[cnt].normal, 1), vlm_panelm[
cnt].normal, 1);
156
157 vlm_panelm[cnt].A = A0;
158 vlm_panelm[cnt].pb = tbn;
159 vlm_panelm[cnt].k = cblas_dnrm2(3, dz1, 1)/(double)b[tbn].zpan;
160 cblas_dcopy(3, dz1, 1, vlm_panelm[cnt].kv, 1);
161 cblas_dscal(3, 1./(double)b[tbn].zpan, vlm_panelm[cnt].kv, 1);
162 cnt += 1;
163 }
164
165 //paneles uzsparnyje
166 for(int j = 0; j < b[tbn].xpanf; j += 1){
167     //2 krastines vektoriaus itakos koeficientai
168     double wxv1, wxv2, wxi, wzv1, wzv2, wzi;
169     //krastines vektoriaus daugiklis
170     double axv1, axv2, axi, azv1, azv2, azi;
171     wxv1 = (double)k/(double)b[tbn].zpan;
172     wxv2 = ((double)k + 1.0)/(double)b[tbn].zpan;
173     wxi = ((double)k + 0.5)/(double)b[tbn].zpan;
174     wzv1 = ((double)j + 0.25)/(double)b[tbn].xpanf;
175     wzv2 = wzv1;
176     wzi = ((double)j + 0.75)/(double)b[tbn].xpanf;
177     axv1 = wzv1;
178     axv2 = wzv2;
179     axi = wzi;
180     azv1 = wxv1;
181     azv2 = wxv2;
182     azi = wxi;
183
184     memcopy(vlm_panelm[cnt].v1p, n[pnn].pos, 3*sizeof(double));
185     memcopy(vlm_panelm[cnt].v2p, n[pnn].pos, 3*sizeof(double));
186     memcopy(vlm_panelm[cnt].ip, n[pnn].pos, 3*sizeof(double));
187
188     //pirmo sukurio prisijungimo tasko apskaiciavimas
189     cblas_daxpy(3, (1.0-wxv1), dx10, 1, vlm_panelm[cnt].v1p, 1);
190     cblas_daxpy(3, (1.0-wxv1)*axv1, dx1f, 1, vlm_panelm[cnt].v1p, 1);
191     cblas_daxpy(3, (wxv1), dx20, 1, vlm_panelm[cnt].v1p, 1);

```

```

192         cblas_daxpy(3, (wxv1)*axv1, dx2f, 1, vlm_panelm[cnt].v1p, 1);
193         cblas_daxpy(3, (1.0-wzv1)*azv1, dz1, 1, vlm_panelm[cnt].v1p, 1);
194         cblas_daxpy(3, (wzv1)*azv1, dz2, 1, vlm_panelm[cnt].v1p, 1);
195
196         //antro sukurio prisijungimo tasko apskaiciavimas
197         cblas_daxpy(3, (1.0-wxv2), dx10, 1, vlm_panelm[cnt].v2p, 1);
198         cblas_daxpy(3, (1.0-wxv2)*axv2, dx1f, 1, vlm_panelm[cnt].v2p, 1);
199         cblas_daxpy(3, (wxv2), dx20, 1, vlm_panelm[cnt].v2p, 1);
200         cblas_daxpy(3, (wxv2)*axv2, dx2f, 1, vlm_panelm[cnt].v2p, 1);
201         cblas_daxpy(3, (1.0-wzv2)*azv2, dz1, 1, vlm_panelm[cnt].v2p, 1);
202         cblas_daxpy(3, (wzv2)*azv2, dz2, 1, vlm_panelm[cnt].v2p, 1);
203
204         //integracijos tasko apskaiciavimas
205         cblas_daxpy(3, (1.0-wxi), dx10, 1, vlm_panelm[cnt].ip, 1);
206         cblas_daxpy(3, (1.0-wxi)*axi, dx1f, 1, vlm_panelm[cnt].ip, 1);
207         cblas_daxpy(3, (wxi), dx20, 1, vlm_panelm[cnt].ip, 1);
208         cblas_daxpy(3, (wxi)*axi, dx2f, 1, vlm_panelm[cnt].ip, 1);
209         cblas_daxpy(3, (1.0-wzi)*azi, dz1, 1, vlm_panelm[cnt].ip, 1);
210         cblas_daxpy(3, (wzi)*azi, dz2, 1, vlm_panelm[cnt].ip, 1);
211
212         //paneles normales apskaiciavimas
213         double R1[3], R2[3];
214         cblas_dcopy(3, vlm_panelm[cnt].v1p, 1, R1, 1);
215         cblas_daxpy(3, -1., vlm_panelm[cnt].ip, 1, R1, 1);
216         cblas_dcopy(3, vlm_panelm[cnt].v2p, 1, R2, 1);
217         cblas_daxpy(3, -1., vlm_panelm[cnt].ip, 1, R2, 1);
218         cblas_dcopy(3, vcross(R2, R1), 1, vlm_panelm[cnt].normal, 1);
219         cblas_dscal(3, 1./cblas_dnrm2(3, vlm_panelm[cnt].normal, 1), vlm_panelm[
cnt].normal, 1);
220
221         //printf("dx2f %f %f %f\n", dx2f[0], dx2f[1], dx2f[2]);
222         //printf("R1 %f %f %f\n", R1[0], vlm_panelm[cnt].normal[1], vlm_panelm[
cnt].normal[2]);
223
224         vlm_panelm[cnt].A = Af;
225         vlm_panelm[cnt].pb = tbn;
226         vlm_panelm[cnt].k = cblas_dnrm2(3, dz1, 1)/(double)b[tbn].zpan;
227         cblas_dcopy(3, dz1, 1, vlm_panelm[cnt].kv, 1);
228         cblas_dscal(3, 1./(double)b[tbn].zpan, vlm_panelm[cnt].kv, 1);
229         cnt += 1;
230     }
231 }
232 }
233     cnt = vlm_mesh(n, b, vlm_panelm, n[tnn].beams[1], tnn, cnt);
234 }
235 }
236 return cnt;
237 }
238
239 //Funkcija paraleliniam VLM matricos sudarymui atlikti
240 void *thread_vlm(void* args){
241
242     struct vlm_thread_st *inst = (struct vlm_thread_st*) args;
243     struct beam *b = inst->b;
244     struct vlm_panel *vlmp = inst->vlmp;
245     double air_density = inst->air_density;
246     int panel_count = inst->panel_count;
247     int beam_count = inst->beam_count;
248     double *A = inst->A;
249
250     double R1[3], R2[3];
251     double nR1, nR2;
252     double BV[3], HBV[3];
253     double nBV;
254     double cABV, cBBV;
255     double ABV;

```

```

256     double SV1[3], HSV1[3];
257     double cBSV1;
258     double ASV1;
259     double SV2[3], HSV2[3];
260     double cASV2;
261     double ASV2;
262     double CDP;
263     double CCP[3];
264
265     //sukuriu itakos statmenam panelems srautui koeficientu matricos sudarymas
266     for(int i = inst->i0; i < inst->in; i += 1){
267         for(int j = 0; j < panel_count; j += 1){
268             //vektoriai nuo integracijos iki sukuriu prisijungimo
269             cblas_dcopy(3, vtmp[i].ip, 1, R1, 1);
270             cblas_dcopy(3, vtmp[i].ip, 1, R2, 1);
271             cblas_daxpy(3, -1., vtmp[j].v1p, 1, R1, 1);
272             cblas_daxpy(3, -1., vtmp[j].v2p, 1, R2, 1);
273             nR1 = cblas_dnrm2(3, R1, 1);
274             nR2 = cblas_dnrm2(3, R2, 1);
275
276             //prisijunges sukuryms
277             cblas_dcopy(3, vtmp[j].v2p, 1, BV, 1);
278             cblas_daxpy(3, -1., vtmp[j].v1p, 1, BV, 1);
279             nBV = cblas_dnrm2(3, BV, 1);
280             cblas_dcopy(3, R1, 1, HBV, 1);
281             CDP = cblas_ddot(3, R1, 1, BV, 1);
282             cblas_daxpy(3, -CDP/cblas_ddot(3, BV, 1, BV, 1), BV, 1, HBV, 1);
283             cABV = CDP/nBV/nR1;
284             cBBV = cblas_ddot(3, BV, 1, R2, 1)/nBV/nR2;
285             vcross_par(BV, R1, CCP);
286             ABV = (cABV-cBBV)/4/M_PI/cblas_dnrm2(3, HBV, 1)/cblas_dnrm2(3, CCP, 1);
287             ABV = cblas_ddot(3, CCP, 1, vtmp[i].normal, 1)*ABV;
288
289             //pirmasis atitrukes sukuryms
290             cblas_dcopy(3, vtmp[j].v, 1, SV1, 1);
291             cblas_dscal(3, -1/cblas_dnrm2(3, SV1, 1), SV1, 1);
292             cblas_dcopy(3, R1, 1, HSV1, 1);
293             CDP = cblas_ddot(3, R1, 1, SV1, 1);
294             cblas_daxpy(3, -CDP/cblas_ddot(3, SV1, 1, SV1, 1), SV1, 1, HSV1, 1);
295             cBSV1 = CDP/nR1;
296             vcross_par(SV1, R1, CCP);
297             ASV1 = (1-cBSV1)/4/M_PI/cblas_dnrm2(3, HSV1, 1)/cblas_dnrm2(3, CCP, 1);
298             ASV1 = cblas_ddot(3, CCP, 1, vtmp[i].normal, 1)*ASV1;
299
300             //antrasis atitrukes sukuryms
301             cblas_dcopy(3, vtmp[j].v, 1, SV2, 1);
302             cblas_dscal(3, 1/cblas_dnrm2(3, SV2, 1), SV2, 1);
303             cblas_dcopy(3, R2, 1, HSV2, 1);
304             CDP = cblas_ddot(3, R2, 1, SV2, 1);
305             cblas_daxpy(3, -CDP/cblas_ddot(3, SV2, 1, SV2, 1), SV2, 1, HSV2, 1);
306             cASV2 = CDP/nR2;
307             vcross_par(SV2, R2, CCP);
308             ASV2 = (1+cASV2)/4/M_PI/cblas_dnrm2(3, HSV2, 1)/cblas_dnrm2(3, CCP, 1);
309             ASV2 = cblas_ddot(3, CCP, 1, vtmp[i].normal, 1)*ASV2;
310
311             A[panel_count*i+j] = ABV + ASV1 + ASV2;
312         }
313     }
314 }
315
316 //VLM based aero force calculator
317 double * vlm(struct beam *b, double flowv[3], struct vlm_panel *vtmp, double air_density, int
    panel_count, int beam_count, double *A, int *ipiv, double *w, double *G){
318     static double FM[300];
319     static double FM0[300];
320

```

```

321 cblas_dscal(beam_count*6, 0., FM, 1);
322
323 //statmenu panelems srauto komponenciu apskaiciavimas
324 for(int i = 0; i < panel_count; i += 1){
325     double dr[3];
326     cblas_dcopy(3, flowv, 1, vtmp[i].v, 1);
327     cblas_daxpy(3, -1., b[vtmp[i].pb].v, 1, vtmp[i].v, 1);
328     cblas_dcopy(3, vtmp[i].ip, 1, dr, 1);
329     cblas_daxpy(3, -1., b[vtmp[i].pb].pos, 1, dr, 1);
330     cblas_daxpy(3, -1., vcross(b[vtmp[i].pb].omega,dr), 1, vtmp[i].v, 1);
331     vtmp[i].w = cblas_ddot(3, vtmp[i].v, 1, vtmp[i].normal, 1);
332     w[i] = -vtmp[i].w;
333     cblas_dcopy(3, vtmp[i].normal, 1, vtmp[i].wv, 1);
334     cblas_dscal(3, -vtmp[i].w, vtmp[i].wv, 1);
335     //ipiv[i] = i;
336 }
337
338 double R1[3], R2[3];
339 double nR1, nR2;
340 double BV[3], HBV[3];
341 double nBV;
342 double cABV, cBBV;
343 double ABV;
344 double SV1[3], HSV1[3];
345 double cBSV1;
346 double ASV1;
347 double SV2[3], HSV2[3];
348 double cASV2;
349 double ASV2;
350 double CDP;
351
352 int NTHR = 6;
353 pthread_t thread_id[NTHR];
354
355 //structure declaration for passing values to threads, thread creation
356 struct vlm_thread_st thr[NTHR];
357 for(int i = 0; i < NTHR; i += 1){
358     thr[i].b = b;
359     thr[i].vtmp = vtmp;
360     thr[i].air_density = air_density;
361     thr[i].panel_count = panel_count;
362     thr[i].beam_count = beam_count;
363     thr[i].A = A;
364     thr[i].i0 = panel_count/NTHR*i;
365     thr[i].in = panel_count*(i+1)/NTHR;
366     pthread_create(&thread_id[i], NULL, thread_vlm, (void*) &(thr[i]));
367 }
368
369 //paralelines dalies sinchronizacija
370 for(int i = 0; i < NTHR; i += 1){
371     pthread_join(thread_id[i], NULL);
372 }
373
374 //sukuriu stiprumo apskaiciavimas
375 cblas_dcopy(panel_count, w, 1, G, 1);
376 int err = clapack_dgesv(CblasRowMajor, panel_count, 1, A, panel_count, ipiv, G, panel_count);
377 double L[3], R[3], M[3];
378
379 //aerodinamines jegos apskaiciavimas
380 for(int i = 0; i < beam_count; i += 1){
381     int jb = 0;
382     while(vtmp[jb].pb != i && jb < panel_count){
383         jb += b[vtmp[jb].pb].xpan*b[vtmp[jb].pb].zpan;
384     }
385     for(int j = jb; j < jb+b[i].zpan*b[i].xpan; j += b[i].xpan){
386         double Lp = 0.;

```



```

387     double Ap = 0.;
388     for(int k = j; k < j + b[i].xpan; k += 1){
389         Ap += vtmp[k].A;
390         cblas_daxpy(3, 1., vtmp[k].v, 1, vtmp[k].wv, 1);
391         cblas_dscal(3, G[k]*air_density, vtmp[k].wv, 1);
392         cblas_dcopy(3, vcross(vtmp[k].wv, vtmp[k].kv), 1, L, 1);
393         // cblas_dcopy(3, vtmp[k].normal, 1, L, 1);
394         // cblas_dscal(3, G[k]*cblas_dnrm2(3, vtmp[k].v, 1)*air_density*vtmp[k].k*2, L, 1);
395         cblas_dcopy(3, vtmp[k].ip, 1, R, 1);
396         cblas_daxpy(3, -1., b[i].pos, 1, R, 1);
397         cblas_dcopy(3, vcross(R, L), 1, M, 1);
398         Lp += cblas_ddot(3, L, 1, vtmp[k].normal, 1);
399         FM[i*6] += L[0];
400         FM[i*6+1] += L[1];
401         FM[i*6+2] += L[2];
402         FM[i*6+3] -= M[0];
403         FM[i*6+4] -= M[1];
404         FM[i*6+5] -= M[2];
405     }
406     double CLp = Lp/cblas_dnrm2(3, vtmp[j].v, 1)/cblas_dnrm2(3, vtmp[j].v, 1)*2./ (air_density*Ap
);
407     int k = 0;
408     while(prof_cl[k] < CLp && k < 12){
409         k += 1;
410     }
411     double CDp = interp(CLp, prof_cl[k-1], prof_cd[k-1], prof_cl[k], prof_cd[k]);
412
413     //printf("CLp %f    clmax %f    clmin %f\n", CLp, clmax[i] - b[i].fa*flapclslp[i], clmin
[i] - b[i].fa*flapclslp[i]);
414     //printf("coeff %f\n", (clmax[i]-b[i].fa*flapclslp[i])/CLp);
415
416     if(CLp > clmax[i] - b[i].fa*flapclslp[i]){
417         //cblas_dscal(3, (clmax[i]-b[i].fa*flapclslp[i])/CLp, FM+i*6, 1);
418     }
419     else if(CLp < clmin[i] - b[i].fa*flapclslp[i]){
420         //cblas_dscal(3, (clmin[i]-b[i].fa*flapclslp[i])/CLp, FM+i*6, 1);
421     }
422
423     cblas_dcopy(3, vtmp[j].v, 1, L, 1);
424     cblas_dscal(3, cblas_dnrm2(3, vtmp[j].v, 1)*Ap*air_density/2.*CDp, L, 1);
425     FM[i*6] += L[0];
426     FM[i*6+1] += L[1];
427     FM[i*6+2] += L[2];
428 }
429 //printf("F %f %f %f %f %f %f \n", FM[i*6], FM[i*6+1], FM[i*6+2], FM[i*6+3], FM[i*6+4], FM[i
*6+5]);
430 }
431
432 //AERODINAMINIS VELAVIMAS
433 double vc[3];
434 double vcn, tc, aexp;
435 double c = 200.;
436 for(int i = 0; i < beam_count; i++){
437     cblas_dcopy(3, flowv, 1, vc, 1);
438     cblas_daxpy(3, -1., b[i].v, 1, vc, 1);
439     vcn = cblas_dnrm2(3, vc, 1);
440     tc = c/vcn*aero_lag;
441     //printf("tc %f    ", tc);
442     for(int j = 0; j < 6; j++){
443         aexp = (FM[i*6+j]-FM0[i*6+j])*dt/tc;
444         if(fabs(aexp) < fabs(FM[i*6+j]-FM0[i*6+j])){
445             FM[i*6+j] = FM0[i*6+j] + aexp;
446         }
447         //printf("exp %f    ", aexp);
448     }
449     //printf("\n");

```

```
450     }
451
452     cblas_dcopy(beam_count*6, FM, 1, FM0, 1);
453
454     return FM;
455 }
```

## 2 priedas. Konferencijos „Transport Means 2021“ straipsnis

*Proceedings of 25<sup>th</sup> International Scientific Conference. Transport Means 2021.*

### Development of Aeroservoelastic Analysis Method for High Flexibility Aircraft

**M. Dagilis**

*Kaunas University of Technology, Studentų 56, 51424, Kaunas, Lithuania, E-mail: [m.dagilis@ktu.edu](mailto:m.dagilis@ktu.edu)*

#### Abstract

Aeroservoelasticity modelling is increasingly important in modern aviation. This paper describes a computational aeroservoelastic model, design to model the flight of high flexibility aircraft. The aeroservoelastic model is created by combining the vortex lattice method, a lumped mass structural model and a traditional kinematic model. A flight control algorithm is developed to control the modelled aircraft. Two flight scenarios are calculated and analysed to showcase the capabilities of the developed model.

**KEY WORDS:** *aeroservoelasticity; vortex lattice method; computational aerodynamics*

#### 1. Introduction

There is a trend of increasing use of composite materials in aircraft construction [1]. Composite materials usually have excellent strength-mass ratios, but their stiffness-mass ratios are more similar to traditional structural materials. This results in structures that have low mass, but high flexibility, which causes a decrease in flutter speed and controllability of the aircraft [2]. To partially alleviate these adverse effects, active flutter damping [3, 4] and flight control algorithms [5] can be used. To develop such algorithms good aeroservoelastic flight models are required, taking into account the aerodynamic forces, the structural deformations and the control systems of the aircraft. Such models require both high fidelity and a fast computational speed (to make optimisation possible), making computationally expensive CFD based methods and less precise traditional analytical methods less suitable. In this work a computational aeroservoelastic flight model which runs faster than a CFD based model and is more precise than analytical methods will be designed.

#### 2. Physical Aeroservoelastic Model

##### 2.1. General Description

The developed model uses the vortex lattice method for aerodynamic calculations, a structural model based on angular springs and dampeners and motion equations derived from a classical kinematic model. These methods were chosen because they provide sufficient precision for the intended purpose and are computationally inexpensive.

The modelled aircraft is divided into a low number of rigid elements (For the aircraft used to test the model, 8 elements were used), to make the model less computationally expensive. The relations between the elements form a tree graph, with one root element and multiple branches. The connections between the elements are rigid in linear motion and damped-elastic in rotational motion. Each element is divided into panels for aerodynamic calculations.

##### 2.2. Kinematic Model

A traditional kinematic method is used to calculate the motion of the elements. The linear and angular accelerations of each element are determined by solving the force equations. The acceleration is then integrated to calculate the velocity, and the velocity is integrated to calculate the new position of the elements [6].

Because there are imprecisions in the discrete computational implementation of this model, the positional integration errors accumulate over time. To avoid this, a positional correction is performed during calculations. The correction is performed on the linear position of the elements based on their rotational position. The global location of each element is recalculated based on the rotation of the element and the rotation and global location of its parent element.

##### 2.3. Structural Model

The elements of the aircraft are considered to be perfectly rigid. They are connected with angular damped springs, described by a stiffness matrix and a damping matrix. The connections between elements are perfectly rigid in translation. This model is useful for calculating bending and torsion deformations, but neglects tension and shear deformations. Since bending and torsion are dominant in aircraft, this drawback is considered acceptable.

The stiffness moment acting on each element is calculated by equation [7]

$$M_{k,i} = -\sum_{j=1}^n (K_{ij} \times \Delta\theta_{ij}), \quad (1)$$

where  $M_{k,i}$  – stiffness moment acting on the  $i$ -th element;  $n$  – number of elements connected to the  $i$ -th element;  $K_{ij}$  – the stiffness matrix of the connection between  $i$ -th and  $j$ -th elements;  $\Delta\theta_{ij}$  – difference of angular positions between  $i$ -th and  $j$ -th elements.

The damping moment acting on each element is calculated by equation[7]

$$M_{c,i} = -\sum_{j=1}^n (C_{ij} \times \Delta\omega_{ij}), \quad (2)$$

where  $M_{c,i}$  – damping moment acting on the  $i$ -th element;  $n$  – number of elements connected to the  $i$ -th element;  $C_{ij}$  – the damping matrix of the connection between  $i$ -th and  $j$ -th elements;  $\Delta\omega_{ij}$  – difference of angular velocities between  $i$ -th and  $j$ -th elements.

## 2.4. Aerodynamic Model

Aerodynamic forces acting upon the aircraft are calculated using the vortex lattice method (VLM)[8]. This method is based on dividing the aerodynamic surfaces of an aircraft into panels. In this specific implementation, each element representing a section of an aerodynamic surface is divided into panels. Flaps and control surfaces are implemented by one or more rows of panels along the trailing edge of the aerodynamic surfaces, which position changes as the control surface is actuated.

The basis of VLM is the assumption that the wing vortex can be divided into many small vortices (one vortex for each wing panel). Each vortex is made up of three parts – the bound vortex, matching the quarter chord line of the panel; and two trailing vortices, going from the ends of the bound vortex to infinity, parallel to the free stream. Each panel also has an integration point (usually at the centre of the three-quarter chord line), at this point the flow velocity induced by the vortices is calculated.

The main task when applying VLM is to compute the strength of these panel vortices (which can then be used to calculate lift). It is calculated from equations [8]

$$w_i + V \sin(\alpha_i) = 0, \text{ and} \quad (3)$$

$$w_i = \sum_{j=1}^N A_{ij} \Gamma_j, \quad (4)$$

where  $w_i$  – component of total velocity induced by the vortices at the  $i$ -th panel, perpendicular to the panel;  $V$  – free flow velocity;  $\alpha_i$  – angle of attack of the  $i$ -th panel;  $N$  – total number of panels;  $A_{ij}$  – coefficient describing the linear relation between  $w_i$  and  $\Gamma_j$ ;  $\Gamma_j$  – the strength of the vortex of the  $j$ -th panel. Because  $V$  and  $\alpha_i$  are known, Eq. (3) gives the value of  $w_i$ . Each  $A_{ij}$  can be calculated using the Biot-Savart law, then a system of linear equations based on Eq. (4) can be solved to calculate each  $\Gamma_j$ . These can then be used to calculate lift at each panel by using the Kutta-Joukowski theorem [9]

$$L_i = 2\rho V \Gamma_i k_i, \quad (5)$$

where  $L_i$  – lift force acting on the  $i$ -th panel;  $\rho$  – air density;  $k_i$  – width of the  $i$ -th panel.

Because basic VLM is an inviscid method, it greatly underestimates the drag force of an aerodynamic surface. To correct for this, a drag coefficient is interpolated from aerofoil aerodynamic data based on the local coefficient of lift at each wing section and added to the force acting on the aerodynamic surface. The aerodynamic forces acting on non-aerodynamic surfaces (e.g. the fuselage) are ignored in the calculations.

## 3. Computational Aeroservoelastic Model

### 3.1. General Description

The aeroservoelastic model described in section 2 was implemented computationally using the C programming language. The elements and nodes of the aircraft are described in a JSON format file, which is read by the program. After that, the program performs calculations based on the developed model, while the motion and deformation of the aircraft are shown live on screen, while also being written to an output file. For live rendering, the SDL2 graphical library [10] is used.

Linear vector and matrix operations are performed using the BLAS library [11] and custom-written functions, and linear system solving (for calculating aerodynamic forces and aircraft element accelerations) is performed using the

LAPACK library [12]. The custom VLM calculating function was parallelized after it was identified as a bottleneck in the performance of the program. While analysing the aircraft described in subsection 4.1. (structure divided into 8 elements), the program calculates 400 discrete timesteps per second when running on three processor cores.

### 3.2. Flight Control Algorithm

Because the created program runs slower than real-time, controlling the aircraft with live inputs from a pilot is not possible. To control the aircraft, a flight control algorithm was written, actuating the control surfaces based on flight data. The algorithm is of a proportional-integral-derivative (PID) type, meaning that each deviation from a desired value is differentiated and integrated, and these values multiplied by a coefficient are used to set the control variables. The algorithm uses the current altitude, velocity vector and angular position of the aircraft as data inputs, and required velocity vector and altitude as objective inputs. It tries to achieve the flight described by the objective inputs by controlling the aerodynamic control surfaces and thrust of the aircraft.

## 4. Study Cases of the Developed Model

### 4.1. Aircraft Model Used for Calculations

The developed model was used to calculate two different flight scenarios. The aircraft model used for these scenarios is shown in Fig. 1 with element and node numbers, and each element is described in Table 1.

Table 1

Elements of modelled aircraft

Element number	Element mass, g	Element aerodynamic span, mm	Element aerodynamic chord, mm	Element control surface	Parent element
0	1000	0	0	-	-
1	250	750	200	-	0
2	150	750	200	Aileron	1
3	250	750	200	-	0
4	150	750	200	Aileron	3
5	25	250	150	Elevator	0
6	25	250	150	Elevator	0
7	30	300	150	Rudder	0

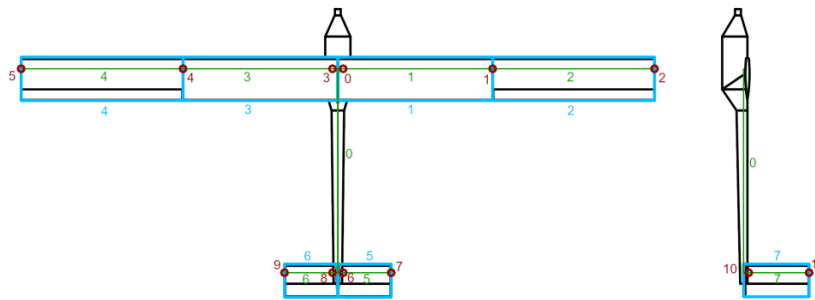


Fig. 1 Elements and nodes of the aircraft used for calculations

The mass of the aircraft is 1,88 kg. The wingspan of the aircraft is 3000 mm, the chord length of the wing is 200 mm. The chord length of the tail surfaces is 150 mm, the span of the horizontal stabilizer is 500 mm, the height of the vertical stabilizer is 300 mm. For the main wing and tail surfaces of the aircraft the NACA 0010 aerofoil was used. The outer 750 mm of each wing is an aileron, taking up 25% of the wing chord. The rudder and elevator both take up 40% of the chord of the tail surfaces and span their whole length. All control surfaces are limited to 30° deflection in both directions. The maximum thrust of the aircraft is set to be 10 N.

### 4.2. Two Turns Scenario

The first tested scenario is flight with two turns. It is intended to test the capability of the flight control algorithm to reach and hold a required heading. It can be described by the following sequence of actions:

1. the flight starts straight and level at a velocity of 9,5 m/s;
2. the required heading is set to 45° right;
3. after 4 s, the required heading is set to 45° left (a 90° left turn).

During the flight, a constant required flight speed of 9,6 m/s and altitude was set. The results of the calculations are presented in Fig. 2 and Fig. 3. In Fig. 2 the trajectory of the aircraft viewed from above is shown. The aircraft starts a right turn immediately when the scenario starts, and at 2 s the turn is mostly finished. At 4 s, when the right turn is fully finished, the aircraft starts a left turn. At 8 s, the left turn is finished, and the aircraft continues to fly straight.

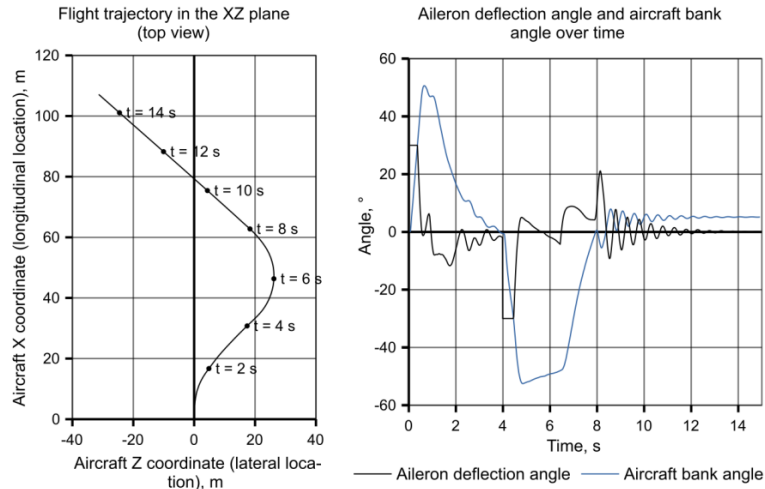


Fig. 2 Trajectory of the aircraft during the two turns flight scenario

Fig. 3 Aileron deflection angle and aircraft bank angle over time during the two turns flight scenario

In Fig. 3 it can be seen that the aircraft starts the flight with ailerons fully deflected to perform the first turn. The aircraft goes into a bank of  $50,64^\circ$  (the algorithm is limited to  $50^\circ$  banks, but can overshoot because of rolling inertia) and returns to level flight as it approaches the required heading. At 4 s as the second turn starts, the ailerons are deflected fully in the opposite direction. The aircraft enters a bank of  $52,52^\circ$  and holds it until 6,5 s, returning to level flight as it approaches the new heading. There is an oscillation as the aircraft exits the manoeuvre, caused by rolling inertia and wing flexing, but it is damped by the control algorithm. After the manoeuvre the aircraft stays in a  $5^\circ$  right bank, attempting to correct its course (in Fig. 2 it is shown that the aircraft is heading slightly more left than the required  $45^\circ$  heading).

### 4.3. Pitch Up Scenario

The second flight scenario was a pitch up manoeuvre. It is intended to test the bending of the wings under aerodynamic loading. It can be described by the following sequence of actions:

1. the flight starts straight and level at a velocity of 10 m/s;
2. after two seconds elevator is set to a fixed  $20^\circ$  angle;
3. while pitching up, the rudder and ailerons are used to hold the heading straight and the wings level.

In Fig. 4 and Fig. 5 the results of this calculation are presented. Fig. 4 shows a side-view trajectory of the aircraft. There is an initial dip in altitude because the aircraft starts at zero angle-of-attack. By 1 s, it is corrected. The aircraft then continues to fly straight until 2 s, when it starts pitching up. The radius of the trajectory arc is decreasing as the velocity of the aircraft decreases. By 5 s the aircraft is past the vertical and partially inverted.

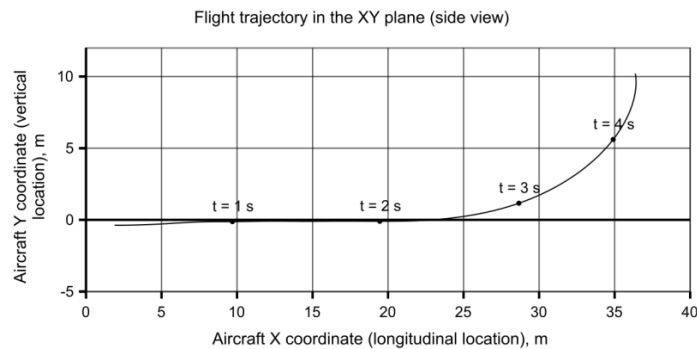


Fig. 4 Trajectory of the aircraft during the pitch up flight scenario

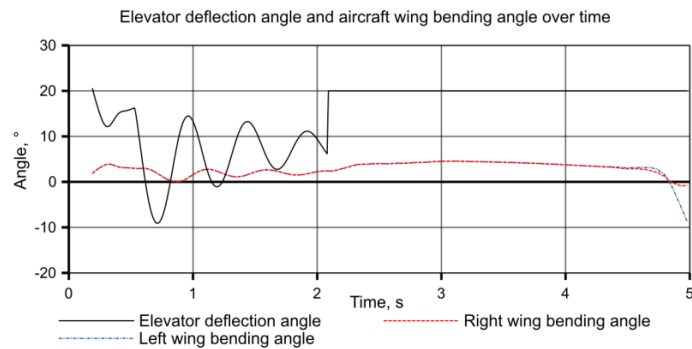


Fig. 5 Elevator deflection angle and aircraft wing bending angle over time during the pitch up flight scenario

Fig. 5 Shows the elevator deflection angle and aircraft wing bending angle changing over time. Over the first two seconds of the flight, the wings and elevator are in a damped oscillation as the aircraft returns to level flight from the initial dip. At 2 s, the elevator is set to a constant  $20^\circ$  angle, the deflection of the wings start increasing until it reaches  $4,57^\circ$  at about 3 s, then starts decreasing as the velocity of the aircraft and the aerodynamic loads start decreasing. Because the wings are kept level and the aerodynamic loads are symmetrical, both wings are bent at the same angle, until the plane becomes inverted. After that, the flight algorithm tries to roll the plane back up using ailerons, which causes asymmetrical loading and bending of the wings.

### 3. Conclusions

A physical aeroservoelastic model for analysing the flight characteristics of high flexibility aircraft was developed by combining the vortex lattice method, a lumped mass structural model and a traditional kinematic model.

The developed physical model was successfully implemented computationally using the C programming language. The program calculates 400 discrete timesteps per second when running on three processors and analyses an aircraft with an 8-element structure.

A flight control algorithm was created to control the aircraft during the calculations. The algorithm is of the proportional-integral-differential type.

Two test scenarios were successfully calculated using the developed program. It was shown that the flight algorithm can reach and hold a required heading and that the structure of the aircraft deforms under aerodynamic loading as expected.

### Acknowledgement

This research was funded by a grant (project No. 09.3.3-LMT-K-712-24-0020) from European Social Fund.

### References

1. Slayton, R.; Spinardi, G. 2015. Radical innovation in scaling up: Boeing's Dreamliner and the challenge of socio-technical transitions, *Technovation* 47: 47-58.
2. Swaim, R.L.; Yen, W.Y. 1978. Effects of dynamic aeroelasticity on handling qualities and pilot rating, 4th Atmospheric Flight Mechanics Conference, Palo Alto 266-271.
3. Brillante, C.; Mannarino, A. 2016. Improvement of aeroelastic vehicles performance through recurrent neural network controllers, *Nonlinear Dynamics* 84(3):1479-1495.
4. Zhang, K.; Marzocca, P.; Behal, A. 2015. Adaptive aeroelastic control of nonlinear airfoil with multiple flaps under unsteady flow, *Journal of vibration and control* 23(10):1593-1606.
5. Livet, T.; Kubica, F.; Magni, J.F. 1995. Robust Flight Control Design With Respect to Delays and Control Efficiencies, *Control Engineering Practice* 3(10):1373-1384.
6. Beggs, J.S. 1983. *Kinematics*, New York: Hemisphere Publishing Corporation.
7. Fedaravičius, A.; Jurkauskienė, E.; Mikučionienė, J.; Rudgalvienė N. 2009. *Teorinė mechanika: teorijos pagrindai ir uždaviniai*. Kaunas: Technologija (in Lithuanian).
8. Cummings, R.M.; Hason, W.H.; Morton, S.A.; McDaniel, D.R. 2015. *Applied Computational Aerodynamics: A Modern Engineering Approach* (Cambridge Aerospace Series), Cambridge: Cambridge University Press.
9. Anderson, J. 2017. *Fundamentals of Aerodynamics* (6th edition), New York: McGraw-Hill Education, 275 p.
10. Simple DirectMedia Layer [online cit: 2021-07-20]. Available from: <https://www.libsdl.org/>
11. BLAS (Basic Linear Algebra Subprograms) [online cit: 2021-07-16]. Available from: <https://www.netlib.org/blas/>
12. LAPACK—Linear Algebra PACKage [online cit: 2021-07-16]. Available from: <https://www.netlib.org/lapack/>