

MAANPUOLUSTUSKORKEAKOULU

TEKOÄLY HÄVITTÄJÄN OHJAAJANA KAARTOTAISTELUSSA

Pro gradu -tutkielma

Yliluutnantti
Mikko Kytö

SMOHJ16
Ilmasotalinja

Maaliskuu 2022

Kurssi SMOHJ16	Linja Ilmasotalinja
Tekijä Yliluutnantti Mikko Kytö	
Opinnäytetyön nimi Tekoäly hävittäjän ohjaajana kaartotaistelussa	
Oppiaine, johon työ liittyy Sotatekniikka	Säilytyspaikka Maanpuolustuskorkeakoulun kirjasto
Aika Maaliskuu 2022	Tekstisivuja 61 Liitesivuja 13
<p>TIIVISTELMÄ</p> <p>Tekoälyn alaan kuuluvat syväoppiminen ja koneoppiminen sekä niiden menetelmiin kuuluvat keinotekoiset neuroverkot ja evoluutioalgoritmit ovat yleisesti käytettyjä kuluttajasovelluksissa. Tekoälyä hyödyntävät sotilaalliset sovellukset tulevat yleistymään yhä kiihtyvällä tahdilla, minkä vuoksi tekoälytutkimukseen panostetaan Puolustusvoimissa. Tämän tutkimuksen tarkoituksena oli selvittää, mitä neuroverkot ja evoluutioalgoritmit ovat ja miten niitä voidaan hyödyntää. Tekoälylle valittiin tehtäväksi ilmassa hävittäjillä tapahtuva kaartotaistelu, sillä se on luonteeltaan selkeä mutta haastava optimointiongelma. Tutkimuksessa todennettiin kokeellisesti edellä mainittuihin menetelmiin perustuvan tekoälyn käytettävyys kaartotaistelussa hävittäjäohjaajan korvaajana. Tutkimuksen tavoitteet olivat teoreettiset, eli valmiiseen sovellutukseen ei pyritty. Sen sijaan tavoite oli todentaa menetelmien toimivuus konseptitasolla.</p> <p>Koeasetelmaa varten luotiin matemaattinen kuuden vapausasteen (6-DOF) -malli F-16-hävittäjästä, ja mallia hyödyntämällä simuloitiin hävittäjien välistä kaartotaistelua eräajo-ohjelmalla. Simuloiduissa kaartotaisteluissa kaksi identtistä F-16-hävittäjää aloittivat kaartotaistelun tasavertaisesta lähtöasetelmasta tavoitteena päästä hallinta-asemaan vastustajan taakse hävittäjän keula kohti vastustajaa. Hävittäjien ohjaaminen annettiin tehtäväksi neuroverkkopohjaiselle tekoälypopulaatiolle, jota kehitettiin evoluutioalgoritmeihin kuuluvalla geneettisellä algoritmilla simulaatiota iteroimalla.</p> <p>Kokeen tuloksia analysoitiin kvantitatiivisesti. Geneettinen algoritmi antoi kullekin tekoälyn suoritukselle kelpoisuusarvon kelpoisuusfunktion perusteella. Kelpoisuusarvon kehitystä analysoimalla voitiin päätellä, kykeneekö kokeen tekoäly oppimaan eli kehittymään paremmaksi kaartotaistelusimulaatiossa iteraatioiden myötä. Lisäksi tekoälyn suoritustasoa analysoitiin kvalitatiivisesti visuaaliseen tarkasteluun perustuen asiantuntijoiden lausuntojen perusteella. Ilmavoimien hävittäjälentolaivue 11:n ilmataisteluolettajista kootulle asiantuntijaryhmälle näytettiin visuaalisesti simuloituja kaartotaistelukohtaamisia tekoälyn koulutuksen eri vaiheista. Asiantuntijoiden tehtävä oli antaa suorituksista numeerinen sekä sanallinen palaute samaan tapaan kuin he antaisivat sen oikealle ilmataisteluoppilaalle. Asiantuntijoiden numeerisen palautteen tunnuslukuja verrattiin eräjön sukupolvien määrään. Näiden kahden muuttujan välistä korrelaatiota analysoimalla tehtiin johtopäätöksiä tekoälyn oppimisesta.</p> <p>Kokeen eräajojen aikana suoritettiin simulaatioita 3 974 sukupolven verran, eli yhteensä 158 960 kaartotaistelukohtaamista, populaation koon ollessa 80 neuroverkkoyksilöä. Kokeen tuloksia analysoimalla havaittiin selkeä, syväoppimiselle tyypillinen, logaritmista funktiota mukaileva nouseva trendi sukupolvien parhaiden yksilöiden kelpoisuudessa sekä sukupolvien populaatioiden kelpoisuuden keskiarvossa. Kelpoisuutta kuvaavien kehityskäyrien kulmakertoimia ja erotusta analysoimalla voitiin todeta tekoälyllä olevan vielä kehittymispotentiaalia, mikäli iteraatioita jatketaan. Asiantuntijoiden lausunnot tukevat kvantitatiiviseen analyysiin perustuvaa näkemystä tekoälyn kehittymisestä. Tekoäly ei kuitenkaan saavuttanut koulutetun hävittäjäohjaajan suoritustasoa kaartotaistelusimulaatiossa.</p>	
<p>AVAINSANAT simulointi, mallintaminen, tekoäly, neuroverkot, geneettiset algoritmit, hävittäjät, ilmasota, ilmataistelut</p>	

SISÄLLYS

1.	JOHDANTO.....	1
1.1.	Tutkimustilanne.....	2
1.2.	Tutkimustehtävä ja tutkimuskysymykset.....	4
1.3.	Tutkimusmenetelmät	4
1.4.	Näkökulma ja rajaukset	5
1.5.	Lähdemateriaalin esittely ja lähdekritiikki	6
1.6.	Määritelmät	7
2.	LENTOMALLIN JA TEKOÄLYN TEORIA.....	9
2.1.	F-16 6-DOF -simulaatio.....	9
2.1.1.	Liiketyhtälöt	10
2.1.2.	Ilmakehämalli	13
2.1.3.	Moottorimalli.....	14
2.1.4.	F-16-lentomalli.....	15
2.1.5.	Iterointi Runge–Kutta-menetelmällä	21
2.2.	Tekoälystä koneoppimiseen ja syväoppimiseen	23
2.2.1.	Syväoppiminen ja Neuroverkot	28
2.3.	Evoluutioalgoritmi	31
2.3.1.	Geneettinen algoritmi.....	34
2.3.1.1.	Alustus.....	35
2.3.1.2.	Valinta	35
2.3.1.3.	Risteytys.....	36
2.3.1.4.	Mutaatio	37
3.	ERÄAJOSIMULAATION JA TEKOÄLYN PARAMETRIT	38
3.1.	Erääjot	38
3.1.1.	Erääjön parametrit.....	39
3.1.2.	Simulaation parametrit	40
3.1.3.	Neuroverkon parametrit.....	41
3.1.4.	Geneettisen algoritmin parametrit	44
4.	ERÄAJOSIMULAATIOIDEN TULOKSET	47
4.1.	Erääjön tulosten esittely ja analyysi	47
4.2.	Asiantuntija-arviot	54
5.	POHDINTAA	57
5.1.	Jatkotutkimustarpeet.....	59
6.	JOHTOPÄÄTÖKSET.....	60

LÄHTEET

LIITTEET

LIITE 1: F-16-lentomallin matemaattiset kaavat

LIITE 2: Taulukot F-16-hävittäjän aerodynaamisten voimien kertoimista

LIITE 3: Asiantuntija-arviot

LIITE 4: Lähdekoodi ja eräajon tulokset

TEKOÄLY HÄVITTÄJÄN OHJAAJANA KAARTOTAISTELUSSA

1. JOHDANTO

Tutkimuksessa sovelletaan neuroverkkopopulaatiota ja evoluutioalgoritmia kaartotaistelusimulaatiossa ja luodaan tekoäly, joka kykenee taistelemaan näköetäisyyden sisäpuolella toista identtistä hävittäjää vastaan. Tekoälyn suorituskykyä arvioidaan määrällisesti dynaamisen pisteityksen avulla, sekä laadullisesti asiantuntijalausuntojen perusteella.

Tutkimus toteutetaan kokeellisesti Python-ohjelmointikielellä rakennetulla eräajosimulaatio-ohjelmalla, joka pitää sisällään kuuden vapausasteen (6-DOF) -fysiikkamallin General Dynamicsin F-16-monitoimihävittäjästä ja syväoppivan algoritmin. Syväoppiva algoritmi koostuu neuroverkkopopulaatiosta, jota kehitetään iteratiivisesti evoluutioalgoritmeihin kuuluvalla geneettisellä algoritmilla.

Tutkielman ensimmäisessä luvussa määritellään tutkimuksen lähtökohdat, joihin kuuluvat tutkimustilanne, tutkimusongelma ja menetelmät. Lisäksi määritellään tutkimuksessa käytetyt käsitteet, näkökulma ja rajaukset. Luvun lopussa luodaan yleiskatsaus tärkeimpiin tutkimuksessa käytettyihin lähteisiin ja arvioidaan niiden luotettavuutta.

Toisessa luvussa määritellään tutkimuksen teoreettinen viitekehys. Luku pitää sisällään tarkan kuvauksen simulaatiosta ja F-16-mallista. Tutkimuksessa käytetyt koneoppimisen ja syväoppimisen menetelmät esitellään luvussa yleisellä tasolla. Lisäksi luvussa kerrotaan miten teoreettisista malleista, esitettyjä menetelmiä hyödyntämällä, luodaan tutkimuksen koasetelma.

Kolmannessa luvussa esitellään koetta varten kirjoitettu eräajosimulaatio-ohjelma yksityiskohdallisesti. Luku pitää sisällään kuvauksen ohjelman rakenteesta, sekä siinä käytettävistä menetelmistä ja parametreista edellisen luvun teoreettiseen viitekehukseen tukeutuen. Ohjelman lähdekoodi on saatavissa kokonaisuudessaan liitteenä 4. Lähdekoodi on liitetty tutkielmaan, jotta jatkotutkimusta tekevä tutkija voi lukujen 2 ja 3 avulla käyttää olemassa olevaa koodia jatko-tutkimuksessa.

Neljännessä luvussa esitetään tutkimuksen tulokset ja niitä analysoidaan laadullisin sekä määrällisin menetelmin. Tekoälyn kouluttamista varten syväoppivaan algoritmiin on kirjoitettu kelpoisuusfunktio, jonka tarkoitus on kuvata tekoälylle mitä sen tulisi tehdä. Kelpoisuusfunktio antaa tulokseksi tekoälyn kelpoisuusarvon numeerisena arvona, joten sitä voidaan käyttää tekoälyn suoritustason arvioimisessa kvantitatiivisesti eli määrällisesti. Kvalitatiivisesti eli laadullisesti tekoälyn toimintaa arvioidaan asiantuntijoiden toimesta, jotka antavat visuaalisen tarkastelun perusteella lausuntonsa tekoälyn suoritustasosta.

Ennen johtopäätöksiä pohdintaluvussa tutkija tarkastelee havaintojaan tutkimuksen aikana esille nousseista teemoista, sekä kokeen toteuttamisen aikana havaituista haasteista. Tutkimuksen tulosten merkitystä tarkastellaan kriittisesti, minkä lisäksi luvussa esitetään mahdollisia jatkotutkimusaiheita.

Viimeisessä luvussa esitetään tutkimuksen tuloksien perusteella tehdyt johtopäätökset tukeutuen aikaisempiin lukuihin. Johtopäätökset antavat vastauksen luvussa 1.2 esitettyyn tutkimusongelmaan.

1.1. Tutkimustilanne

Ilmataistelusimulaation tulosten analysointia tilastotieteen ja koneoppimisen keinoin ovat Suomessa tutkineet muun muassa Jirka Poropudas ja Kai Virtanen. Tutkimuksessaan Poropudas ja Virtanen ovat sovittaneet ilmataistelusimulaation datan dynaamiseen Bayesialaiseen verkkoon. Tuloksena on todennäköisyyksiin perustuva malli, joka kuvaa tilanteen kehitystä ilmaistelussa ja mahdollistaa erilaisten ilmataistelutilanteiden tarkastelun ja tutkimisen. [1]

Ilma-aluksen optimaalisen lentoradan määrittämistä epälineaarisen ohjelmoinnin keinoin ovat tutkineet T. Raivio, H. Ehtamo ja R.P. Hämäläinen raportissaan ”*Aircraft trajectory optimization using nonlinear programming*”. Tutkijoiden esittämällä menetelmällä pyritään löytämään optimaalinen lentorata määritetystä aloitustilanteesta haluttuun lopputilanteeseen. [2]

Lisäksi Janne Karellahti, Kai Virtanen ja Tuomas Raivo ovat tutkineet keskikantaman aktiivisen tutkaohjuksen optimaalisen päivitysajan etsimistä peliteorian keinoin. Tutkimuksen tilanteessa kaksi identtistä hävittäjää pyrkii tuhoamaan toisensa pysyen samalla itse hengissä. Taistelutilanne on jaettu kolmeen vaiheeseen. Ensimmäisessä vaiheessa laukaiseva hävittäjä lähettää vastustajan paikkatietoa ohjukselle. Toisessa vaiheessa hävittäjä pyrkii itse väistämään vastustajan ohjusta ja oma ohjus hakeutuu ilman paikkatiedon päivitystä ekstrapoloituun kohtaamispisteeseen vastustajaan nähden. Viimeisessä vaiheessa ohjus avaa oman tutkansa ja pyrkii lukitsemaan hakupäänsä vastustajan hävittäjään. Hävittäjän ohjaajalla on mahdollisuus valita päivitysvaiheen pituus vapaasti. Pidempi päivitysvaihe lyhentää ekstrapolointivaihetta, mikä puolestaan nostaa todennäköisyyttä, jolla ohjus oman tutkan avatessaan kykenee lukittumaan vastustajaan. Pidempi päivitysvaihe nostaa toisaalta riskiä, sillä se vaati lentämistä kohti vastustajan ohjusta. Todennäköisyydet riippuvat tietenkin myös vastustajan toimista. Tutkimusongelmana on maksimoida todennäköisyydet vastustajan tuhoutumiselle ja omalle selviytymiselle tilanteessa, jossa myös vastustajan oletetaan toimivan rationaalisesti. [3]

Suurimmat kehitysaskeleet tämän tutkimuksen viitekehyksessä ovat viime vuosina tapahtuneet kaupallisella puolella sotilasteknologia-alan yritysten toimesta. Vuosina 2019–2020 DARPA järjesti avoimen kilpailun, jossa osallistujien tehtävänä oli luoda tekoäly, joka kykenee tehokkaaseen kaartotaisteluun F-16-hävittäjällä DARPA:n kehittämässä ACE-simulaatiossa. Kilpailuun ottivat osaa mm. suuret asevoimien yritykset, yliopistot sekä tekoälytyöpajat. Kilpailun viimeisessä vaiheessa kahdeksasta osallistujasta valittiin paras tekoäly, joka voitti kilpailun finaalissa kokeneen hävittäjälentäjän. Finaalissa sekä ihmisen ohjaaman, että tekoälyn ohjaaman hävittäjän tehtävänä oli osoittaa lentokoneen asekselillä vastustajaa kumulatiivisesti vähintään kolmen sekunnin ajan, mikä kuvaa tähdätyn tykkiammunnan suorittamista vastustajaa vastaan. [4]

Kaartotaistelu on luonteeltaan haastava ja dynaaminen optimointiongelma. Optimaalinen liikehtelygeometria riippuu aina vastustajan liikehtelystä, minkä vuoksi optimaalista liikehtelygeometriaa ei voida laskea etukäteen, vaan se on laskettava reaaliajassa. DARPA:n järjestämä kilpailu ja sen tulokset ovat osoitus nykyisen teknologian ja ohjelmointimenetelmien kyvystä saavuttaa ja jopa ylittää inhimillinen suoritustaso kaartotaistelussa. Kilpailu oli luonteeltaan PR-tempaus, minkä lisäksi tarkat raportit osallistujien tekoälyohjelmista ovat valtaosin yritysalaisuuksia. Tämän vuoksi kilpailun tulosten tieteellinen luotettavuus on kyseenalaista. Tämän tutkimuksen tarkoitus on toistaa DARPA:n kilpailussa nähdyt havainnot tarkasti raportoiden ja dokumentoiden.

1.2. Tutkimustehtävä ja tutkimuskysymykset

Neuroverkot ja evoluutioalgoritmit ovat vakiintuneita menetelmiä tekoälytutkimuksen alalla ja kyseisiä syväoppivia sekä koneoppivia algoritmeja hyödyntävät sotatekniset sovellukset tulevat yleistymään tulevaisuudessa, minkä vuoksi tekoälytutkimukseen panostetaan Puolustusvoimissa. Tässä tutkimuksessa tavoitteena on selvittää yksityiskohtaisesti, miten syväoppimisen ja koneoppimisen menetelmät toimivat ja todentaa niiden hyödynnettävyys simuloitussa ilmaististelussa.

Tutkimustehtävänä on selvittää, pystytäänkö neuroverkko kouluttamaan evoluutioalgoritmia hyödyntäen ohjaamaan hävittäjää kaartotaistelusimulaatiossa. Tutkimustehtävästä voidaan johtaa seuraava päätutkimuskysymys:

- Voidaanko hävittäjäohjaaja korvata neuroverkkoon ja evoluutioalgoritmiin perustuvalla tekoälyllä kaartotaistelusimulaatiossa?

Päätutkimuskysymyksestä voidaan johtaa seuraavat alakysymykset, joihin vastaamalla saadaan vastaus päätutkimuskysymykseen:

- Mitä ovat neuroverkko ja evoluutioalgoritmi ja miten ne toimivat?
- Miten neuroverkkoon ja evoluutioalgoritmiin perustuvaa tekoälyä voidaan käyttää hävittäjäohjaajana kaartotaistelusimulaatiossa?
- Onko kaartotaistelusimulaatioita iteroitaessa havaittavissa nouseva trendi tekoälyn suoritusosassa?

Ensimmäiseen alakysymykseen vastataan luvussa kaksi. Toiseen alakysymykseen vastataan luvussa kolme tukeutuen toisen luvun teoreettiseen viitekehykseen. Kolmanteen alakysymykseen saadaan vastaus kokeen tuloksista, jotka esitellään kappaleessa neljä. Johtopäätöksissä alakysymysten vastauksista johdetaan vastaus päätutkimuskysymykseen.

1.3. Tutkimusmenetelmät

Edellisessä alaluvussa kuvattuun tutkimusongelmaan pyritään saamaan vastaus kokeellisesti. Kokeessa matemaattisesta F-16-fysiikkamallista luodaan simulaatio ja siinä hyödynnetään koneoppimisen menetelmiä neuroverkon ja evoluutioalgoritmin osalta.

Mallilla tarkoitetaan yleensä epätäydellistä ja yksinkertaistettua kuvausta objektista. Mallit voidaan ryhmitellä esittäviin, teoreettisiin ja kuvitteellisiin malleihin. Jos teoreettinen malli esittää matemaattisina yhtälöinä, on kyseessä matemaattinen malli. [5] Koeasetelmaa varten on luotu matemaattinen fysiikkamalli F-16-hävittäjästä. Fysiikkamallia iteroimalla ajan suhteen, saadaan lopputuloksena aikahistoriasimulaatio [6, s. 3]. Simulaatiossa tekoälylle annetaan tehtäväksi kaartotaistelun voittaminen toista vastaavaa tekoälyn ohjaamaa hävittäjää vastaan.

Kokeellisessa tutkimuksessa mitataan muuttujien välisiä suhteita. Koejärjestelyn valossa harmitusti ja systemaattisesti olosuhteita muunnellen tutkitaan, miten saadaan aikaan muutos yhdessä tai useammassa muuttujassa. Muutokset mitataan numeerisesti, minkä lisäksi kokeellisessa tutkimuksessa tyypillisesti testataan hypoteeseja. [7, s. 130]

Kvantitatiivisessa tutkimuksessa tietoa tarkastellaan numeerisesti, eli tutkittavaa asiaa ja sen ominaisuuksia kuvataan numeroiden avulla. Määrällisessä tutkimuksessa tulokset saadaan numeerisesti tai laadullinen aineisto ryhmitellään numeeriseen muotoon. Tutkijan tehtävä on tulkita ja selittää olennainen numeerinen tieto sanallisesti. [8]

Tässä tutkimuksessa kokeen tuloksia analysoidaan kvantitatiivisesti, eli määrällisesti pisteyttämällä tekoälyn suoritus eri koulutusvaiheissa. Tavoitteena on tutkia muutosta valitusta muuttujassa, eli tekoälyn kelpoisuudessa, eräajo määrien kasvaessa. Mahdollinen nouseva trendi muuttujassa vahvistaa hypoteesin, jonka mukaan hävittäjäohjaaja voidaan korvata tekoälyllä kaartotaistelusimulaatiossa. Hypoteesi perustuu tutkijan havaintoihin kappaleessa 1.1 esitetystä DARPA:n järjestämästä tekoälykilpailusta. Lisäksi tekoälyn suoriutumista arvioidaan laadullisesti asiantuntijalausuntojen perusteella.

1.4. Näkökulma ja rajaukset

Tutkimuksen näkökulma on teoreettinen eikä sillä pyritä käytännön sovellukseen. Tutkimuksen koeasetelma rajataan käsittelemään näköetäisyyden sisäpuolella tapahtuvaa hävittäjätaistelua, eli kaartotaistelua F-16-fysiikkamalliin perustuvassa simulaatiossa. Tutkimuksessa tekoäly ohjaa taistelevia hävittäjiä jatkuvan ja täydellisen tilannetiedon perusteella. Todellisessa maailmassa täydellisen tilannetiedon saaminen ei ole mahdollista, sillä oman ja vastustajan paikan määrittämiseen tarvitaan mittaamista ja mittaustulos pitää aina sisällään virheitä ja kohinaa. Tämän lisäksi jatkuva mittaamisen toteuttaminen on haastavaa, joten tilannetieto voi olla lisäksi puutteellista. Edellä mainittujen seikkojen johdosta tutkimuksen lopputulosta ei voida yleistää simulaatiosta todelliseen maailmaan.

Syväoppimisen ja koneoppimisen menetelmien osalta tutkimuksen koasetelmassa on valittu käytettäväksi lähdemateriaaliin perustuen soveltuvimmat, mutta riittävän yksinkertaiset menetelmät, jotta tutkijan työpanoksen ja tutkimuksesta saavutetun hyödyn suhde pysyy hyvänä. Kokeen tavoitteena on todentaa tekoälyn käytettävyys kaartotaistelussa etsimällä nousevaa trendiä tekoälyn suoritusasossa. Tekoälylle annetaan tehtäväksi liikehtiä vastustajan hallintaselemaan ja minimoida tähtäysvirhe, eli mahdollistaa vakaan tykkiammunnan suorittaminen. Jotta tekoälyn tehtävä pysyy riittävän yksinkertaisena, on mallin hävittäjän tehovivun sekä sivuperäsimen asennot asetettu vakioiksi. Samasta syystä vastustajan ammuntojen väistämistä ei ole asetettu tekoälylle tehtäväksi, vaikka se onkin olennainen osa ilmataistelua.

1.5. Lähdemateriaalin esittely ja lähdekritiikki

NASA:n tekninen julkaisu *Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability* [9] pitää sisällään F-16-hävittäjää koskevat parametrit ja suorituskykytaulukot, joiden avulla F-16-hävittäjän dynaaminen lentomalli on mahdollista rakentaa kokeen simulaatioon.

NASA:n tekninen julkaisu perustuu tuulitunnelitesteihin ja simulaatioihin, joissa lennettiin pienen nopeuden kaartotaisteluliikkeitä F-16-hävittäjän sakkausominaisuuksien selvittämiseksi. Testien ja simulaatioiden perusteella F-16-hävittäjän ohjaintietokoneen ohjelmistoja päivitettiin lento-ominaisuuksien parantamiseksi. [9] Julkaisu on peräisin vuodelta 1979, mutta F-16-hävittäjän lento-ominaisuudet eivät ole olennaisesti muuttuneet kuluneiden vuosien aikana. Lähdetä voidaan pitää luotettavana.

Tutkimuksen kannalta tärkein lähde on Brian L. Stevensin, Frank L. Lewisin ja Eric N. Johnsonin *Aircraft Control and Simulation* [6], josta saadaan lentosimulaation ohjelmointia varten tarvittava teorian ja kaavat. Teoksen liitteenä on valmis MATLAB-ohjelmointikielellä rakennettun F-16-simulaatio-ohjelman, joka hyödyntää NASA:n teknisen julkaisun [9] taulukoita. Tätä tutkimusta varten kirjoitettiin vastaava F-16-simulaatio-ohjelma Python-ohjelmointikielellä.

Aircraft Control and Simulation [6] on ilmailutekniikan opiskelijoille ja ammattilaisille suunnattu oppikirja, johon on Google Scholarin mukaan viitattu 4 762 kertaa. Kirjan kirjoittajat ovat alan tunnettuja tutkijoita, joiden töihin on viitattu yhteensä lähes 100 000 kertaa. Kirjan kolmas painos on julkaistu vuonna 2015, joten tieto on ajantasaista. Lähdetä voidaan pitää erittäin luotettavana.

Model of F-16 Fighter Aircraft - Equation of Motions [10] on osa Ying Huon konferenssijulkaisua [11] ja sitä on käytetty työn tukena ohjelmoinnin aikana. Malli hyödyntää tämän tutkimuksen tapaan *Aircraft Control and Simulation* -teoksen valmista lentomallia ja auttoi siten tutkijaa kyseisen mallin soveltamisessa tätä tutkimusta varten. Lisäksi Huon konferenssijulkaisussa on esitetty mallin käyttämät matemaattiset kaavat helposti luettavassa muodossa.

Ying Huo on kirjoittanut julkaisunsa kollegoineen Etelä Kalifornian yliopistossa ja julkaisu on esitelty AIAA:n Guidance, Navigation and Control -konferenssissa. Julkaisun kaavat ovat yhteneviä *Aircraft Control and Simulation* -kirjan [6] kaavojen ja lähdekoodin kanssa. Lähdettä voidaan pitää luotettavana.

John D. Kelleherin kirjan *Syväoppiminen* [12] avulla määritellään tekoälyyn liittyvät käsitteet ja se antaa perustan neuroverkkojen teorialle. Kirjassa käsitellään tarkemmin neuroverkkojen kouluttamista ohjatun oppimisen menetelmällä. Tässä tutkimuksessa käytetty koulutusmenetelmä on vahvistusoppiminen, joten kirjan sisältö ei täysin vastaa tutkimuksen viitekehystä. Kirjaa on kuitenkin hyödynnetty neuroverkkojen teorian sekä suomenkielisen alan termistön osalta. Alan sanasto ei ole vakiintunut suomen kielessä, mikä näkyy myös ristiriitoina suomenetussa lähdemateriaalissa. Väärinkäsitysten välttämiseksi tässä tutkimuksessa käytetään Kelleherin kirjan sanastoa aina kun se on mahdollista.

Kelleher on Dublinin yliopiston akateeminen johtaja ja hänen vuonna 2019 julkaistu kirjansa *Syväoppiminen* [12] on osa MIT Press Essential Knowledge -sarjaa. Lähdettä voidaan pitää luotettavana.

Seyedali Mirjalilin *Neural Networks and Evolution Algorithms* -kirja [13] on evoluutio algoritmien teorian osalta tutkimuksen tärkein lähde. Kirjassa esitellään useita eri tyyppisiä evoluutioalgoritmeja ja niiden ominaisuuksia. Tämän lisäksi niiden suorituskykyä vertaillaan eri tyyppisissä tehtävissä. Kirja antaa kattavan teoreettisen perustan tässä tutkimuksessa käytettävälle evoluutioalgoritmille ja sen soveltamiselle neuroverkon kouluttamisessa.

Springerin julkaisema Mirjalilin kirja [13] on osa Studies in Computational Intelligence -kirjasarjaa. Kirja antaa kattavan katsauksen eri tyyppisiin neuroverkkoihin ja evoluutioalgoritmeihin laadukkaisiin lähteisiin viitaten. Kirjaa voidaan pitää luotettavana lähteenä.

1.6. Määritelmät

Tässä luvussa määritellään tutkielmassa käytettävät käsitteet, eli kerrotaan mitä kullakin käsitteellä tarkoitetaan sen esiintyessä tutkielmassa.

F-16 – F-16 Fighting Falcon on General Dynamicsin suunnittelema monitoimihävittäjä.

6-DOF – Objektilla on vapaus liikkua kuudessa asteessa eli x-, y- ja z-akseleilla, sekä pyöriä näiden akselien ympäri ("*Six Degree of Freedom*").

Malli – Kuvaus tarkasteltavasta objektista. Tässä tutkimuksessa käsitellään F-16-fysiikkamallia, joka on matemaattinen kuvaus kyseisen hävittäjän ominaisuuksista sen liikkuessa tila-avaruudessa.

Simulaatio – Mallista muodostuu simulaatio, kun sitä iteroidaan ajan suhteen.

Kaartotaistelu – Ilmassa näköetäisyyden sisäpuolella hävittäjillä tapahtuva taistelu.

Hallinta-asema – Edullinen asema kaartotaistelussa vastustajan siipilinjan takana ja riittävän lähellä. Hävittäjän keula kohti vastustajaa.

Tekoäly – Ohjelma, joka pyrkii ihmisen toimintaa simuloiden toteuttamaan sille asetettua tehtävää mahdollisimman hyvin.

Syväoppiva algoritmi – Tekoälyohjelma, joka pitää sisällään syväoppimisen / koneoppimisen menetelmiä, joita ovat muun muassa neuroverkot ja evoluutioalgoritmit. Syväoppiva algoritmi kykenee omatoimisesti oppimaan paremmaksi tehtävässään.

Neuroverkko – Syväoppivan algoritmin osa. Neuroverkko on funktio, joka tuottaa deterministisen tuloksen sille annetuista syötteistä. Neuroverkko esitellään tarkemmin luvussa 2.2.1.

Funktio – Deterministinen kuvaus syötteistä tulokseen. Ohjelmoinnissa funktiolla tarkoitetaan ohjelman osaa, joka tuottaa säännönmukaisen ratkaisun määrättyistä syötteistä.

Algoritmi – Prosessi, jota tietokoneohjelma noudattaa. Algoritmi voi koostua esimerkiksi joukosta linkitettyjä funktioita.

Evoluutioalgoritmi – Optimointialgoritmi, joka perustuu malliin luonnonvalinnasta: Parhaat ratkaisut jatkavat evoluutiotaan huonojen karsiutuessa pois.

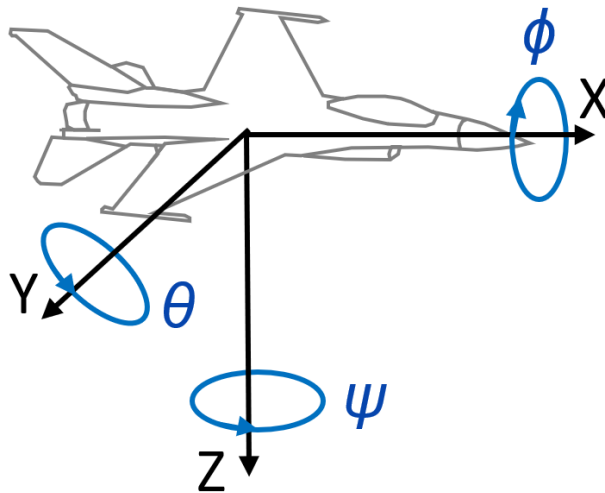
Kelpoisuusfunktio – Määrittelee säännöt, joiden perusteella tekoälyn kelpoisuutta mitataan numeerisesti. Kelpoisuusfunktio asettaa tekoälylle tavoitteet.

2. LENTOMALLIN JA TEKOÄLYN TEORIA

Tämä luku pitää sisällään tutkimuksen teoreettisen viitekehyksen. Luvussa esitellään tutkimuksen kokeessa käytettävä fysiikkamalli ja simulaatio sekä syväoppimisen menetelmät, joilla tekoälyä opetetaan kokeen aikana.

2.1. F-16 6-DOF -simulaatio

Simulaatiossa mallinnetaan kahta identtistä F-16-hävittäjää kaartotaisteluskenaariossa. Simulaatio on rakennettu Python-ohjelmointikielellä ja se pitää sisällään liikeyhtälöt, ilmakehämallin ja F16-lentomallin. Simulaatiota iteroidaan neljännen asteen Runge–Kutta-menetelmällä [6, s. 170-171].



Kuva 1. 6-DOF.

6-DOF (Six Degrees of Freedom) -mallissa kohteella on vapaus liikehtiä kuudessa suunnassa. 3D-koordinaatistossa tämä tarkoittaa liikettä x-, y- ja z-akseleilla sekä rotaatiota näiden akselien ympäri (ϕ , θ , ψ) (kuva 1). Kappale voi siis liikkua ylös/alas, eteen/taakse, sivuille sekä kääntyä kolmen akselinsa ympäri.

Simulaatiossa käytetty matemaattinen malli perustuu NASA:n Langleyssä tekemiin tuulitunnelikokeisiin F-16-pienoismallilla. Mallin taulukot pitävät sisällään aerodynaamiset kertoimet kohtauskulmilla $-20^\circ \sim 90^\circ$ ja sivuluisukulmilla $-30^\circ \sim 30^\circ$. Machin luvun, Reynoldsin luvun ja aeroelastisuuden vaikutuksia ei ole otettu huomioon matemaattisessa mallissa. [9, s. 37]

Lentomalli kaavat ja taulukot ovat peräisin *Aircraft Control and Simulation* -kirjasta [6] ja ne perustuvat edellisessä kappaleessa esitettyihin NASA:n tuulitunnelikokeisiin [9]. Malli on realistinen Machin-lukuun 0,6 asti ja käytettävyyden vuoksi taulukko data on rajattu välille $-10^\circ \leq \alpha \leq 45^\circ$. F-16-hävittäjällä on siiven etureunassa solakot, joita tietokone ohjaa automaattisesti kohtauskulman ja Machin-luvun mukaan. Nopeusalueella, jossa taulukko data on paikkansapitävää, Machin-luvun merkitys on pieni. Tämän vuoksi mallissa etureunasolakot asetetaan kiinteiksi ja niiden aerodynaaminen vaikutus on sulautettu muuhun taulukko dataan. [6, s. 181]

Kaavojen ja taulukon lisäksi *Aircraft Control and Simulation* -kirjan liitteinä on MATLAB-ohjelmointikielellä kirjoitettu koodi F-16-lentomallille [6, s. 714-722]. Tätä tutkimusta varten ohjelma on käännetty Python-ohjelmointikielelle.

2.1.1. Liiketyhtälöt

Seuraavat alaluvut pitävät sisällään matemaattiset yhtälöt, joita käytetään simulaation ilmakehämallissa, F-16-hävittäjän moottorimallissa sekä hävittäjän lentomallissa. Alla olevassa listassa on esitetty kaavoissa käytettävät matemaattiset symbolit, niiden selitykset sekä käytettävät yksiköt. Laskutoimituksissa käytetään yhdysvaltalaisia mittayksiköitä (*U.S Customary System*), joka pohjautuu suurelta osin siirtomaa-aikaiseen brittiläiseen yksikköjärjestelmään [14]. Kyseistä järjestelmää käytetään, koska ne ovat käytössä lähdemateriaalissa F-16-lentomallin osalta. Lisäksi kyseiset mittayksiköt ovat vakiintuneet käyttöön ilmailussa.

Seuraavat listat ja taulukot pitävät sisällään liiketyhtälöissä käytettävät symbolit ja niiden selitykset.

Tilamuuttujat (X):

V	– tosi-ilmanopeus, ft/sek
α	– kohtauskulma, rad ($-10^\circ \sim 45^\circ$)
β	– sivuluisu, rad ($-30^\circ \sim 30^\circ$)
ϕ	– sivukallistuskulma, Euler (Roll), rad
θ	– pituuskallistuskulma, Euler (Pitch), rad
ψ	– suuntakulma, Euler (yaw), rad
p	– pyörimisnopeus (Roll), rad/s
q	– pyörimisnopeus (Pitch), rad/s
r	– pyörimisnopeus (Yaw), rad/s
N_{dis}	– sijainti koordinaatiston N-akselilla, ft
E_{dis}	– sijainti koordinaatiston E-akselilla, ft
h	– sijainti koordinaatiston h-akselilla (korkeus), ft
P_{pow}	– teho

Ohjainmuuttujat (U):

δ_T	– tehovivun asento, (0,0 – 1,0)
δ_E	– korkeusperäsimen asento, deg
δ_A	– siivekkeiden asento, deg
δ_R	– sivuperäsimen asento, deg

Parametrit:

ρ	– ilman tiheys, slugs/ft ³
M	– Machin-luku
T	– työntövoima, N (lb)
m	– kokonaismassa, slugs
$C_{X,t}$	– kokonaisvoimakerroin, x-akseli
$C_{Y,t}$	– kokonaisvoimakerron, y-akseli
$C_{Z,t}$	– kokonaisvoimakerroin, z-akseli
\bar{q}	– dynaaminen paine, psf
p_s	– staattinen paine, psf
$C_{L,t}$	– kokonaiskallistusvoima (Roll)
$C_{M,t}$	– kokonaiskallistusvoima (Pitch)
$C_{N,t}$	– kokonaiskallistusvoima (Yaw)
t	– lämpötila, R
u	– nopeus (x-akseli), ft/s
v	– nopeus (y-akseli), ft/s
w	– nopeus (z-akseli), ft/s
W	– paino (lbs)
b	– kärkiväli (ft)
S	– siipipinta-ala (ft ²)
\bar{c}	– jänne (ft)
I_x	– inertiamomentti (Roll) (slug-ft ²)
I_y	– inertiamomentti (Pitch) (slug-ft ²)
I_z	– inertiamomentti (Yaw) (slug-ft ²)
I_{xz}	– inertiamomentti xz-termi (slug-ft ²)
I_{xy}	– inertiamomentti xy-termi (slug-ft ²)
I_{zy}	– inertiamomentti zy-termi (slug-ft ²)
X_{cgR}	– painopisteen referenssi (ft)
X_{cg}	– painopiste (ft)
g	– putoamiskiihtyvyyys (ft/s ²)
h_E	– moottorin vääntömomentti (slug-ft ² /s)
d_r	– radiaani-aste -vakiokerroin, = 57.29578

Taulukossa 1 esitetään kaavoissa käytettävien ilma-aluksen massa ja geometriaan liittyvien vakioparametrien arvot.

Taulukko 1. F-16-hävittäjän ominaisuudet (massa ja geometria) [10].

Symboli	Parametri	Arvo
W	Ilma-aluksen massa (lbs)	20 500
b	Kärkiväli (ft)	30
S	Siipipinta-ala (ft ²)	300
\bar{c}	Jänne	11,32
X_{cgR}	Painopisteen referenssi	0,35
I_x	Inertiamomentti (Roll) (slug-ft ²)	9 496
I_y	Inertiamomentti (Pitch) (slug-ft ²)	55 814
I_z	Inertiamomentti (Yaw) (slug-ft ²)	63 100
I_{xz}	Inertiamomentti xz-termi (slug-ft ²)	982
I_{xy}	Inertiamomentti xy-termi (slug-ft ²)	0
I_{zy}	Inertiamomentti zy-termi (slug-ft ²)	0

Taulukossa 2 esitetään ilma-aluksen ohjainpinnat ja niiden ominaisuudet. Ohjainpintojen liikkettä on rajoitettu taulukon mukaisesti asentorajoilla, sekä asennon muutosnopeutta rajoittavilla parametreilla. Lisäksi määritetään numeerisesti positiivinen suunta ohjainpintojen asennolle, sekä vaikutus järjestelmälle eli ilma-alukselle.

Taulukko 2. F-16-hävittäjän ohjainpinnat ja niiden ominaisuudet [10].

Symboli	Nimi	Asento- raja	Nopeus- raja	Positiivinen suunta	Vaikutus
δ_E	Korkeus- peräsin	$\pm 25,0^\circ$	$60^\circ/s$	Takareuna alas	Negatiivinen <i>pitch</i> -momentti
δ_A	Siiveke	$\pm 21,5^\circ$	$80^\circ/s$	Oikean siiven taka- reuna alas	Negatiivinen <i>roll</i> -momentti
δ_R	Sivupe- räsin	$\pm 30,0^\circ$	$120^\circ/s$	Takareuna alas	Negatiivinen <i>yaw</i> -momentti, posi- tiivinen <i>roll</i> -momentti

Taulukko 3 pitää sisällään muut mallin kaavoissa tarvittavat vakioparametrit. Moottorin vääntömomentti oletetaan tässä tapauksessa vakioksi kierrosluvusta riippumatta.

Taulukko 3. F-16-hävittäjän muut parametrit [10].

Symboli	Parametri	Arvo	Yksikkö
g	Putoamiskiihtyvyys	32,174	ft/s ²
h_E	Moottorin vääntömomentti	160	slug-ft ² /s
d_r	Radiaani - Aste vakio	57,29578	

Taulukko 4 selventää mistä kunkin aerodynaamisen kertoimen arvo saadaan ja mitä muuttujia tarvitaan kyseisen arvon interpoloimiseksi taulukosta.

Taulukko 4. F-16-hävittäjän aerodynaamiset kertoimet.

Kerroin	Lähde	Muuttujat ($\alpha_d = d_r \alpha, \beta_d = d_r \beta$)
C_{Xq}	Liite 3, Taulukko 1	α_d
C_X	Liite 3, Taulukko 2	α_d, δ_E
C_{Yr}	Liite 3, Taulukko 1	α_d
C_{Yp}	Liite 3, Taulukko 1	α_d
$C_{z,1}$	Liite 3, Taulukko 3	α_d
C_{Zq}	Liite 3, Taulukko 1	α_d
$C_{l,1}$	Liite 3, Taulukko 4	α_d, β_d
C_{Lr}	Liite 3, Taulukko 1	α_d
C_{Lp}	Liite 3, Taulukko 1	α_d
$C_{l,2}$	Liite 3, Taulukko 5	α_d, β_d
$C_{l,3}$	Liite 3, Taulukko 6	α_d, β_d
C_{Mq}	Liite 3, Taulukko 1	α_d
C_m	Liite 3, Taulukko 7	α_d, δ_E
$C_{n,1}$	Liite 3, Taulukko 8	α_d, β_d
C_{Nr}	Liite 3, Taulukko 1	α_d
C_{Np}	Liite 3, Taulukko 1	α_d
$C_{n,2}$	Liite 3, Taulukko 9	α_d, β_d
$C_{n,3}$	Liite 3, Taulukko 10	α_d, β_d

2.1.2. Ilmakehämalli

Simulaatiossa käytetään standardi ilmakehämallia [6, s. 715], jonka kaavoilla lasketaan ilman tiheys ρ (slugs/ft³), lämpötila t (Rankine), dynaaminen paine \bar{q} (psf), staattinen paine p_s (psf) sekä Machin-luku M [6, s. 715]:

$$\rho = 2,377 \times 10^{-3} (1,0 - 0,703 \times 10^{-5} h)^{4,14} \quad (1)$$

$$t = \begin{cases} 519(1,0 - 0,703 \times 10^{-5}h) & h < 35000,0 \\ 390,0 & h \geq 35000,0 \end{cases} \quad (2)$$

$$\begin{cases} \bar{q} = \frac{1}{2}\rho V^2 & \text{dynaaminen paine} \\ p_s = 1715\rho t & \text{staattinen paine} \end{cases} \quad (3)$$

$$M = \frac{V}{\sqrt{1,4 \times 1716,3 \times t}} \quad (4)$$

joissa h viittaa korkeuteen merenpinnasta jalkoina (ft) ja V tosi-ilmanopeuteen jalkoina sekunnissa (ft/s). Lähdekoodi on saatavilla liitteen 4 tiedostona "adc.py".

2.1.3. Moottorimalli

Moottorimalli koostuu funktioista *pdot*, *tgear* ja *thrust*, joiden lähdekoodi on saatavana liitteen 4 tiedostoina "pdot.py", "tgear.py" ja "thrust.py". Alkuperäinen lähdekoodi on peräisin *Aircraft Control and Simulation* -kirjan liitteestä A.7 MATLAB-ohjelmointikielellä [6, s. 715-716]. Tutkija on kääntänyt koodin Python-ohjelmointikielelle tätä tutkimusta varten.

Tehovasteen viive on mallinnettu funktiossa *pdot*. Funktiolle *pdot* annetaan muuttujina moottorille käsketty tehoasetus *cpow* ja nykyinen moottorin tehoasetus *pow*. Funktion tuloksena saadaan tehoasetuksen muutos *power_dot*. [6, s. 715-716]

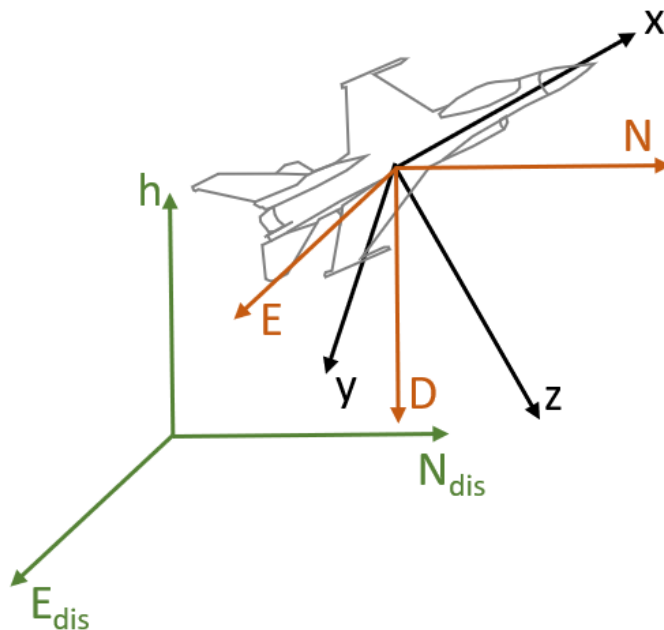
Tehovivun vaihteisto on mallinnettu funktiossa *tgear*. Funktiolle *tgear* annetaan muuttujana tehovivun asento *thtl* ja tuloksena saadaan moottorille käsketty tehoasetus *cpow*. [6, s. 715]

Thrust -funktiolle annetaan muuttujaksi moottorin tehoasetus *pow* ja se antaa tulokseksi moottorin työntövoiman *thrust*. Funktion taulukoissa rivit vastaavat Machin-lukuja 0,2M välein (0 ~ 1,0M). Sarakkeet puolestaan vastaavat korkeuksia 10 000ft välein (0 ~ 50 000ft). Taulukkoja on kolme tehoasetuksille "idle", "military" ja "maximum". Funktioon liitetty algoritmi interpoloi työntövoiman arvon lineaarisesti taulukoiden sisältä. Algoritmi kykenee myös ekstrapoloiamaan arvoja taulukoiden ulkopuolelta, mutta tällöin tulos ei välttämättä ole realistinen. [6, s. 716]

2.1.4. F-16-lentomalli

F-16-lentomallin runko on ohjelmoitu yhdeksi omaksi funktiokseen f_{f16} , joka on saatavissa liitteen 4 tiedostona "f_16.py". Funktiolle syötetään tilamuuttujat X ja ohjainmuuttujat U . Se pitää sisällään laskentaan tarvittavat vakioparametrit ja hakee muut simulaation aikana muuttuvat laskentaan tarvittavat parametrit ilmakehämallin ja moottorimallin funktioista. Aerodynaamiset kertoimet haetaan taulukoista interpolointifunktiota hyödyntäen.

Funktiossa ilma-alukseen vaikuttavat voimat on laskettu xyz-koordinaatistossa, joka on ilma-aluskeskeinen. Tämä tarkoittaa, että x-akseli on ilma-aluksen nokan suuntainen ja positiivinen eteenpäin. Vastaavasti y-akseli on ilma-aluksen siipien suuntainen ja positiivinen oikealle. Pystysuorassa koneen halki kulkee z-akseli, joka on positiivinen alaspäin. Ilma-alus keskeinen xyz-koordinaatisto siis ikään kuin liikkuu ja pyörii ilma-aluksen mukana suhteessa tasokoordinaatistoon N_{dis} , E_{dis} , h (North displacement, East displacement, height), jossa ilma-aluksen liike tapahtuu. Paikallinen referenssikoordinaatisto NED (North, East, Down) on tasokoordinaatiston suuntainen lukuun ottamatta D-akselia, joka on h-akselin vastainen. NED-koordinaatisto liikkuu ilma-aluksen mukana ja sitä käytetään referenssinä ilma-aluksen asentoa määritettäessä (kuva 2).



Kuva 2. xyz-, NED- ja N_{dis} , E_{dis} , h -koordinaatistot.

Ilma-aluksen osanopeudet u , v ja w suhteessa xyz-koordinaatistoon saadaan kaavoilla:

$$u = V \cos \alpha \cos \beta \quad (5)$$

$$v = V \sin \beta \quad (6)$$

$$w = V \sin \alpha \cos \beta \quad (7)$$

x-akselin suuntainen nopeus u lasketaan kaavalla (5), y-akselin suuntainen nopeus v kaavalla (6) ja z-akselin suuntainen nopeus w kaavalla (7). Kaavoissa V on ilma-aluksen tosi-ilmanopeus, α on kohtauskulma ja β on sivuluisukulma.

Kaavalla (8) lasketaan tosi-ilmanopeus V kun osanopeudet u , v ja w tunnetaan:

$$V = \sqrt{u^2 + v^2 + w^2} \quad (8)$$

Osanopeuksien u , v ja w kiihtyvyydet \dot{u} , \dot{v} ja \dot{w} lasketaan kaavoilla:

$$\dot{u} = rv - qw - g \sin \theta + \frac{1}{m} (\bar{q} S C_{x,t} + T) \quad (9)$$

$$\dot{v} = pw - ru + g \cos \theta \sin \phi + \frac{\bar{q} S}{m} C_{y,t} \quad (10)$$

$$\dot{w} = qu - pv + g \cos \theta \cos \phi + \frac{\bar{q} S}{m} C_{z,t} \quad (11)$$

joissa r on ilma-aluksen pyörimisnopeus z-akselin ympäri (*yaw rate*), q on pyörimisnopeus y-akselin ympäri (*pitch rate*) ja p on pyörimisnopeus x-akselin ympäri (*roll rate*) radiaaneina sekunnissa (rad/s). Kaavoissa g on putoamiskiihtyvyys, jolle tässä mallissa käytetään vakioarvoa 32,174 ft/s². ϕ on sivukallistuskulma (*Euler, Roll*) ja θ pituuskallistuskulma (*Euler, Pitch*) radiaaneina (rad). \bar{q} on dynaaminen paine paunoina neliöjalalla (psf) ja se lasketaan ilmähämällin kaavalla (3). S on puolestaan ilma-aluksen siipipinta-ala, jolle F-16-mallissa käytetään vakiota 300 ft/s². m on ilma-aluksen kokonaismassa, jolle käytetään vakiota 20500 lbs.

Kaavassa (9) T on moottorin työntövoima, joka saadaan luvussa 2.1.3 Moottorimalli esitellyllä funktiolla *thrust*. Kaavoissa (9), (10) ja (11) $C_{x,t}$, $C_{y,t}$ ja $C_{z,t}$ ovat F-16 hävittäjälle tyypillisiä x-, y- ja z -akseleiden suuntaisia kokonaisvoimakertoimia, jotka lasketaan kaavoilla (26), (27) ja (28).

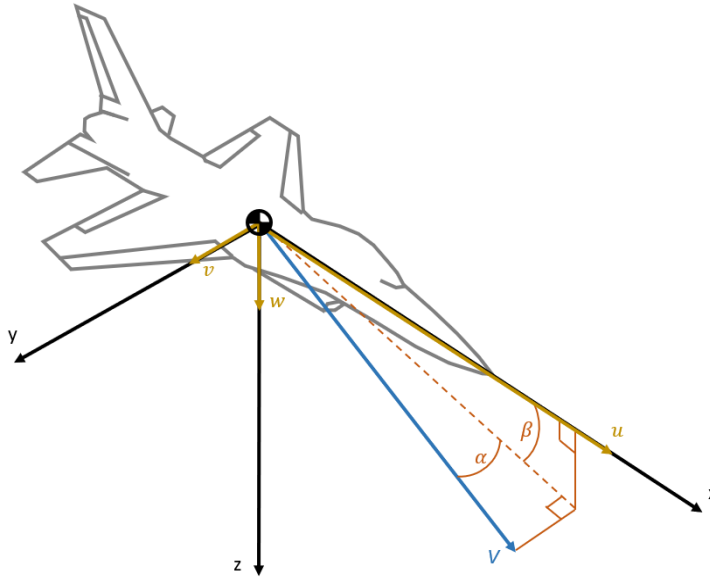
Kaavalla (12) lasketaan tosi-ilmanopeuden V kiihtyvyys \dot{V} .

$$\dot{V} = \frac{u\dot{u} + v\dot{v} + w\dot{w}}{V} \quad (12)$$

Kaavalla (13) lasketaan kohtauskulma α ja kaavalla (14) sivuluisukulma β :

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \quad (13)$$

$$\beta = \sin^{-1}\left(\frac{v}{V}\right) \quad (14)$$



Kuva 3. Kohtauskulma α ja sivuluisukulma β sekä nopeusvektorit u , v , w ja V (Mukaiiltu [10]).

Kohtauskulman α kulmakihtyvyys $\dot{\alpha}$ lasketaan kaavalla:

$$\dot{\alpha} = \frac{u\dot{w} - w\dot{u}}{(v \cos \beta)^2} \quad (15)$$

Sivuluisukulman β kulmakihtyvyys $\dot{\beta}$ lasketaan kaavalla:

$$\dot{\beta} = \frac{V \cos \beta \dot{v} - v \cos \beta \dot{V}}{(v \cos \beta)^2} \quad (16)$$

Euler-kulmien ϕ (*roll*), θ (*pitch*) ja ψ (*yaw*) (kuva 4) kulmakihtyvyydet $\dot{\phi}$, $\dot{\theta}$ ja $\dot{\psi}$ lasketaan kaavoilla:

$$\dot{\phi} = p + \tan \theta (q \sin \phi + r \cos \phi) \quad (17)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (18)$$

$$\dot{\psi} = \frac{q \sin \phi + r \cos \phi}{\cos \theta} \quad (19)$$

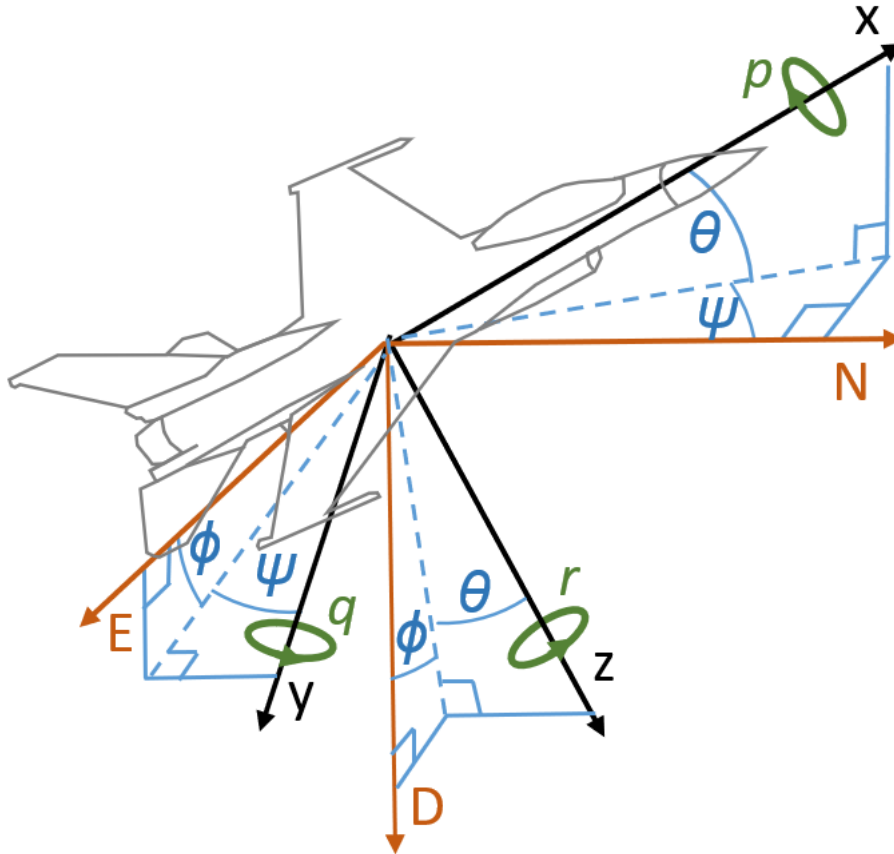
Pyörimisnopeuksien p , q ja r kulmakihtyvyydet \dot{p} , \dot{q} ja \dot{r} lasketaan kaavoilla:

$$\dot{p} = \frac{I_y - I_z}{I_x} qr + \frac{I_{xz}}{I_x} (\dot{r} + pq) + \frac{\bar{q} S b}{I_x} C_{L,t} \quad (20)$$

$$\dot{q} = \frac{I_z - I_x}{I_y} pr + \frac{I_{xz}}{I_y} (r^2 - p^2) + \frac{\bar{q}S\bar{c}}{I_y} C_{M,t} - h_E r \quad (21)$$

$$\dot{r} = \frac{I_x - I_y}{I_z} pq + \frac{I_{xz}}{I_z} (\dot{p} - qr) + \frac{\bar{q}Sb}{I_z} C_{N,t} - h_E q \quad (22)$$

joissa käytetään inertiaomenteja I_x , I_y , I_z ja I_{xz} , joille tässä F-16-mallissa annetaan taulukon 1 mukaiset vakioarvot. Kaavoissa b on ilma-aluksen siipien kärkiväli ja \bar{c} siiven jänne, joille on annettu vakioarvot taulukossa 1. $C_{L,t}$, $C_{M,t}$ ja $C_{N,t}$ ovat F-16-hävittäjälle tyypillisiä kokonaiskallistusvoimakertoimia, jotka lasketaan kaavoilla (29), (30) ja (31).



Kuva 4. Euler-kulmat ja niiden suhde koordinaatistoihin sekä näiden ympäri tapahtuvaan rotaatioon.

Ilma-aluksen nopeudet \dot{N}_{dis} , \dot{E}_{dis} ja \dot{h}_{dis} akseleilla N_{dis} , E_{dis} ja h lasketaan kaavoilla:

$$\begin{aligned} \dot{N}_{dis} = & V \cos \alpha \cos \beta \cos \theta \cos \psi + V \sin \beta (\sin \phi \cos \psi \sin \theta - \cos \phi \sin \psi) \\ & + V \sin \alpha \cos \beta (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \end{aligned} \quad (23)$$

$$\begin{aligned} \dot{E}_{dis} = & V \cos \alpha \cos \beta \cos \theta \sin \psi + V \sin \beta (\sin \phi \sin \psi \sin \theta - \cos \phi \cos \psi) \\ & + V \sin \alpha \cos \beta (\cos \phi \sin \theta \sin \psi + \sin \phi \cos \psi) \end{aligned} \quad (24)$$

$$\dot{h}_{dis} = V \cos \alpha \cos \beta \sin \theta - V \sin \beta \sin \phi \cos \theta - V \sin \alpha \cos \phi \cos \theta \quad (25)$$

Kaavassa (9) käytettävä x-akselin suuntainen kokonaisvoimakerroin $C_{X,t}$ lasketaan kaavalla:

$$C_{X,t} = \frac{\bar{c}}{2V} C_{Xq}(\alpha_d)q + C_x(\alpha_d, \delta_E) \quad (26)$$

jossa funktio C_x on ilma-aluksen x-akselin suuntaisten aerodynaamisten voimien kerroin. Mm. ilmanvastus kuuluu näihin voimiin. Funktion C_x muuttujista α_d on ilma-aluksen kohtauskulma asteina ja δ_E korkeusperäsimen asento asteina. C_x arvo interpoloidaan liitteen 3 taulukosta 2 funktiolla cx , joka on saatavana liitteen 4 tiedostona "cx.py". Kaavassa (26) käytettävä vaimennuskerroin C_{Xq} interpoloidaan liitteen 3 taulukosta 1 funktiolla *damping*, joka on saatavana liitteen 4 tiedostona "damping.py".

Kaavassa (10) käytettävä y-akselin suuntainen kokonaisvoimakerroin $C_{Y,t}$ lasketaan kaavalla:

$$C_{Y,t} = C_y(\beta_d, \delta_A, \delta_R) + \frac{b}{2V} [C_{Yr}(\alpha_d)r + C_{Yp}(\alpha_d)p] \quad (27. a)$$

joka voidaan F-16-hävittäjän tapauksessa johtaa muotoon:

$$C_{Y,t} = -0,02\beta_d + \frac{b}{2V} [rC_{Yr}(\alpha_d) - C_{Yp}(\alpha_d)p] + 0,021 \frac{\delta_A}{20,0} + 0,086 \frac{\delta_R}{30,0} \quad (27. b)$$

jossa β_d on ilma-aluksen sivuluisukulma asteina. δ_A on siivekkeen asento ja δ_R sivuperäsimen asento asteina. Vaimennuskertoimet C_{Yr} ja C_{Yp} interpoloidaan liitteen 3 taulukosta 1 funktiolla *damping*, joka on saatavana liitteen 4 tiedostona "damping.py". Funktio C_y on saatavana liitteen 4 tiedostona "cy.py".

Kaavassa (11) käytettävä z-akselin suuntainen kokonaisvoimakerroin $C_{Z,t}$ lasketaan kaavalla:

$$C_{Z,t} = C_z(\alpha_d, \beta_d, \delta_E) + \frac{\bar{c}}{2V} C_{Zq}(\alpha_d)q \quad (28. a)$$

joka voidaan F-16-hävittäjän tapauksessa johtaa muotoon:

$$C_{Z,t} = C_{z,1}(\alpha_d, \beta_d) + \frac{\bar{c}}{2V} C_{Zq}(\alpha_d)q - 0,19 \frac{\delta_E}{25,0} \quad (28. b)$$

jossa C_z on ilma-aluksen z-akselin suuntaisten aerodynaamisten voimien kerroin, jonka arvo interpoloidaan liitteen 3 taulukosta 3 funktiolla cz , joka on saatavana liitteen 4 tiedostona "cz.py". Vaimennuskerroin C_{zq} interpoloidaan liitteen 3 taulukosta 1 funktiolla *damping*, joka on saatavana liitteen 4 tiedostona "damping.py". Ilma-alukseen vaikuttavia voimia z-akselin suunnassa ovat mm. nostovoima sekä korkeusperäsimen aiheuttama ilma-aluksen runkoon nähden alaspäin suuntautuva aerodynaaminen voima.

Kaavassa (20) käytettävä x-akselin ympäri tapahtuvan pyörimisen kokonaiskallistusvoimakerroin $C_{L,t}$ lasketaan kaavalla:

$$C_{L,t} = C_l(\alpha_d, \beta_d, \delta_A, \delta_R) + \frac{b}{2V} [rC_{Lr}(\alpha_d) + C_{Lp}(\alpha_d)p] \quad (29. a)$$

joka voidaan F-16-hävittäjän tapauksessa johtaa muotoon:

$$C_{L,t} = C_{l,1}(\alpha_d\beta_d) + \frac{b}{2V} [C_{Lr}(\alpha_d)r + C_{Lp}(\alpha_d)p] + C_{l,2}(\alpha_d, \beta) \frac{\delta_A}{20,0} + C_{l,3}(\alpha_d, \beta_d) \frac{\delta_R}{30} \quad (29. b)$$

johon vaimennuskertoimet C_{Lr} ja C_{Lp} interpoloidaan liitteen 3 taulukosta 1 funktiolla *damping*, joka on saatavana liitteen 4 tiedostona "damping.py". $C_{l,1}$ on ilma-aluksen runkoon kohdistuvien aerodynaamisten voimien aiheuttama kallistusvoimakerroin, jonka arvo interpoloidaan liitteen 3 taulukosta 4 funktiolla cl , joka on saatavana liitteen 4 tiedostona "cl.py". $C_{l,2}$ on siivekkeiden aiheuttama kallistusvoimakerroin, joka on riippuvainen siivekkeiden asennosta δ_A ja se interpoloidaan liitteen 3 taulukosta 5 funktiolla *dlda*, joka on saatavilla liitteen 4 tiedostona "dlda.py" $C_{l,3}$ on sivuperäsimen aiheuttama kallistusvoimakerroin, joka on riippuvainen sivuperäsimen asennosta δ_R ja se interpoloidaan liitteen 3 taulukosta 6 funktiolla *dldr*, joka on saatavana liitteen 4 tiedostona "dldr.py".

Kaavassa (21) käytettävä y-akselin ympäri tapahtuvan pyörimisen kokonaiskallistusvoimakerroin $C_{M,t}$ lasketaan kaavalla:

$$C_{M,t} = \frac{\bar{c}}{2V} C_{Mq}(\alpha_d)q + C_{z,t}(X_{cgr} - X_{cg}) + C_m(\alpha_d, \delta_E) \quad (30)$$

jossa X_{cgr} on ilma-aluksen referenssipainopiste ja X_{cg} ilma-aluksen todellinen painopiste. Nostovoimakertoimen $C_{Z,t}$ vaikuttaa ilma-aluksen aerodynaamiseen keskiöön, joka F-16-hävittäjän tapauksessa sijaitsee painopisteen takana, minkä vuoksi se aiheuttaa ilma-aluksen keulaa alas-päin pyörittävän voiman. Vaimennuskertoimen C_{Mq} interpoloidaan liitteen 3 taulukosta 1 funktiolla *damping*, joka on saatavana liitteen 4 tiedostona "damping.py". C_m on y-akselin ympäri tapahtuvan pyörimisen kallistusvoimakertoimen ja se interpoloidaan liitteen 3 taulukosta 7 funktiolla *cm*, joka on saatavissa liitteen 4 tiedostona "cm.py".

Kaavassa (22) käytettävä z-akselin ympäri tapahtuvan pyörimisen kokonaiskallistusvoimakertoimen $C_{N,t}$ lasketaan kaavalla:

$$C_{N,t} = C_n(\alpha_d, \beta_d, \delta_A, \delta_R) + \frac{b}{2V} [C_{Nr}(\alpha_d)r + C_{Np}(\alpha_d)p] - \frac{\bar{c}}{b} C_{Y,t}(X_{cgr} - X_{cg}) \quad (31.a)$$

joka voidaan F-16-hävittäjän tapauksessa johtaa muotoon:

$$C_{N,t} = C_n(\alpha_d, \beta_d) + \frac{b}{2V} [C_{Nr}(\alpha_d)r + C_{Np}(\alpha_d)p] - \frac{\bar{c}}{b} C_{Y,t}(X_{cgr} - X_{cg}) + C_{n,2}(\alpha_d, \beta_d) \frac{\delta_A}{20,0} + C_{n,3}(\alpha_d, \beta_d) \frac{\delta_R}{30} \quad (31.b)$$

jossa C_n on ilma-aluksen runkoon kohdistuvien aerodynaamisten voimien aiheuttama z-akselin ympäri tapahtuvan pyörimisen kallistusvoimakertoimen ja se interpoloidaan liitteen 3 taulukosta 8 funktiolla *cn*, joka on saatavissa liitteen 4 tiedostona "cn.py". C_{Nr} ja C_{Np} ovat vaimennuskertoimia, joiden arvot interpoloidaan liitteen 3 taulukosta 1 funktiolla *damping*, joka on saatavana liitteen 4 tiedostona "damping.py". Koska kokonaisvoimakertoimen $C_{Y,t}$ vaikuttaa ilma-aluksen painopisteen takana sijaitsevaan aerodynaamiseen keskiöön, aiheuttaa se samalla z-akselin ympäri tapahtuvaa kiertymistä ja on siksi otettava huomioon tässä yhtälössä. Siivekkeiden aiheuttama kallistusvoimakertoimen $C_{n,2}$ interpoloidaan liitteen 3 taulukosta 9 funktiolla *dnda*, joka on saatavana liitteen 4 tiedostona "dnda.py". Sivuperäsimen aiheuttama kallistusvoimakertoimen $C_{n,3}$ interpoloidaan liitteen 3 taulukosta 10 funktiolla *dndr*, joka on saatavan liitteen 4 tiedostona *dndr.py*.

2.1.5. Iterointi Runge–Kutta-menetelmällä

Ilma-aluksen liikettä voidaan kuvata sarjana samanaikaisia ensimmäisen asteen differentiaaliyhtälöitä, jotka ratkaistaan eksplisiittisesti derivaatoiksi, n määrälle muuttujia X_i ja m määrälle syötteitä U_i yleinen muoto on [6, s. 2]:

$$\begin{aligned}
\dot{X}_1 &= f_1(X_1, X_2, \dots, X_n, U_1, U_2, \dots, U_m) \\
\dot{X}_2 &= f_2(X_1, X_2, \dots, X_n, U_1, U_2, \dots, U_m) \\
&\vdots \\
\dot{X}_n &= f_n(X_1, X_2, \dots, X_n, U_1, U_2, \dots, U_m)
\end{aligned} \tag{32}$$

jossa funktiot f_n ovat epälineaarisia funktioita, jotka kuvaavat ilma-aluksen mallintamisessa kohdattavia ilmiöitä. Jos muuttujat X_i yhdessä syötteiden U_i kanssa kuvaavat järjestelmän (ilma-aluksen) käyttäytymistä täysin, voidaan muuttujien X_i sanoa olevan järjestelmän tilamuuttujia ja kaavan (32) kuvaavan muutosta järjestelmän tilassa. Kaava (32) kirjoitetaan muotoon:

$$\dot{X} = f(X, U) \tag{33}$$

jossa tilavektori X on n pituinen lista n määrälle tilamuuttujia ja ohjausvektori U on m pituinen lista m määrälle ohjainmuuttujia. [6, s. 2]

Ilma-aluksen tilan muutokset ajan suhteen voidaan laskea Euler-menetelmällä:

$$X_{k+1} = X_k + f(X_k, U_k)\delta t \tag{34}$$

jossa X_k on k :s arvo tilavektorille ja U_k on k :s arvo ohjainvektorille laskettuna ajassa $k \delta t$, kun $k = 0, 1, 2, \dots$, alkaen lähtötilasta X_0 ja U_0 . Integroinnin aikahyppy δt on pyrittävä pitämään riittävän pienenä, jotta yhden hypyn δt aikana syötevektori U voidaan olettaa vakioksi. Tämä numeerinen iterointi mahdollistaa tilavektorin laskemisen eteenpäin aikahypyin δt , jolloin mallista voidaan luoda aikahistoria simulaatio. [6, s. 3]

Eulerin menetelmän heikkous on sen epätarkkuus, mikäli aikahyppy ei ole erittäin pieni, mikä puolestaan tekee laskennasta hyvin raskaan. [6, s. 171]

Runge–Kutta-menetelmää käyttämällä iteroinnin tarkkuutta voidaan parantaa ilman, että aikahyppyä tarvitsee lyhentää epäkäytännöllisen pieneksi. Yleisin Runge–Kutta-menetelmä on neljännen asteen Runge–Kutta-menetelmä, josta usein käytetään nimitystä "RK4". [6, s. 170-171]

RK4-menetelmää käyttäen tilavektori arvo X_{k+1} voidaan laskea kaavalla:

$$X_{k+1} = X_k + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \tag{35. a}$$

jossa:

$$k_1 = \delta t f(X_k, U_k) \quad (35. b)$$

$$k_2 = \delta t f\left(X_k + \frac{1}{2}k_1, U_k\right) \quad (35. c)$$

$$k_3 = \delta t f\left(X_k + \frac{1}{2}k_2, U_k\right) \quad (35. d)$$

$$k_4 = \delta t f(X_k + k_3, U_k) \quad (35. e)$$

Kertoimet k_i voidaan tulkita mallinnettavan funktion kulmakertoimiksi tarkasteluvälin eri pisteissä. Kertoimien k_i avulla lasketaan tilavektori X_{k+1} arvio aikahypyn δt jälkeen kaavalla (35.a).

Tämän tutkimuksen simulaatiossa RK4-menetelmällä iteroitava funktio on:

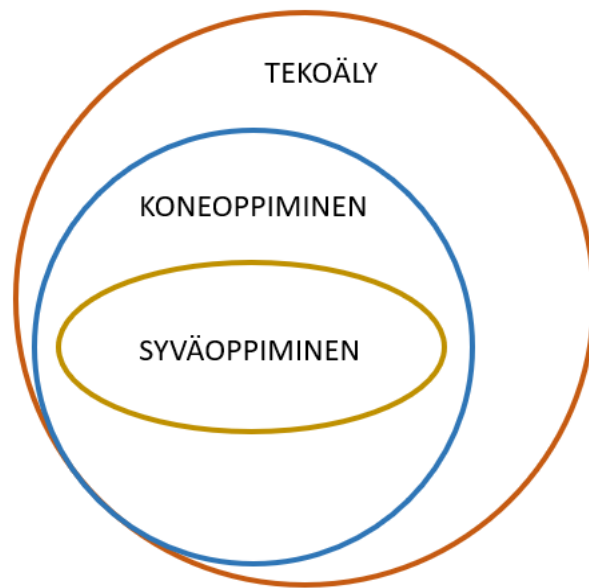
$$f_{f16}(X_k, U_k) \quad (36)$$

jossa X_k on tilamuuttujavektori, joka pitää sisällään F-16-hävittäjän tilamuuttujat. U_k on ohjausmuuttujavektori, joka pitää sisällään F-16-hävittäjän ohjainmuuttujat. Tilamuuttujat ja ohjausmuuttujat on esitetty luvussa 3.1.1. Funktio f_{f16} pitää sisällään alaluvuissa 3.1.2 - 3.1.4 esitetyt ilmakehämallin, moottorimallin, sekä F-16-lentomallin sekä näiden matemaattiset kaavat (1-31).

Tutkimuksen kaartotaistelusimulaatiota varten hävittäjän ohjainmuuttujien U_k arvojen valitseminen annetaan tehtäväksi tekoälylle, jonka teoriaa tarkastellaan seuraavassa alaluvussa tarkemmin.

2.2. Tekoälystä koneoppimiseen ja syväoppimiseen

Syväoppiminen on tekoälyalan osa, joka keskittyy sellaisten suurten neuroverkkomallien luomiseen, jotka tekevät täsmääviä, aineistoon perustuvia päätöksiä. Syväoppiminen sopii erityisesti tilanteisiin, joissa aineisto on monimutkaista ja sitä on paljon. [12, s. 11,15] Kuva 5 havainnollistaa Tekoälyn, koneoppimisen ja syväoppimisen suhdetta.



Kuva 5. Tekoälyn, koneoppimisen ja syväoppimisen suhde (Mukailtu [12, s. 15]).

Koneoppimisen alalla kehitetään ja arvioidaan algoritmeja, joilla tietokone eristää (eli oppii) funktioita tietoaaineistosta (joukosta esimerkkejä). [12] On siis syytä ymmärtää termit tietoaaineisto, algoritmi ja funktio.

Tietoaaineisto yksinkertaisimmillaan tarkoittaa taulukkoa, jonka kukin rivi kuvaa yhtä esimerkiksi joltakin alalta ja kukin sarake pitää sisällään yhden piirteen informaation. Esimerkkinä taulukko 5, joka on lainahakemuksista koostuva tietoaaineisto. Taulukko pitää sisällään yksityiskohdat neljästä lainahakemuksesta, joista kukin pitää sisällään kolme ominaisuutta (vuositulot, nykyinen velka ja maksukyky). [12, s. 16]

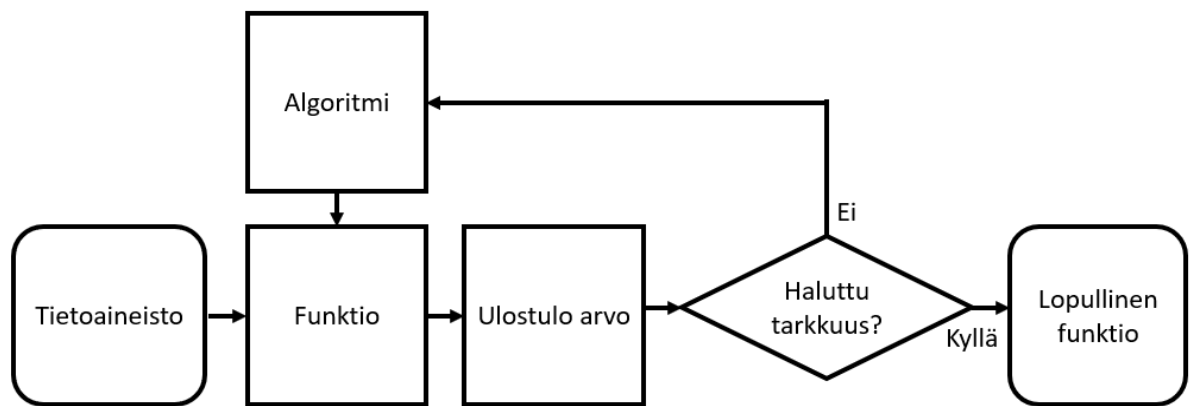
Taulukko 5: Tietoaaineisto lainan hakijoista ja heidän tunnetusta lainanmaksukyvyystään [12, s. 16].

ID	Vuositulot	Nyk. velka	Maksukyky
1	150	-100	100
2	250	-300	-50
3	450	-250	400
4	200	-350	-300

Algoritmi puolestaan on prosessi, jota tietokone noudattaa. Koneoppimisessa algoritmi määrittelee prosessin, jolla tietoaaineistoa analysoidaan pyrkien löytämään siinä toistuvat hahmot (säännönmukaisuudet). Algoritmi voi esimerkiksi löytää hahmon, joka liittyy ihmisen vuositulot ja nykyisen velan hänen lainanmaksukykyynsä (taulukko 5). Matematiikassa tällaisia suhteita kutsutaan funktioiksi. [12, s. 16]

Funktio on deterministinen kuvaus joukolta syötteitä yhteen tai useampaan lopputulokseen. Deterministisyys tarkoittaa, että samat syötearvot tuottavat aina saman lopputuloksen. Funktio voi olla yksinkertainen aritmeettinen operaatio, sarja jos-niin-sääntöjä tai paljon monimutkaisempi esitys syötteestä lopputulokseen. [12, s. 18]

Yleisesti koneoppimiseen käytetty algoritmi on hakuprosessi, joka etsii mahdollisten funktioiden joukosta sen, joka kuvaa parhaiten tietoaineistossa ilmeneviä piirteitä ja niiden välisiä suhteita. Oppimisella tarkoitetaan tässä tapauksessa funktion eristämistä aineistosta. [12, s. 18] Tämä prosessi voidaan esittää yksinkertaisena prosessikaaviona kuvan 6 mukaisesti.



Kuva 6. Koneoppiminen prosessina.

Tässä prosessissa tietoaineisto syötetään yhdelle mahdollisten funktioiden joukosta satunnaisesti valitulle funktiolle ja tämän tuottama lopputulos (ulostulo arvo) tarkistetaan. Mikäli lopputulokseen ei olla tyytyväisiä, käytetään algoritmia, joka muokkaa valittua funktiota tai valitsee mahdollisten funktioiden joukosta uuden ehdokkaan. Prosessia iteroidaan, kunnes saavutetaan haluttu tarkkuus lopputuloksen osalta.

Prosessin yksinkertaisuudesta huolimatta koneoppiminen on laskennallisesti vaativa tehtävä. Tietokoneiden avustamanakin koneoppimiseen liittyy monia haasteita, joista suurimmat liittyvät tietoaineistoihin. Aineistossa on hyvin usein kohinaa, joten opettamiseen käytettyä aineistoa täsmällisesti vastaavan funktion tavoittelu ei välttämättä ole paras ratkaisu, sillä se vastaa kohinan oppimista. Lisäksi useissa tapauksissa mahdollisten funktioiden joukko on suurempi kuin tietoaineiston esimerkkien joukko. Tämä tarkoittaa, että koneoppiminen on niin kutsuttu ”huonosti asetettu ongelma”, eli esitetty informaatio ei riitä yhden parhaan ratkaisun löytämiseksi. [12, s. 24]

Huonosti asetetun ongelman ratkaisemiseksi voidaan kerätä lisää aineistoa toivoen uusien esimerkkien auttavan erottamaan oikean taustalla olevan funktion jäljellä olevien vaihtoehtojen joukosta. Usein tämä ei kuitenkaan ole mahdollista, sillä joko tietoaineistoa ei ole olemassa, tai sen kerääminen on liian kallista. Sen sijaan ongelma voidaan kiertää asettamalla joukko parasta funktiota koskevia oletuksia ja vaikuttamalla prosessiin, jolla parasta funktiota etsitään. Näistä oletuksista käytetään nimitystä induktiiviset vinoumat (*inductive bias*). [12, s. 25]

Koneoppiva algoritmi siis valitsee parhaan funktion perustuen tietoaineistoon ja funktiolle asetettuihin oletuksiin, induktiivisiin vinoumiin. Vinouman ollessa liian vahva, jättää algoritmi huomiotta tärkeää informaatiota aineisosta, eikä lopullinen funktio tavoita aineiston todellista luonnetta. Tätä ongelmaa kutsutaan alisovittamiseksi (*underfitting*). Toisaalta vinouman ollessa liian heikko, saa algoritmi liikaa vapautta löytää funktio, joka sopii yhteen aineiston kanssa. Tällöin lopullinen funktio on todennäköisesti liian monimutkainen ja sopii yhteen aineiston kohinan kanssa, mikä vähentää funktion kykyä yleistyä uuteen aineistoon. Tällöin on kyse ylisovittamisesta (*overfitting*). [12, s. 26-27]

Keskeistä on löytää algoritmi, joka tasapainottaa aineistoa ja induktiivista vinoumaa siten, että lopullinen funktio ei ole alisovitettu eikä ylisovitettu ja täten päättelee täsmävästi ja käsittelee uusia esimerkkejä menestyksekkäästi. Ongelman ollessa riittävän monimutkainen, koneoppimisen käytön oikeuttamiseksi ei voida etukäteen tietää, mitkä vinouttavat oletukset valikoivat oikean funktion aineistosta. Niinpä on tehtävä valistuneita arvauksia ja käytettävä yritykseen ja erehdykseen perustuvaa kokeilua parhaiten toimivan algoritmin löytämiseksi. [12, s. 27-28]

Koneoppimisen prosessin onnistumisen edellytyksen on kolmen keskeisen tekijän oikea valinta. Ensinnäkin tarvitaan aineisto (*data set*), joka voi olla mm. joukko historiallisia esimerkkejä. Toiseksi on valittava joukko funktioita, jotka algoritmi käy läpi löytääkseen niistä parhaiten aineiston kanssa yhteensopivan. Kolmanneksi tarvitaan kelpoisuusfunktio (*fitness function*), jolla arvioidaan, kuinka kukin ehdokasfunktio sopii yhteen aineiston kanssa. [12, s. 29]

Ominaisuuksien valinta tietoaineistoon on vaihtokauppaa. Mikäli halutaan käyttää minimaalinen määrä tietoaineiston ominaisuuksia, jää jokin erittäin tärkeä piirre todennäköisesti aineiston ulkopuolelle. Jos taas halutaan ottaa mahdollisimman monta piirrettä huomioon, mukaan tulee todennäköisesti epäolennaisia ja tarpeettomia piirteitä. Tämäkin johtaa todennäköisesti huonosti toimivaan funktioon. [12, s. 30]

Toinen prosessin tärkeistä tekijöistä on ehdokasfunktioiden joukko, joita algoritmi käy läpi mahdollisina selityksinä aineiston hahmoille. Yleensä funktiojoukko määritellään implisiittisesti koneoppivan algoritmin induktiivisen vinouman ja käytetyn funktion esityksen avulla. [12, s. 31]

Viimeinen koneoppimisen prosessin tekijöistä kelpoisuuden mitta, jolla tarkoitetaan funktiota, joka vertaa ehdokasalgoritmin ulostuloja määrätyllä tavalla aineistoon. Vertailusta saadaan tulokseksi ehdokasfunktioille arvo, joka kuvaa sen kelpoisuutta suhteessa aineistoon. [12, s. 31]

Edellä mainittujen kolmen tekijän määrittelystä riippuen päädytään koneoppimisen kolmeen luokkaan, jotka ovat: ohjattu-, ohjaamaton- ja vahvistusoppiminen. [12, s. 32]

Ohjattu oppiminen on koneoppimisen yleisin muoto. Siinä ehdokasfunktion ulostuloarvoa verrataan kohdearvoon. Tämän menetelmän käyttö siis edellyttää aineistoa, joka pitää sisällään sen jokaista esimerkkiä vastaavan kohdearvon. Ohjaamatonta oppimista puolestaan käytetään yleisesti, kun aineisto halutaan jakaa ryppäisiin. Tässä tapauksessa aineisto ei pidä sisällään kohdearvoja, minkä vuoksi algoritmi ei voi suoraan arvioida funktion kelpoisuutta. Sen sijaan koneoppiva algoritmi yrittää tunnistaa funktiot, jotka kuvaavat samankaltaiset esimerkit ryppäiksi siten, että kunkin ryppään sisällä esimerkit ovat samankaltaisempia kuin verrattaessa muiden ryppäiden esimerkkeihin. [12, s. 32-34]

Vahvistusoppiminen puolestaan sopii parhaiten niin kutsuttuihin online-säätö-tehtäviin, joista paras esimerkki lienee pelien pelaaminen. Tällaisessa tehtävässä toimijan on opittavan, miten sen tulee toimia ympäristössään saadakseen palkinnon. Funktio siis pyrkii kuvaamaan nykyisen ympäristönsä ja sisäisen tilansa (muistinsa) tilanteeseen sopivaksi toiminnaksi (ulostulo). Tällaisista tilanteista on vaikea luoda historiallisia tietoaineistoja, joten vahvistusoppiminen toteutetaan vapauttamalla toimija ympäristöönsä, jossa se kokeilee alustavasti satunnaisia toimintatapoja päivittäen näitä vasteena ympäristöstään saamilleen palkinnoille. [12, s. 34-35]

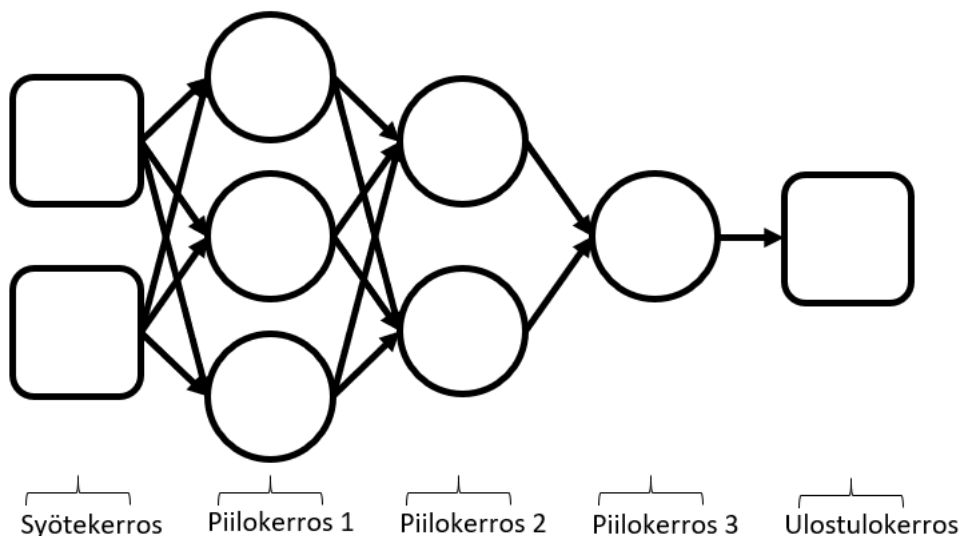
Vahvistusoppimisessa ulostulo funktion (eli toimijan tekojen) kohdefunktio ei ole kytketty palkintomekanismiin. Tämän seurauksena suoraa välitöntä palautetta ei ehkä ole saatavilla, vaan toiminnan seuraukset paljastuvat vasta myöhemmin. Esimerkiksi shakkiottelussa palaute voi olla +1, kun toimija voittaa ottelun ja -1, kun toimija häviää ottelun. Palaute ei kuitenkaan ole saatavilla ennen ottelun viimeistä siirtoa. Yksi vahvistusoppimisen haasteista on suunnitella palkitsemisfunktio, joka mahdollistaa palautteen antamisen toimijalle koko suorituksen ajan. [12, s. 35]

Tietokonepeliympäristöt sopivat erityisen hyvin vahvistusoppimiselle, sillä toimija voi pelata samaa tilannetta tuhansia kertoja oppiakseen palkitsevimman toimintavan. Toimija voi kokeilla valtavan määrän erilaisia toimintavaihtoehtoja ilman niistä luotua raskasta ja suurta tietokantaa. [12, s. 35]

Tämän tutkimuksen kokeessa aineisto ei pidä sisällään kohdearvoja, joten ohjattu oppiminen on poissuljettu vaihtoehto. Ohjaamaton oppiminen ei ole sopiva menetelmä, sillä tutkimuksen tavoitteena ei ole luoda aineistosta ryppäitä, vaan arvioida funktioiden kelpoisuutta. Simulaatioympäristöt muistattavat monilta osin tietokonepeliympäristöjä. Vaikka näitä kahta käsitettä ei voida pitää synonyymeinä, ovat peruseriaatteet niissä toimijalle hyvin samankaltaisia. Edellä mainittujen tekijöiden vuoksi tätä tutkimusta varten koneoppimismenetelmäksi valittiin vahvistusoppiminen, jota käytetään menetelmänä tutkimuksen neuroverkkoa kouluttaessa.

2.2.1. Syväoppiminen ja Neuroverkot

Syväoppiminen on koneoppimisen alalaji ja siinä keskitytään syviin keinotekoisii neuroverkkomalleihin. Neuroverkko on yksi tapa esittää edellisessä luvussa esitetty funktio, jolla aineistoa käsitellään ulostuloarvon saamiseksi. Neuroverkot ovat siis tapa esittää syväoppivan algoritmin aineistosta löytämät hahmot, eli funktiot. [12, s. 18] Kuvassa 7 esitetään periaatteellinen piirros yksinkertaisesta neuroverkosta.

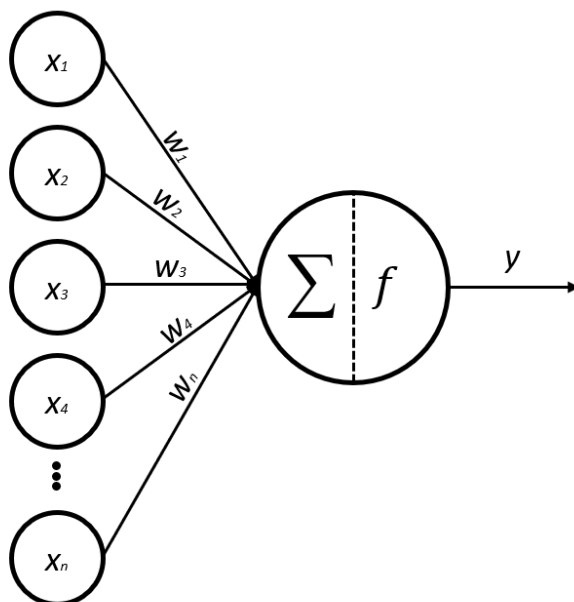


Kuva 7. Periaatepiirros neuroverkosta (Mukailtu [12, s. 66]).

Tapana on ajatella verkon neuronien muodostavan kerroksia. Syväoppivasta neuroverkosta puhutaan, kun verkossa on vähintään kaksi piilokerrosta. [12, s. 66] Kuvan 7:n verkolla on viisi kerrosta: yksi syötekerros, kolme piilokerrosta sekä ulostulokerros.

Kuvassa 7:n syötekerroksen laatikot kuvaavat verkon muistipaikkoja. Muistipaikkojen kautta aineistosta valitut syötteet ohjataan verkkoon. Muistipaikkoja voidaan pitää aineistoneuroneina. Ne eivät käsittele tietoa millään tavalla, vaan toimivat tallennustilana aineistosta saaduille arvoille. Kuvan ympyrät puolestaan ovat neuroneita, joista kukin tuottaa oman funktionsa, jonka syötteet ovat joukko lukuja, jotka se kuvaa ulostulo arvoksi. Nuolet puolestaan ovat verkon kytkentöjä ja ne kertovat, miten kunkin neuronin ulostulot siirtyvät syötteinä seuraaville neuroneille. Kullakin kytkennällä on oma painoarvona, joka on yksinkertainen luku, jolla edellisestä kerroksesta saatu ulostulo kerrotaan ennen kuin se annetaan syötteeksi seuraavan kerroksen neuronille. Neuronien kuvaamat yksinkertaiset funktiot siis yhdistetään monimutkaisemmaksi funktioiden verkoksi painotettujen kytkentöjen avulla. Kytkentöjen kertoimet vaikuttavat eniten siihen, miten verkko käsittelee sille syötettyä tietoa. Neuroverkon kouluttamisen ydin on etsiä verkolle parhaat mahdolliset kytkentöjen kertoimet. [12, s. 19,68]

Verkon kukin neuroni toteuttaa kaksivaiheisen prosessin, joka kuvaa sille annetut syötteet ulostuloksi. Ensimmäinen vaihe on laskea neuronille annettujen painotettujen syötteiden summa. Toisessa vaiheessa tämän laskutoimituksen tulos annetaan aktivointifunktiolle, joka kuvaa painotetun summan neuronin lopulliseksi tulosarvoksi. [12, s. 69]



Kuva 8. Neuronin rakenne (Mukailtu [12, s. 70]).

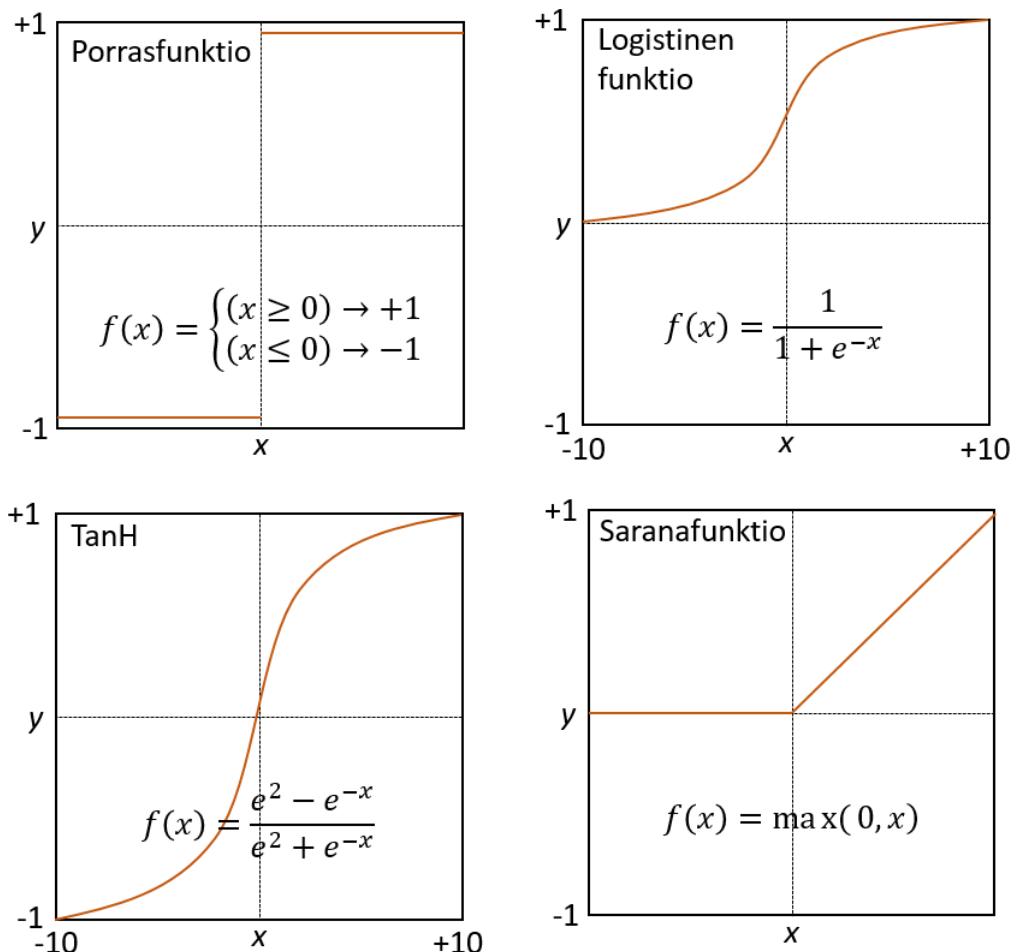
Kuvasta 8 nähdään miten edellä mainitut tietojenkäsittelyn vaiheet heijastuvat neuronin rakenteeseen. Kuvassa Σ tarkoittaa painotetun summan laskemista ja f aktivointifunktiota, joka käsittelee painotetun summan lopulliseksi tulokseksi y . Neuronien saamat syötteet ovat x_n ja niitä vastaavat kytkentöjen painokertoimet ovat w_n .

Matemaattisesti tämä prosessi voidaan kuvata kaavalla:

$$y = f\left(\sum_{i=1}^n x_i w_i\right) \quad (37)$$

joka on johdettu Kelleherin esittämistä kaavoista [12, s. 70].

Aktivointifunktio määrittelee, miten kukin neuroni käsittelee sille syötettyä tietoa. Yksinkertaisimmillaan aktivointifunktio voi olla lineaarinen funktio kuten: $f(x) = x$. Aktivointifunktio voi olla myös monimutkaisempi epälineaarinen funktio. Epälineaarinen aktivointifunktio on hyödyllinen, jos mallinnettava ongelma on epälineaarinen. Mikäli verkon yksittäisten neuroneiden funktiot ovat lineaarisia, rajoittaa se koko verkon luoman kuvauksen lineaarisiksi. Yritys kuvata epälineaarista ongelmaa lineaarisella mallilla todennäköisesti epäonnistuu. Silloin neuronin aktivoimista varten on syytä valita epälineaarinen funktio, joita ovat muun muassa porrask-funktio, logistinen funktio, TanH-funktio ja saranafunktio. [12, s. 71,72] Nämä funktiot esitetään kuvaajina matemaattisten kaavojensa kanssa kuvassa 10.



Kuva 10. Erilaisia neuroverkon aktivointifunktioita (Mukaiitu [12, s. 70]).

Neuroverkkojen rakenteellisen luonteen ansiosta niiden toiminta voidaan ymmärtää perustavalla tasolla. Sama rakenteellinen luonne johtaa moniin valintoihin, jotka verkkoa rakentaessa on tehtävä. Verkkoa rakentaessa on valittava käytettävät aktivointifunktiot, kerrosten määrä, neuronien määrä kerroksissa sekä kytkentöjen sijainti ja suunta. Näitä neuroverkon ominaisuuksia kutsutaan koneoppimisen termein hyperparametreiksi. Niiden toimivuutta ei ole helppo arvioida aineistoon perustuen, minkä vuoksi hyperparametrit on valittava joko heuristisin säännöin, intuitiolla tai yrityksen ja erehdyksen kautta. [12, s. 92,93]

Neuroverkot voidaan jakaa rakenteensa mukaan useisiin eri tyyppisiin, joista esimerkkinä mainittakoon: *Feedforward Neural Network (FNN)*, *Convolutional Neural Network (CNN)*, *Radial Basis Network (RBF)*, *Kohonen Self-organizing Network* ja *Recurrent Neural Network* [13, s. 141]. Eri neuroverkkotyypeistä käytetään tässä niiden englanninkielisiä termejä, sillä suomen kielessä termit eivät ole vakiintuneet ja voivat johtaa väärinkäsityksiin.

Tämän tutkimuksen koetta varten käytettäväksi valittiin *Feedforward Neural Network*, josta tutkija jatkossa käyttää nimitystä myötäsytöneuroverkko. Myötäsytöneuroverkossa kytkennät kuljettavat tietoa ainoastaan yhteen suuntaan, kuten kuvassa 7. Myötäsytöneuroverkko valittiin tutkijan aikaisemman kokemuksen perusteella erinomaisen käytettävyyden ja skaalattavuutensa vuoksi.

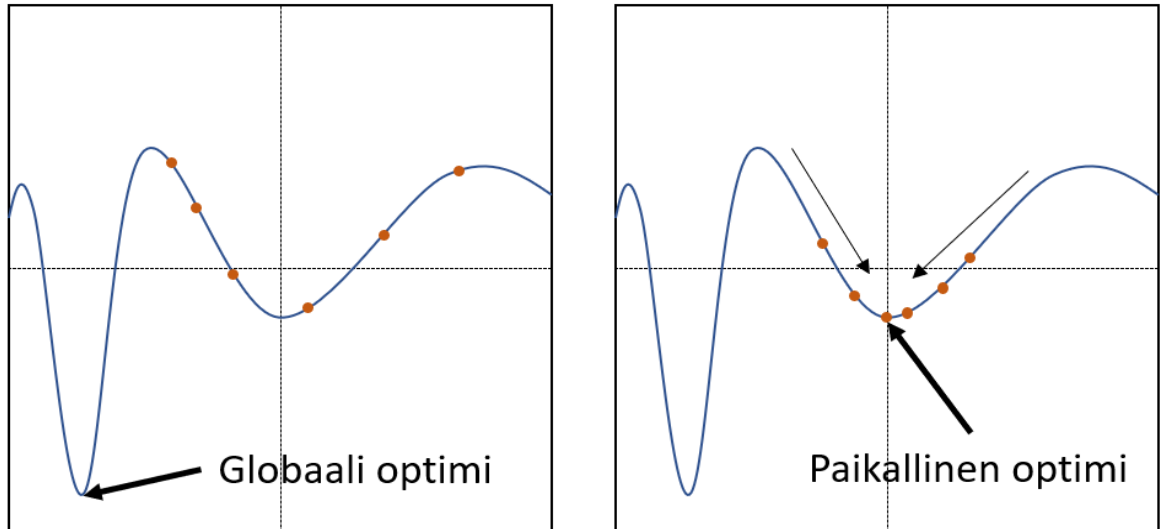
Koetta varten myötäsytöneuroverkoista luodaan populaatio, jota kehitetään iteratiivisesti vahvistusoppimisen menetelmin käyttäen geneettistä algoritmia, joka on yksi evoluutioalgoritmien tyypeistä.

2.3. Evoluutioalgoritmi

Evoluutioalgoritmi on työkalu, jolla etsitään parasta mahdollista ratkaisua optimointiongelmaan. Evoluutioalgoritmeja on useita eri tyyppisiä, mutta yleisesti voidaan sanoa kaikkien evoluutioalgoritmien toimivan samojen peruseräiteiden mukaisesti. Optimointiprosessi aloitetaan luomalla joukko satunnaisia ratkaisuvaihtoehtoja. Tätä joukkoa kutsutaan populaatioksi. Tämän jälkeen populaation yksilöiden suorituksia arvioidaan iteratiivisesti kelpoisuusfunktion avulla ja populaatiota kehitetään optimoimaan kelpoisuusfunktion tulos. [13, s. 3,4]

Yllä kuvattu prosessi vaikuttaa yksinkertaiselta, mutta se pitää sisällään useita haasteita, joista merkittävimpiä ovat: paikallinen optimi, laskentakustannukset, rajoitukset, useat tavoitteet sekä epävarmuustekijät. [13, s. 4]

Todellisen maailman ongelmilla on usein tuntematon etsintäavaruus, joka saattaa pitää sisällään yhden tai useampia paikallisia optimaalisia ratkaisuja, jotka eivät kuitenkaan vastaa ongelman globaalia optimaalista ratkaisua. Paikalliseen optimiin juuttuminen on hyvin yleinen ongelma optimointiprosesseissa. Evoluutioalgoritmit pitävätkin sisällään useita stokastisia laskutoimituksia, joiden tehtävä on estää paikalliseen optimiin juuttuminen. Tästä huolimatta kyseinen ilmiö on ongelma myös evoluutioalgoritmeja käytettäessä. [13, s. 4]



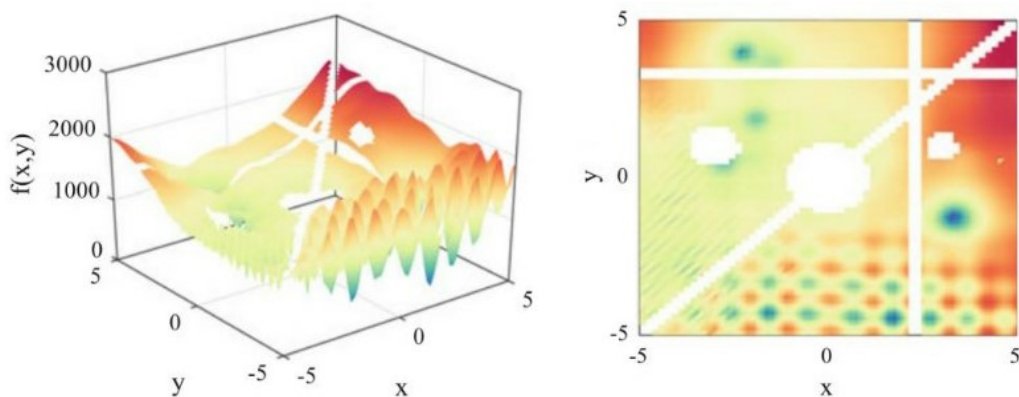
Kuva 11. Populaatio saattaa juuttua paikalliseen optimiin.

Kuva 11 havainnollistaa paikallisen optimin ongelmaa. Kuvassa vasemmalla oranssit pisteet kuvaavat alustavan satunnaisen populaation ratkaisuehdotuksia. Optimointialgoritmillä populaation yksilöitä siirretään kohti parempaa ratkaisua. Alkuperäiset satunnaiset ratkaisuvaihtoehdot eivät kuitenkaan kattaneet aluetta, jossa globaali optimi sijaitsee. Tämän vuoksi populaatio juuttuu kehittyessään paikalliseen optimiin. Kuvan esimerkissä etsintäavaruus on yksiulotteinen jana rajatulla alueella. Ratkaistavissa ongelmissa etsintäavaruus voi kuitenkin olla moniulotteinen, mikä tekee ongelmasta vielä haastavamman.

Evoluutioalgoritmit pitävät sisällään yleensä populaation mahdollisia ratkaisuja, mikä auttaa välttämään paikalliseen optimiin juuttumista. Populaation olemassaolosta johtuen ongelma pitää ratkaista jokaisen populaation jäsenen kohdalla erikseen jokaisen optimointi-iteraation aikana. Laskennallisesti raskaiden ongelmien ratkaiseminen ei siis aina ole käytännöllistä evoluutioalgoritmia käyttäen ilman vaadittavien laskutoimitusten määrään vähentämistä eri keinoin. [13, s. 4]

Usein ongelmissa optimoitavia parametreja on useita, minkä lisäksi tavoitteita saattaa olla useampi. Tämän johdosta etsintäavaruudesta tulee moniulotteinen hyperkuutio, joka usean tavoitteen vuoksi ei enää pidä sisällään yhtä optimaalista ratkaisua, vaan kyseessä on kompromissi. Tämän lisäksi rajoituksista johtuen etsintäavaruus saattaa olla jakaantunut useisiin eri osiin, jotka eivät välttämättä ole yhteydessä toisiinsa. [13, s. 4]

Lisäksi ratkaisun tulee olla luotettava, eli ratkaisun tulee tuottaa mahdollisimman vähän hajontaa, kun otetaan huomioon kaikki mahdolliset epävarmuustekijät. Epävarmuustekijöitä on useita ja kaikissa ongelmissa niiltä ei voida välttyä. [13, s. 4]



Kuva 12. Kaksiulotteinen etsintäavaruus [13, s. 7].

Kuvassa 12 laatikkoon rajattu kaksiulotteinen etsintäavaruus pitää sisällään punaisia alueita, jotka kuvaavat huonoja ratkaisuja. Vihreät alueet puolestaan kuvaavat aluetta, joissa ratkaisut ovat parempia ja sävyltään vihrein pistemäinen alue kuvaa globaalia optimiratkaisua. Etsintäavaruus pitää sisällään useita rajoituksista johtuvia valkoisia alueita, jotka jakavat avaruuden erillisiin alueisiin. Näiden ominaisuuksien lisäksi etsintäavaruuden tulosarvot, eli kuvassa värisävyt, saattavat epävarmuustekijöiden tai ongelman luonteen vuoksi olla jatkuvassa muutoksessa. [13, s. 7]

Toimivan optimointialgoritmin tulee siis kyetä tutkimaan etsintäavaruutta laajasti, jotta globaali optimi osuu sen etsintäalueelle. Tämän lisäksi algoritmin on tunnistettava lupaavat alueet etsintäavaruudesta, ja kyettävä hienosäätämään ratkaisua kohti lähistöllä sijaitsevaa optimia. [13, s. 11]

Globaalin optimin etsimisessä voidaan käyttää useita eri tyyppisiä evoluutioalgoritmeja, joista mainittakoon geneettinen algoritmi ja biografiapohjainen optimointi. Geneettinen algoritmi pitää sisällään populaation mahdollisia ratkaisuita ja tätä populaatiota kehitetään iteratiivisesti globaalin optimin löytämiseksi. Biografiapohjaisen optimointi muistuttaa hyvin paljon geneettistä algoritmia, mutta se pitää sisällään useita populaatioita. Näiden populaatioiden välillä tapahtuu muuttoliikettä määrätyn ehdon. [13]

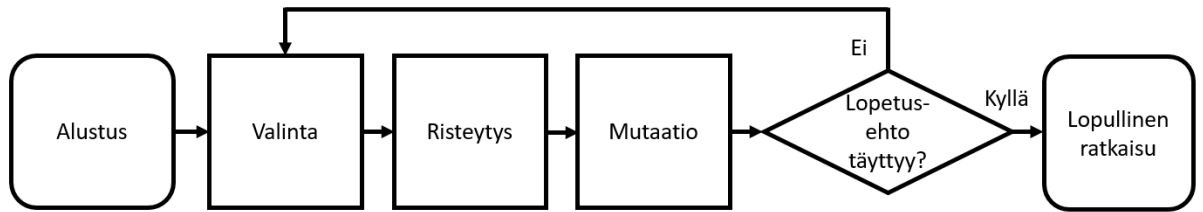
Mirjalilin kirjassa esitettyjen useiden testien perusteella voidaan todeta geneettinen algoritmin ja biografiapohjaisen optimoinnin suoriutuvan useimmista optimointitehtävistä selkeästi muita evoluutioalgoritmeja ja optimointialgoritmeja paremmin. Geneettinen algoritmi suoriutui ongelmista vain hieman biografiapohjaista optimointia heikommin. [13]

Biografiapohjaisen optimoinnin sijasta geneettinen algoritmi valittiin tässä tutkimuksessa käytettäväksi, sillä se on merkittävästi biografiapohjaista optimointia yksinkertaisempi toteuttaa ja laskentatehovaatimuksiltaan kevyempi. Geneettistä algoritmia tarkastellaan seuraavassa alaluvussa tarkemmin.

2.3.1. Geneettinen algoritmi

Geneettinen algoritmi on yksi ensimmäisistä populaation perustuvista stokastisista algoritmeista. Se on ottanut vaikutteita Darwinin evoluutioteoriasta ja simuloi luonnonvalintaa. Populaation yksilöiden voidaan ajatella vastaavan luonnon kromosomeja ja yksittäisten parametrien voidaan ajatella vastaavan yksittäistä geeniä kromosomissa. Geneettinen algoritmi testaa kunkin yksilön kykyä suoriutua sille annetusta tehtävästä kelpoisuusfunktion avulla. Populaatiosta valitaan satunnaisesti yksilöitä jatkamaan sukua siten, että parhaiten tehtävästä suoriutuneilla yksilöillä on suurempi todennäköisyys tulla valituksi. Osaltaan paikalliseen optimiin juuttumista estää huonompien yksilöiden mahdollisuus tulla valituksi, sillä mikäli parhaat yksilöt juuttuvat, voivat muut yksilöt jatkaa kehittymistä. [13, s. 43]

Satunnaisuudestaan huolimatta geneettinen algoritmi kykenee luotettavasti arvioimaan globaalin optimin määrättyyn ongelmaan, sillä se säilyttää parhaat ratkaisut ja käyttää niitä uusien ratkaisuiden kehittämiseen. Tämän seurauksena koko populaatio kehittyy sukupolvi kerrallaan. Risteyttäminen mahdollistaa ratkaisun tarkentamisen tutkimalla etsintäavaruutta kahden vanhemmaksi valitun yksilön tuottaman ratkaisuvaihtoehdon väliltä. Lisäksi geneettinen algoritmi hyödyntää mutaatioita, eli yksilöiden parametreja muutetaan satunnaisesti. Tämän ansiosta populaatio pysyy monimuotoisempuna ja aika ajoin mutaatiot johtavat selkeästi aikaisempaa parempiin tuloksiin, mikä auttaa koko populaatiota kehittymään. [13, s. 43,44]



Kuva 13. Geneettinen algoritmi prosessina.

Geneettisen algoritmin toimintapa voidaan kuvata prosessikaaviona, kuten kuvassa 13. Prosessi alkaa aina populaation alustamisella. Alustamisen jälkeen kelpoisuusfunktiolla testataan ja pisteytetään yksilöt. Parhaat valitaan jatkamaan sukua ja niistä risteytetään seuraavan sukupolven populaatio. Lisäksi mutaatiofunktio vaihtaa satunnaisesti yksilöiden yksittäisiä parametreja. Tämän jälkeen kelpoisuusfunktio ajetaan populaatiolle uudestaan ja palataan valintavaiheeseen. Samaa toistetaan, kunnes algoritmin lopetusehto täyttyy. Seuraavissa alaluvuissa perehdytään prosessin vaiheisiin tarkemmin.

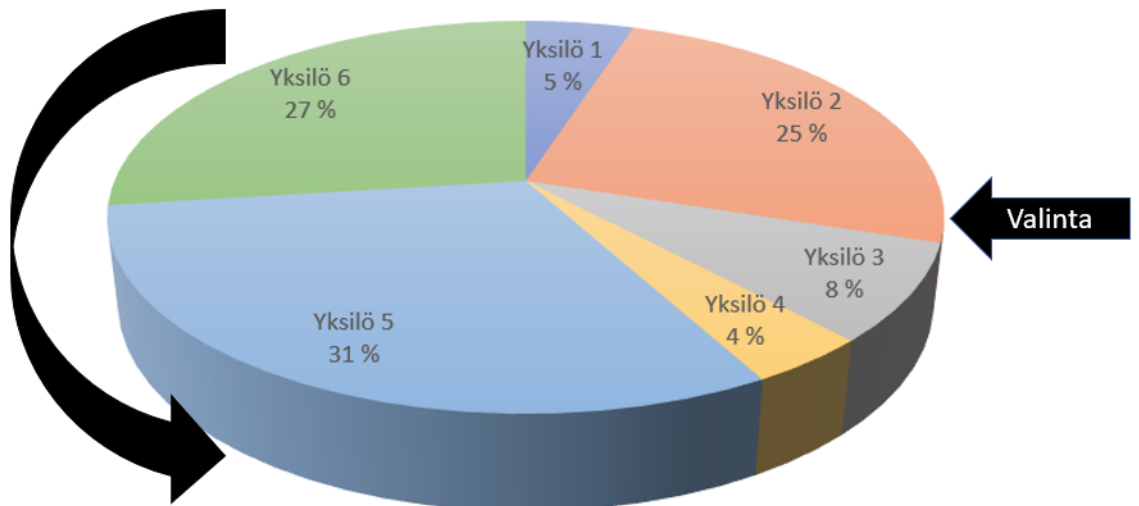
2.3.1.1. Alustus

Geneettinen algoritmi aloittaa alustamalla satunnaisen populaation. Populaatio pitää sisällään lukuisia yksilöitä, joista kukin vastaa yhtä ratkaisuvaihtoehtoa ongelmaan. Jokainen yksilö saa alustuksessa omat satunnaiset parametrinsa. Alustusvaiheessa tärkeintä on jakaa populaation yksilöiden ratkaisuvaihtoehdot mahdollisimman laajasti ja tasaisesti, sillä monimuotoisuus auttaa populaatiota löytämään lupaavat alueet etsintäavaruudesta. [13, s. 44]

2.3.1.2. Valinta

Valintavaiheessa populaation yksilöt pisteytetään suoritustensa perusteella. Tämän jälkeen luonnonvalintaa simuloiden populaatiosta valitaan satunnaisesti parhaiten suoriutuneita suosien yksilöitä seuraavan sukupolven luomista varten. Valintaprosessissa voidaan käyttää useita eri tyyppisiä menetelmiä. [13, s. 44]

Rulettipyörämenetelmä on yksi vaihtoehto valinnan suorittamiseksi. Rulettipyörämenetelmässä kullekin populaation yksilölle annetaan valintatodennäköisyys pisteytyksen perusteella. Yksilöt saavat prosenttiosuuden, joka vastaa kyseisen yksilön pisteitä populaation kokonaispisteistä. Tämän jälkeen generoidaan satunnaisluku. Yksilö, jota satunnaisluku vastaa, valitaan seuraavan sukupolven vanhemmaksi. Prosessia toistetaan, kunnes yksilöitä on valittu tarpeeksi seuraavaa sukupolvea varten. [13, s. 44]



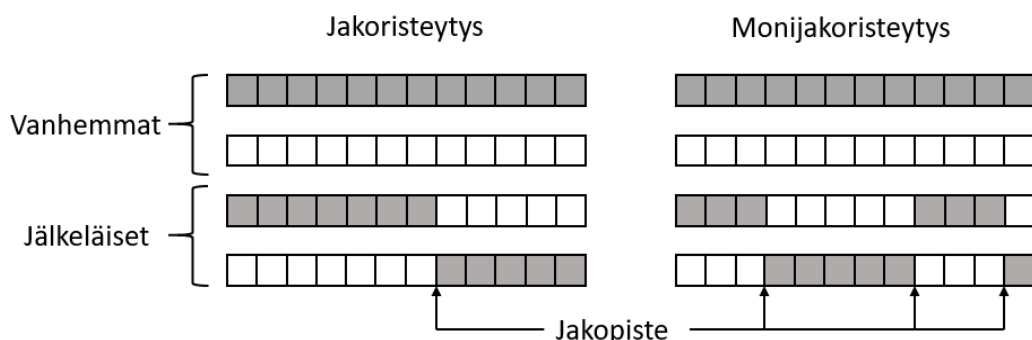
Kuva 14. Rulettipyörämenetelmä (Mukailtu [13, s. 45]).

Kuva 14 havainnollistaa rulettipyörämenetelmää. Satunnaisluvun generoiminen vastaa rulettipyörän pyörittämistä. Pyörä pysähtyy suurimmalla todennäköisyydellä yksilön 5 kohdalle, sillä se kattaa suurimman alueen rulettipyörästä. Myös heikommilla yksilöillä on mahdollisuus tulla valituksi, mikä pitää yllä populaation monimuotoisuutta. Tässä tutkimuksessa käytetään kyseistä rulettipyörämenetelmää valittaessa yksilöt, joista muodostetaan seuraava sukupolvi.

2.3.1.3. Risteytys

Valittuja yksilöitä käytetään risteytysvaiheessa seuraavan sukupolven populaation luomiseen. Vanhempien kromosomien genejä yhdistellään uuden kromosomin synnyttämiseksi. Luonnossa tapahtuvaa prosessia simuloidaan valitsemalla kaksi populaation yksilöä vanhemmiksi, joita risteyttämällä tuotetaan kaksi jälkeläisyksilöä. [13, s. 45,46]

Risteytys voidaan toteuttaa usealla erilaisella menetelmällä, joista jakoristeytys (*Single-point crossover*) [13, s. 45] ja monijakoristeytys (*Multi-point crossover*) [13, s. 46] on esitetty kuvassa 15.

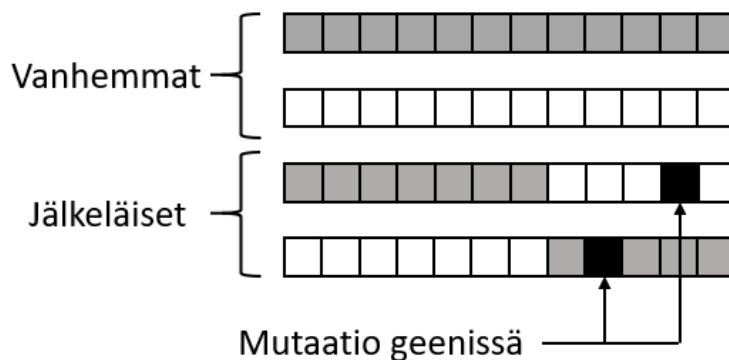


Kuva 15. Jakoristeytys ja monijakoristeytys (Mukailtu [13, s. 46]).

Kuvasta 15 nähdään, että vanhemmiksi valitut yksilöt vaihtavat parametrejaan jakoristeytyksessä jakopisteen jälkeen luoden kaksi jälkeläistä. Monijakoristeytyksessä jakopisteet muodostavat useita välejä parametrien muodostamiin ketjuihin ja parametreja vaihdetaan joka toisella välillä. Tämän tutkimuksen geneettinen algoritmi käyttää menetelmänä yhtenäistä risteytystä (*uniform crossover*), jossa uuden yksilön jokainen kerroin (geeni) valitaan vanhempien vastaavien kertoimien väliltä satunnaisesti.

2.3.1.4. Mutaatio

Mutaatio vaiheessa muutetaan jälkeläisten useita yksittäisiä kertoimia. Mutaatiotodennäköisyys pidetään geneettisessä algoritmissa matalana, jotta ei palattaisi takaisin alkeellisempaan täysin satunnaiseen ratkaisuun etsintään. Mutaatio lisää populaation monimuotoisuutta, mikä auttaa välttämään paikalliseen optimiin juuttumista. [13, s. 46] Kuvassa 16 havainnollistetaan mutaatiota.



Kuva 16. Mutaatio (Mukailtu [13, s. 47]).

Kuvassa 16 esitetyllä tavalla risteytysvaiheessa luotujen jälkeläisten satunnaisesti valittuja kertoimia muutetaan halutulla tavalla. Muutettavien geenien valinnassa ja itse muuttamisessa voidaan käyttää useita erilaisia tekniikoita. Tämän tutkimuksen algoritmissa jälkeläisten kukin kerroin mutatoituu 1% todennäköisyydellä.

Seuraavan sukupolven populaatiota luodessa käytetään edellä kuvattujen menetelmien lisäksi usein elitismiä. Tämä tarkoittaa sitä, että populaation parhaat yksilöt kopioidaan sellaiseen seuraavaan sukupolveen. Tavoitteena on estää näiden yksilöiden tahaton heikentyminen risteytyksen ja mutaation toimesta. [13, s. 47]

3. ERÄAJOSIMULAATION JA TEKOÄLYN PARAMETRIT

Tämän tutkimuksen koeasetelmaa varten on kirjoitettu eräajo-ohjelma Python-ohjelmointikielellä. Ohjelma pitää sisällään luvussa 2.1.4 esitellyn F-16-lentomallin. Mallista luodaan simulaatio iteroimalla sitä luvun 2.1.5 mukaisesti RK4-menetelmällä. Simulaatiossa kaksi identtistä F-16-hävittäjää taistelevat toisiaan vastaan tasavertaisista lähtöasetelmistä. Tässä kaartotaistelusimulaatiossa kummankin hävittäjän tulee pyrkiä hallinta-asemaan, eli vastustajan taakse ja oman ilma-aluksen keula kohti vastustajaa etäisyydellä, joka mahdollistaa ammunnan hävittäjän konetykillä.

Simulaatiossa käytetään luvussa 2.2.1 kuvatun kaltaisia neuroverkkoja hävittäjien tekoälyohjaajina. Neuroverkoille annetaan valitut oman ja vastustajan ilma-aluksen tilamuuttujat ja niiden johdannaiset syötteiksi (taulukko 7). Ulostuloarvoksi neuroverkoilta saadaan valitut ohjainmuuttujat, jotka syötetään hävittäjien lentomalliin. Neuroverkkojen kouluttamista varten verkoista luodaan populaatio, jota kehitetään geneettisellä algoritmilla, jonka periaatteet on kuvattu luvussa 2.3.1.

Simulaatioita ajetaan rinnakkain populaation kokoa vastaava määrä ja kelpoisuusfunktiolla arvostellaan yksittäisten neuroverkkojen suoriutumista. Arvostelun perusteella geneettinen algoritmi valitsee neuroverkkoyksilöt, joista se muodostaa risteytyksen, mutaation ja elitismillä avulla seuraavan sukupolven populaation. Tämän jälkeen prosessia toistetaan, kunnes ollaan tyytyväisiä neuroverkkojen suoritustasoon.

3.1. Eräajot

Eräajot suoritettiin Laskentakeskuksen Puhti-supertietokoneella, joka koostuu 682 CPU-solmista, joissa on 192 GB - 1.5 TB muistia. Kukin solmu pitää sisällään kaksi Intelin Xeon Gold 6230 -prosessoria, joissa kussakin on kaksikymmentä 2,1 GHz ydintä. Supertietokoneen teoreettinen huipputeho on 1,8 Petaflopsia. [15] Tyypillisen työpöytä tietokoneen Intel Core i7-9700K -prosessorin teoreettinen laskentateho on 460,8 Gigaflopsia [16]. Puhti-supertietokone on siis noin 4 000 kertaa tehokkaampi kuin tyypillinen työpöytä tietokone.

Laskentaan käytetään yhtä Puhti-supertietokoneen solmua, ja eräajon parametrit on optimoitu hyödyntämään yhden solmun laskentakapasiteettia mahdollisimman tehokkaasti. Useamman solmun käyttäminen olisi mahdollista, mutta tällöin tiedonsiirto sukupolvien välillä aiheuttaisi ylimääräisiä viiveitä laskentaan. Yhden solmun käyttäminen mahdollistaa myös Puhti-supertietokoneen longrun-osion käyttämisen, jolloin eräajoja voidaan suorittaa yhtäjaksoisesti jopa 14 päivän ajan.

3.1.1. Eräajon parametrit

Eräajon parametreja ovat populaation koko, simulaatioiden ajallinen kesto, simulaatioiden keskeytyskriteeri sekä eräajon keskeytyskriteeri. Parametrien valinnassa on otettu huomioon käytävissä oleva laskentakapasiteetti suhteessa suoritettavaan eräajoon. Suuri populaatio mahdollistaa ratkaisun etsimisen laajasti etsintävaruudesta, mutta se lisää myös merkittävästi laskentakuormaa. Eräajon stokastisen luonteen vuoksi optimaalisen populaatiokoon määrittäminen on vaikeaa, joten populaation koko on määritetty vastaamaan laskennassa käytettävän laitteiston kykyjä. Simulaation ajallisen keston on oltava riittävän pitkä, jotta tekoäly saa mahdollisuuden toteuttaa ratkaisuaan. Toisaalta selkeään lopputulokseen päättyvää simulaatiota on turha jatkaa, minkä vuoksi simulaatioille on asetettu keskeytyskriteerit. Tässä alaluvussa esitetään tämän tutkimuksen eräajon parametrit.

Tutkimuksessa populaation koko on asetettu arvoon 80, eli populaatio pitää sisällään 80 yksittäistä neuroverkkoa. Yhden sukupolven aikana suoritetaan tällöin 40 simulaatiota, joissa neuroverkot kilpailevat pareittain toisiaan vastaan. Simulaatioita voidaan tällöin ajaa tehokkaasti rinnakkain yhden Puhti-supertietokoneen solmun 40 ytimellä.

Simulaatioiden ajalliseksi kestoksi on määritetty 4 minuuttia. Tutkijan työpöytä tietokoneella suoritettujen koeajojen perusteella tämä antaa tekoälylle riittävästi aikaa toteuttaa ratkaisuaan kaartotaistelussa. Pääsääntöisesti taistelut päättyvät toisen hävittäjän päästessä hallinta-asemaan, törmätessä maahan tai hallitsemattomaan lentotilaan. Kahden viimeisen vaihtoehdon kohdalla simulaatio keskeytetään, sillä sen jatkaminen ei tuota lisäarvoa tutkimukselle ja tällöin säästetään käytössä olevaa laskentatehoa.

Eräajoja suoritetaan mahdollisimman pitkään käytävissä olevan laskentakapasiteetin puitteissa, eli käytännössä yhdellä Puhti-solmulla longrun-osiolla noin viikon ajan. Tämä mahdollistaa neuroverkkopopulaation kouluttamisen riittävän pitkään suoritustason kehittymisen arvioimiseksi.

Eräajo-ohjelma tallentaa kunkin sukupolven parhaan neuroverkon kertoimet, parhaan yksilön pisteet, populaation pisteiden keskiarvon, sekä eräajon päätyttyä viimeisen sukupolven populaation. Näitä tuloksia analysoimalla pyritään määrittämään suorituskyvyn kehityksen trendi.

3.1.2. Simulaation parametrit

Eräajojen aikana kukin kaartotaistelusimulaatio alkaa vakioidusta aloitustilanteesta, jossa molemmille hävittäjille, siniselle ja punaiselle, annetaan vakioidut aloitusparametrit tilamuuttujien X ja ohjainmuuttujien U osalta. Aloitustilanteessa hävittäjät ovat 10 000 jalan korkeudella vastakkaisilla lentosuunnilla 1 000 jalan etäisyydellä toisistaan sivuttaissuunnassa ja pituussuunnassa. Lähtötilanteen tilamuuttujat X_0 ja ohjainmuuttujat U_0 esitetään taulukossa 6.

Taulukko 6. Simulaation lähtöarvot tilamuuttujien X ja ohjainmuuttujien U osalta.

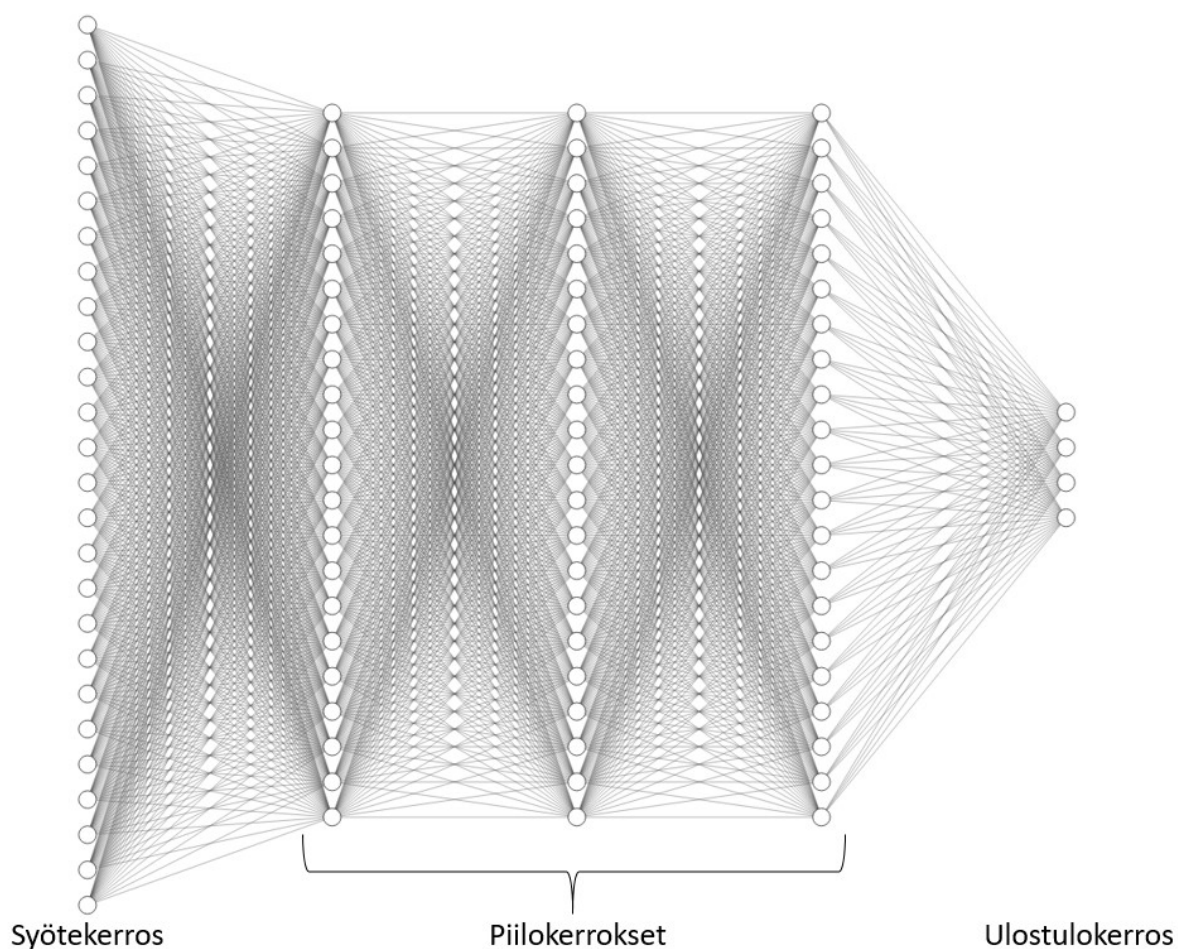
Tilamuuttujat X	Yksikkö	Selite	Lähtöarvo sin.	Lähtöarvo pun.	Huom.
X_1	ft/sec	Velocity	800	800	
X_2	rad	Angle of attack	0	0	
X_3	rad	Sideslip angle	0	0	
X_4	rad	Euler - Roll	0	0	
X_5	rad	Euler - Pitch	0	0	
X_6	rad	Euler - Yaw	π	0	
X_7	rad/sec	Roll rate	0	0	
X_8	rad/sec	Pitch rate	0	0	
X_9	rad/sec	Yaw rate	0	0	
X_{10}	ft	North displacement	1 000	0	
X_{11}	ft	East Displacement	1 000	0	
X_{12}	ft	Altitude	10 000	10 000	
X_{13}	%	Power	0	0	

Ohjainmuuttujat U					
U_1	%	Throttle	100	100	Vakio
U_2	deg	Elevator	0	0	
U_3	deg	Aileron	0	0	
U_4	deg	Rudder	0	0	Vakio

Toisin sanoen Luvun 2.1.5 kaavoissa käytettävä lähtötila siniselle hävittäjälle on tilamuuttujavektorin osalta $X_0 = [800, 0, 0, 0, 0, 0, \pi, 0, 0, 0, 1000, 1000, 10000, 0]$ ja ohjainmuuttujavektorin osalta $U_0 = [100, 0, 0, 0]$. Punaiselle hävittäjälle vastaavasti $X_0 = [800, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10000, 0]$ ja $U_0 = [100, 0, 0, 0]$. Lähtötila on molemmille osapuolille tasavertainen kaartotaistelua ajatellen, sillä molempien nopeus, asema ja asento ovat identtisiä suhteessa toisiinsa.

3.1.3. Neuroverkon parametrit

Eräajossa käytettävät neuroverkot ovat tyypillisiä myötäsytöneuroverkkoja. Ne pitävät sisälleen yhden syötekerroksen, kolme piilokerrosta sekä ulostulokerroksen. Syötekerrokset muodostetaan 26 neuronista. Kussakin piilokerroksessa on 21 neuronia ja ulostulokerroksessa neljä neuronia. Kerrokset liittyvät toisiinsa painotetuilla kytkennöillä. Edellisen kerroksen kustakin neuronista on kytkentä seuraavan kerroksen kaikkiin neuroneihin. Kytkennät ovat yksisuuntaisia, eli tieto verkossa virtaa vain yhteen suuntaan syötekerroksesta piilokerrosten kautta ulostulokerrokseen.



Kuva 17. Visuaalinen malli tutkimuksessa käytettävästä neuroverkosta.

Kuva 17 esittää tutkimuksessa käytettävää neuroverkkoa visuaalisesti. Ympyrät kuvassa ovat verkon yksittäisiä neuroneita, jotka muodostavat viisi kerrosta. Kuvassa vasemmalla on syötekerros, keskellä kolme piilokerrosta ja oikealla ulostulokerros. Viivat kuvaavat kytkentöjä kerrosten välillä. Kytkentöjä on yhteensä 1 512 ja kukin niistä kuvastaa yhtä neuroverkossa käytettävistä kertoimista, joita säätämällä pyritään löytämään optimaalinen funktio syötteistä ulostuloihin.

Neuroverkon syötteen saadaan simulaation aikana sekä positiivisia, että negatiivisia lukuarvoja, joten syötearvot normalisoidaan neuroverkon laskentaa varten -1 ja +1 välille, jotta ne olisivat neuroverkolle helpommin käsiteltävissä. Syötteinä toimivat erikseen valitut ilma-aluksen tilamuuttujat, vastustajan ilma-aluksen tilamuuttujat sekä vastustajan suhteellinen sijainti ilma-alukseen nähden. Syötearvot esitetään tarkemmin taulukossa 7. Taulukossa käytetään muuttujista englanninkielisiä termejä, sillä suomenkieliset termit eivät kaikissa tapauksissa ole yksiselitteisiä tai ilmiötä selkeästi kuvaavia.

Taulukko 7. Neuroverkon syötemuuttujat ja normalisointikaavat.

Syötemuuttuja (X_i)	Syötemuuttujat	Yksikkö	Normalisointi	Huomio
Ownship:				
1	Equivalent airspeed	kt/s	$x/666$	666 kts $\sim 1.0M$ at sea level
2	Axial velocity	ft/s	$x/1124$	
3	Lateral velocity	ft/s	$x/1124$	1124 ft/s $\sim 1.0M$ at sea level
4	Normal velocity	ft/s	$x/1124$	
5	Euler angle (roll)	rad	x/π	
6	Euler angle (pitch)	rad	$x/(\pi/2)$	
7	Altitude	ft	$1-x/20000$	Arvo kasvaa kohti maan pintaa
8	Normal acceleration	G	$x/9$	F-16 G-raja = 9
9	Angle of attack	rad	x/π	
10	Angle of sideslip	rad	$x/(\pi/2)$	
11	Forward displacement of adversary in FRD	ft	$x/6000$	
12	Right displacement of adversary in FRD	ft	$x/6000$	6000ft on likiarvo tykin maksimumtaetäisyydelle
13	Down displacement of adversary in FRD	ft	$x/6000$	
Adversary:				
14	Equivalent airspeed	kt/s	$x/666$	666 kts $\sim 1.0M$ at sea level
15	Axial velocity	ft/s	$x/1124$	
16	Lateral velocity	ft/s	$x/1124$	1124 ft/s $\sim 1.0M$ at sea level
17	Normal velocity	ft/s	$x/1124$	
18	Euler angle (roll)	rad	x/π	
19	Euler angle (pitch)	rad	$x/(\pi/2)$	
20	Altitude	ft	$1-x/20000$	Arvo kasvaa kohti maan pintaa
21	Normal acceleration	G	$x/9$	F-16 G-raja = 9
22	Angle of attack	rad	x/π	
23	Angle of sideslip	rad	$x/(\pi/2)$	
24	Forward displacement of ownship in FRD	ft	$x/6000$	
25	Right displacement of ownship in FRD	ft	$x/6000$	6000ft \sim likiarvo tykin maksimumtaetäisyydelle
26	Down displacement of ownship in FRD	ft	$x/6000$	

Aktivointifunktion valinnassa on otettu huomioon syötearvojen ominaisuudet, sekä haluttujen ulostuloarvojen ominaisuudet. Syötearvot saavat sekä positiivisia, että negatiivisia arvoja, minkä lisäksi ulostuloarvojen halutaan olevan joko positiivisia tai negatiivisia. Tämän vuoksi aktivointifunktioksi valittiin kuvassa 10 esitelty TanH-funktio, sillä se käsittelee neuroverkolle syötettäviä lukuarvoja joustavasti ja normalisoi arvot halutulle välille $-1 \sim +1$.

Tutkimuksessa käytettävä neuroverkko voidaan esittää matemaattisina kaavoin. Ensimmäisen piilokerroksen tulokset lasketaan kaavoilla:

$$s_j = \sum_{i=1}^n (w_{ij}X_i), \quad j = 1,2,3, \dots, h \quad (38. a)$$

$$S_j = \text{TanH}(s_j) = \frac{e^2 - e^{-s_j}}{e^2 + e^{-s_j}}, \quad j = 1,2,3, \dots, h \quad (38. b)$$

jossa S_j on j :nen neuronin ulostuloarvo, n on syöteneuroneiden määrä ja w_{ij} kuvaa kytkennän kerrointa i :nestä syöteneuronista j :nteen piiloneuroniin. X_i tarkoittaa i :ttä syötearvoa, jotka ovat taulukon 7 mukaisia parametreja. Koska tutkimuksessa käytettävän neuroverkon syötekerros on 26 neuronin levyinen ja piilokerrokset 21 neuronin levyisiä, ovat $n = 26$ ja $h = 21$.

Seuraavien piilokerrosten neuroneiden ulostuloarvot lasketaan kaavoilla:

$$s_k = \sum_{j=1}^h (w_{jk}S'_j), \quad k = 1,2,3, \dots, m \quad (39. a)$$

$$S_k = \text{TanH}(s_k) = \frac{e^2 - e^{-s_k}}{e^2 + e^{-s_k}}, \quad k = 1,2,3, \dots, m \quad (39. b)$$

jossa S_k on piiloneuronin ulostuloarvo. S'_j on edellisen piilokerroksen j :nen neuronin ulostuloarvoa ja w_{jk} kuvaa kytkennän kerrointa edellisen kerroksen j :nen ja nykyisen kerroksen k :nen neuronin välillä. Nykyisen piilokerroksen neuroneiden määrä on m , eli tämän tutkimuksen kokeessa h :n tapaan $m = 21$, sillä kyseisen neuroverkon kaikki piilokerrokset ovat saman levyisiä.

Neuroverkon ulostulo lasketaan kaavoilla:

$$o_l = \sum_{k=1}^m (w_{kl}S_k), \quad l = 1,2,3, \dots, p \quad (40. a)$$

$$O_l = \text{TanH}(o_l) = \frac{e^2 - e^{-o_l}}{e^2 + e^{-o_l}}, \quad l = 1, 2, 3, \dots, p \quad (40. b)$$

jossa O_l on ulostulokerroksen neuronin ulostuloarvo p :n ollessa kyseisen kerroksen neuroneiden määrä, eli tämän tutkimuksen neuroverkon tapauksessa $p = 4$. w_{jk} kuvaa kytkentöjen kertoimia edeltävän piilokerroksen j :nen neuronin ja ulostulokerroksen k :nen neuronin välillä.

Neuroverkon ulostuloarvojen O_l perusteella määritetään lentomallissa käytettävät ohjainmuuttajat siivekkeille ja korkeusperäsimmelle. Tekoälylle ei anneta mahdollisuutta ohjata sivupe-räsintä, sillä sen merkitys simulaatiossa on suhteellisen vähäinen ja neuroverkon tehtävä halutaan pitää mahdollisimman yksinkertaisena. Myös teho asetus on samasta syystä määritetty kiinteästi maksimaaliselle teholle, sillä kaartotaistelussa hyvin harvoin on tarve pienemmälle tehoasetukselle.

Kaksi ensimmäistä ulostuloarvoa O_1 ja O_2 määrittävät neuroverkon halun kallistaa ilma-alusta siivekkeiden avulla vastaavasti oikealle ja vasemmalle. Näiden erotuksesta saadaan kohdearvo siivekkeasetukselle. Siivekkeet liikkuvat simulaation edetessä, kohti kohdearvoa taulukon 2 mukaisin perustein. Samalla perusteella määritetään korkeusperäsimen asento kahden viimeisen ulostuloarvon O_3 ja O_4 perusteella.

Kyt kentöjen kertoimet w määritetään aluksi satunnaisesti $w \in \mathbb{R}, w \in [-1, 1]$. Geneettisellä algoritmilla pyritään eräajon edetessä säätämään populaation kunkin neuroverkon kertoimia siten, että yksittäisten neuroverkkojen ja koko populaation suoritustaso kaartotaistelusimulaatiossa paranee.

3.1.4. Geneettisen algoritmin parametrit

Kuten luvussa 2.3.1 kerrottiin, voidaan geneettisen algoritmin toiminta kuvata vaiheina, jotka ovat: alustus, valinta, risteytys ja mutaatio. Alustusvaiheessa luodaan populaatio, joka pitää sisällään 80 listaa kertoimista w , jotka on luotu edellisessä kappaleessa kuvatun neuroverkon parametrien mukaisesti. Populaation voidaan siis ajatella pitävän sisällään 80 neuroverkkoa.

Ennen valintavaihetta populaation neuroverkot kilpailevat pareittain toisiaan vastaan kaartotaistelusimulaatiossa, ja kukin pisteetetään jokaisen mallin iteraation eli aikahypyn jälkeen taulukon 8 mukaisesti. Pisteytysmenetelmästä voidaan käyttää nimeä kelpoisuusfunktio ja neuroverkon tehtävä on liikehtiä saavuttaakseen parhaat mahdolliset kelpoisuusfunktion tuottamat pisteet.

Taulukko 8. Pisteytys neuroverkkojen suorituskyvyn arvioimiseksi eli kelpoisuusfunktiot.

Tilanne	Pisteet (p)	Huomio
Tähtäysvirhe (ATA as- teina)	$((\pi-ATA)/\pi)^2$	jos $p < 0$, $p = 0$
Vastustajan siipilinjan ta- kana ja vastustaja oman siipilinjan edessä	1 - etäisyys(ft) /6000 ft	jos $p < 0$, $p = 0$
Tähtäysvirhe < 500 ft	1	
Tähtäysvirhe < 100 ft	10	
Tähtäysvirhe < 15 ft	100	
Korkeus < 0 ft	0 (vastustajalle + 100)*	Simulaatio keskeytetään
Kohtauskulma > 55.0°	-1	
Kohtauskulma < -55.0°	-1	Tavoitteena välttää epätavalliset lentotilat ja simulaation hajoaminen mallin taulukoiden ulkopuolella
Sivuluisukulma > 20.0°	-1	
Sivuluisukulma > -20.0°	-1	
Epätavallinen lentotila	-100 (vastustajalle +100)	Simulaatio keskeytetään
Normaalikiikkyvyys > 9,0G	-1	F-16 rajoitus

*Ohjelmointivirheen vuoksi maahan törmäämisestä ei rangaista. Simulaatio kuitenkin keskeytyy ennenaikaisesti, minkä vuoksi pisteiden saaminen lakkaa maahan törmätessä ennen neljän minuutin aikarajaa.

Pisteytyksen tarkoituksena on ohjata geneettistä algoritmia suosimaan yksilöitä, jotka liikehtivät siten, että tähtäysvirhe vastustajaan on mahdollisimman pieni ja asema suhteessa vastustajaan on tämän lähellä siipilinjan takana, eli hallinta-asemassa. Lisäksi maan pintaan törmääviä, kohtauskulma- ja kiihtyvyyksirajoja rikkovia yksilöitä rangaistaan. Taulukossa 8 epätavallisella lentotilalla tarkoitetaan lentotilaa, jossa hävittäjän tilamuuttujat päätyvät mallissa käytettävien taulukoiden ulkopuolelle eikä lentotilaa enää kyetä vakiinnuttamaan. Kelpoisuusfunktio ajetaan simulaation jokisen aikahypyn jälkeen ja kelpoisuus pisteet summataan kumulatiivisesti. Pisteytyksen perusteella geneettinen algoritmi valitsee parhaiten suoriutuvat yksilöt todennäköisimmin seuraavan sukupolven vanhemmiksi.

Pisteytystä hyödynnetään valintavaiheen lisäksi populaation suorituskyvyn kehittymisen arvioimisessa. Huomioon on kuitenkin otettava populaation yksilöiden kilpaileminen toisiaan vastaan, ja vaikka kyseessä ei olekaan nollasummapelejä, vaikuttaa vastustajan tehokas liikehtely merkittävästi yksilön mahdollisiin pisteisiin.

Eräajossa käytettävä geneettinen algoritmi hyödyntää luvussa 2.3.1.2 esiteltyä rulettipyörämenetelmää valintavaiheessa, eli kukin yksilö saa simulaation jälkeen populaation kokonaispisteisiin suhteutettuja pisteitään vastaavan todennäköisyyden tulla valituksi. Rulettipyörämenetelmällä valitaan yhteensä 70 paria seuraavan sukupolven vanhemmiksi. Sama yksilö voi tulla valituksi useammin kuin kerran. Rulettipyörämenetelmän lisäksi valinnassa hyödynnetään elitismii, eli tässä tapauksessa populaation 10 parasta yksilöä siirtyvät suoraan seuraavaan sukupolveen ilman risteytystä tai mutaatiota. Elitismillä pyritään estämään populaation suorituskyvyn tahaton heikentyminen.

Seuraavaksi risteytysvaiheessa 70 valitusta parista muodostetaan risteyttämällä 70 uutta yksilöä. Uuden yksilön kukin kerroin valitaan satunnaisesti vanhempien vastaavien kertoimien väliltä. Lisäksi 1% todennäköisyydellä kyseessä oleva kerroin mutatoituu, eli saa uuden satunnaisen arvon $w \in \mathbb{R}, w \in [-1,1]$.

Tämän jälkeen simulaatio ajetaan uuden sukupolven populaatiolle ja vaiheet valinnasta, risteytyksen kautta mutaatioon toistetaan, kunnes käytettävissä oleva laskenta-aika täyttyy.

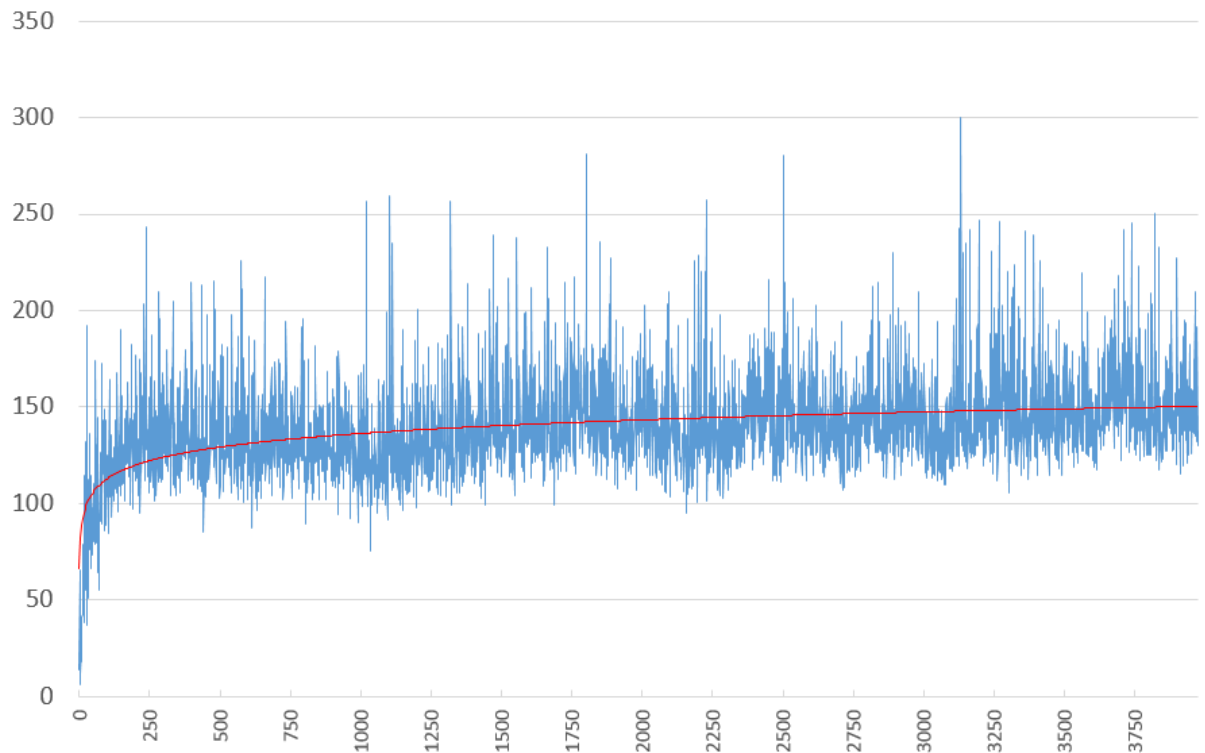
4. ERÄAJOSIMULAATIOIDEN TULOKSET

Eräajot suoritettiin Laskentakeskuksen Puhti-supertietokoneella 3.11.2021 – 10.11.2021 välisenä aikana yhtäjaksoisesti seitsemän vuorokauden ajan. Tänä aikana laskentaa suoritettiin 3 974 sukupolven verran, eli yhteensä 158 960 kaartotaistelukohtaamista. Eräajon tulokset ovat saatavissa kokonaisuudessaan liitteen 4 tiedostoina ”savBN.txt”, ”savB.txt”, ”savA.txt”, ”savP.txt”, jotka pitävät sisällään kunkin sukupolven parhaan neuroverkkoyksilön, parhaan yksilön pisteet, populaation pisteiden keskiarvon sekä eräajon viimeisen sukupolven populaation kokonaisuudessaan. Seuraavissa alaluvuissa esitellään eräajojen tulokset ja analyysi tuloksista.

4.1. Eräajon tulosten esittely ja analyysi

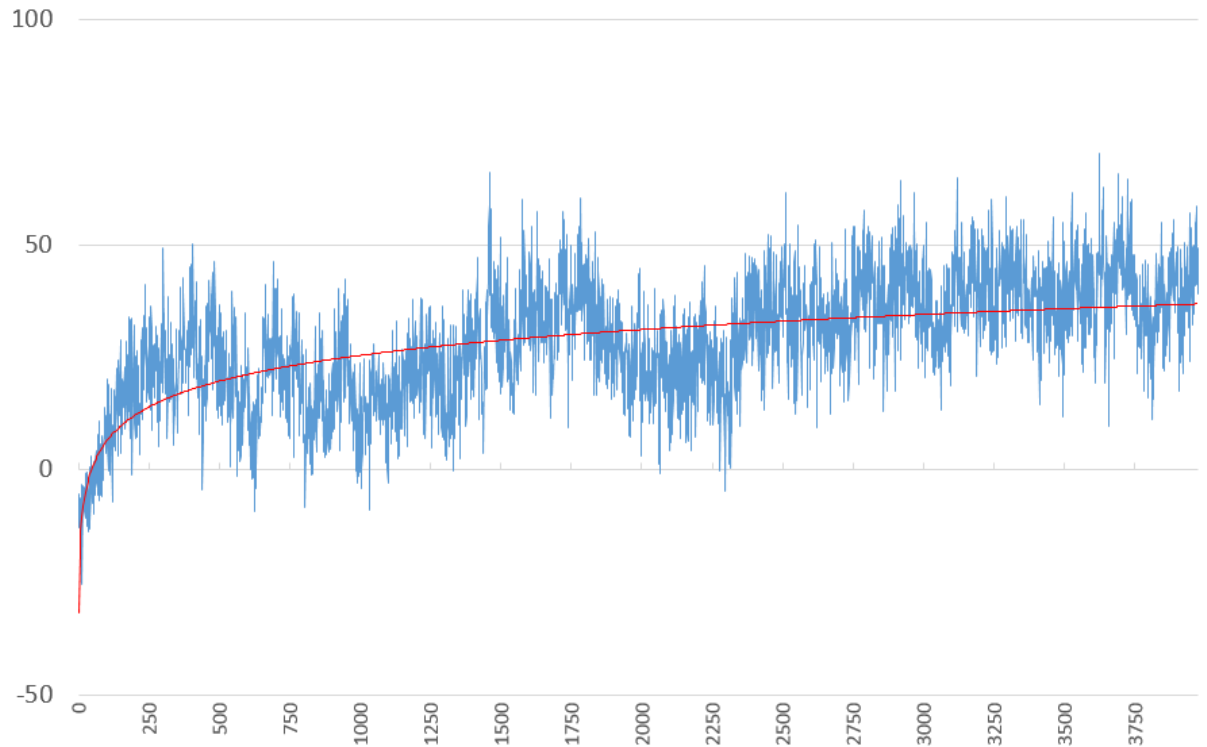
Eräajon tuloksien analysoinnissa hyödynnetään samaa kelpoisuusfunktioita (taulukko 8), jota tutkimuksessa käytettävä geneettinen algoritmi käyttää valitessaan yksilöitä seuraavan sukupolven vanhemmiksi. Kelpoisuusfunktio palkitsee yksilöitä, jotka pyrkivät osoittamaan ilma-aluksensa keulalla vastustajaa ja liikehtimään vastustajan siipilinjan taakse. Koska populaatio kokonaisuutena kehittyy eräajon aikana, muuttuu tekoälyn tehtävä jatkuvasti vaikeammaksi ja pisteiden saaminen on näin ollen haastavampaa.

Paras mahdollinen tulos ilman alle 500 jalan tähtäysvirhettä on noin 200. Mikäli ilma-alus saavuttaa taulukon 8 mukaiset tähtäysvirheet alle 500, 100 tai 15 jalkaa jossakin simulaation vaiheessa, on paras mahdollinen tulos merkittävästi suurempi. Teoreettinen maksimipistemäärä on 240 480, mutta tämä vaatisi täydellistä tähtäystä vastustajaan koko neljän minuutin kaartotaistelun ajan 0 jalan etäisyydeltä vastustajan siipilinjan takana. Teoreettista huipputulosta ei siis ole käytännössä mahdollista saavuttaa, sillä jo simuloitun kaartotaistelun aloitustilanne, jossa ilma-alukset ovat vastakkaisilla lentosuunnilla 1000 jalan sivuttaisetäisyydellä, estää täydellisen tähtäämisen simulaation alusta lähtien. Lisäksi molemmat osapuolet pyrkivät jatkuvasti liikehtimään parempaan asemaan, mikä samalla vaikeuttaa toisen osapuolen tehtävää. Yli 980 pisteen tulos kertoo varmasti kumulatiivisesta alle 15 jalan tähtäysvirheestä yli viiden sekunnin ajan, eli ilma-aluksella olisi tällöin ollut mahdollisuus ampua vastustajaa konetykillä.



Kuva 18. Kunkin sukupolven populaation parhaan yksilön pisteet.

Kuva 18 esittää kunkin sukupolven parhaan yksilön saamia pisteitä sinisellä käyrällä. Pystyakselilla esitetään pisteet ja vaaka-akseli kuvaa sukupolvia eräajojen edetessä. Punainen käyrä on logaritminen trendiviiva, jota tarkasteltava muuttuja näyttää seuraavan. Kuvan perusteella voidaan todeta, että sukupolvien edetessä kehitys on ollut aluksi hyvin nopeaa, minkä jälkeen kehitys hidastuu merkittävästi, jatkuen kuitenkin viimeiseen sukupolveen asti. Kehitys ei ole pysähtynyt, joten oletettavasti kehityspotentiaalia on vielä jäljellä, mikäli eräajoja jatketaan pidemmälle. Kuvan pistejakaumasta voidaan päätellä tekoälyn oppineen liikehtimään tarkoituksenmukaisesti suhteessa vastustajaan ja välttää kohtauskulma- ja kuormituskerroinrajoitusten rikkomista sekä maahan törmäämistä. Tekoäly on ajoittain kyennyt hetkittäisiin tilanteisiin pienellä tähtäysvirheellä, mutta jatkuvia tarkkoja tähtäyksiä ei ole havaittavissa. Pearsonin korrelaatiokerroin [17, s. 188] koko muuttujien populaatiolle on 0,33. Sukupolvivälillä 0-200 korrelaatiokertoimeksi saadaan 0,65. Molemmat kertoimet viittaavat kohtalaiseen korrelaatioon tekoälyn suoritustason ja sukupolven välillä. Kuvassa 18 esitetty data on saatavissa kokonaisuudessaan liitteen 4 tiedostona ”savB.txt”



Kuva 19. Kunkin sukupolven populaation pisteiden keskiarvo.

Kuva 19 esittää kunkin sukupolven populaation pisteiden keskiarvoja samoin periaattein kuin kuvassa 18. Kuvan perusteella voidaan todeta populaation kehityskäyrän olevan vielä 3 974 sukupolvenkin jälkeen melko jyrkkä, ja keskiarvopisteet jäävät vielä selkeästi parhaiden yksilöiden pisteistä (kuva 18). Tämän perusteella voidaan olettaa tekoälyllä olevan vielä kehityspotentiaalia jäljellä erityisesti populaatiotasolla, mikäli eräajoja jatketaan pidemmälle. Kuvassa 19 esitetty data on saatavissa kokonaisuudessaan numeerisesti liitteen 4 tiedostona ”savA.txt”

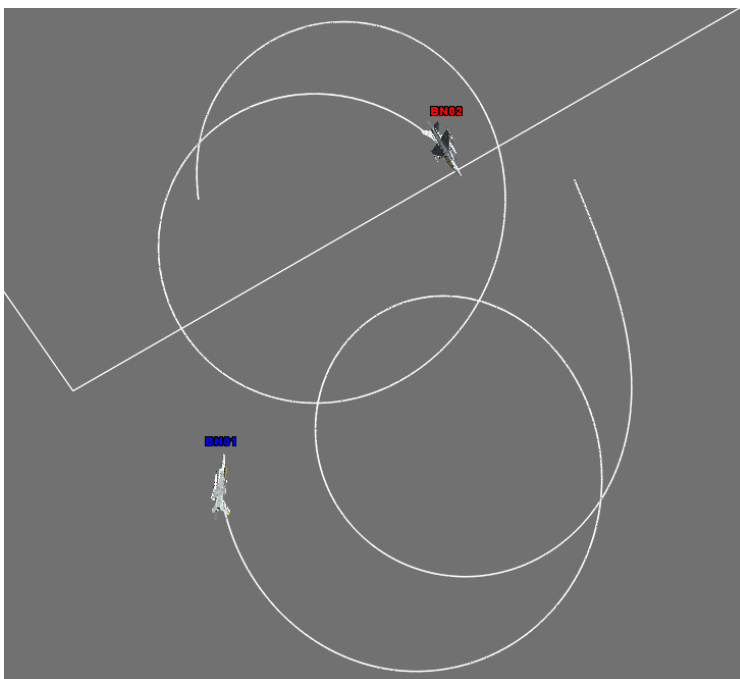
Taulukko 9. Sukupolven 3973 paras yksilö muita sukupolvia vastaan.

Sin.sukupolvi	Sin. Pisteet	Pun #3973 pisteet	Huomio
#1	-0.82	13.92	Simulaatio keskeytyy sinisen törmätessä maahan aikaan 24.23s
#10	-100.00	184.57	Simulaatio keskeytyy sinisen epätavallisen lentotilan vuoksi aikaan 153.96s
#100	27.45	49.68	Simulaatio keskeytyy sinisen törmätessä maahan aikaan 146.92s
#1000	108.86	103.45	
#3974	113.41	93.87	

Kehityksen arvioimiseksi eräajojen jälkeen sukupolvien 1,10,100,1 000 ja 3 974 parhaat yksilöt (sininen) ajettiin sukupolven 3 973 parasta yksilöä (punainen) vastaan. Näiden ajojen tulokset ovat nähtävissä taulukosta 9. Sukupolvien 1,10 ja 100 yksilöt törmäsivät maahan tai päätyivät epätavalliseen lentotilaan, jolloin simulaatiot keskeytyivät. Sukupolvien 1 000 ja 3 974 parhaat yksilöt kykenivät pisteiden osalta jopa hieman parempaan suoritukseen kuin 3 973 sukupolven paras yksilö. Tämän tarkastelun perusteella tekoälyn voidaan todeta kehittyvän eräajojen myötä.

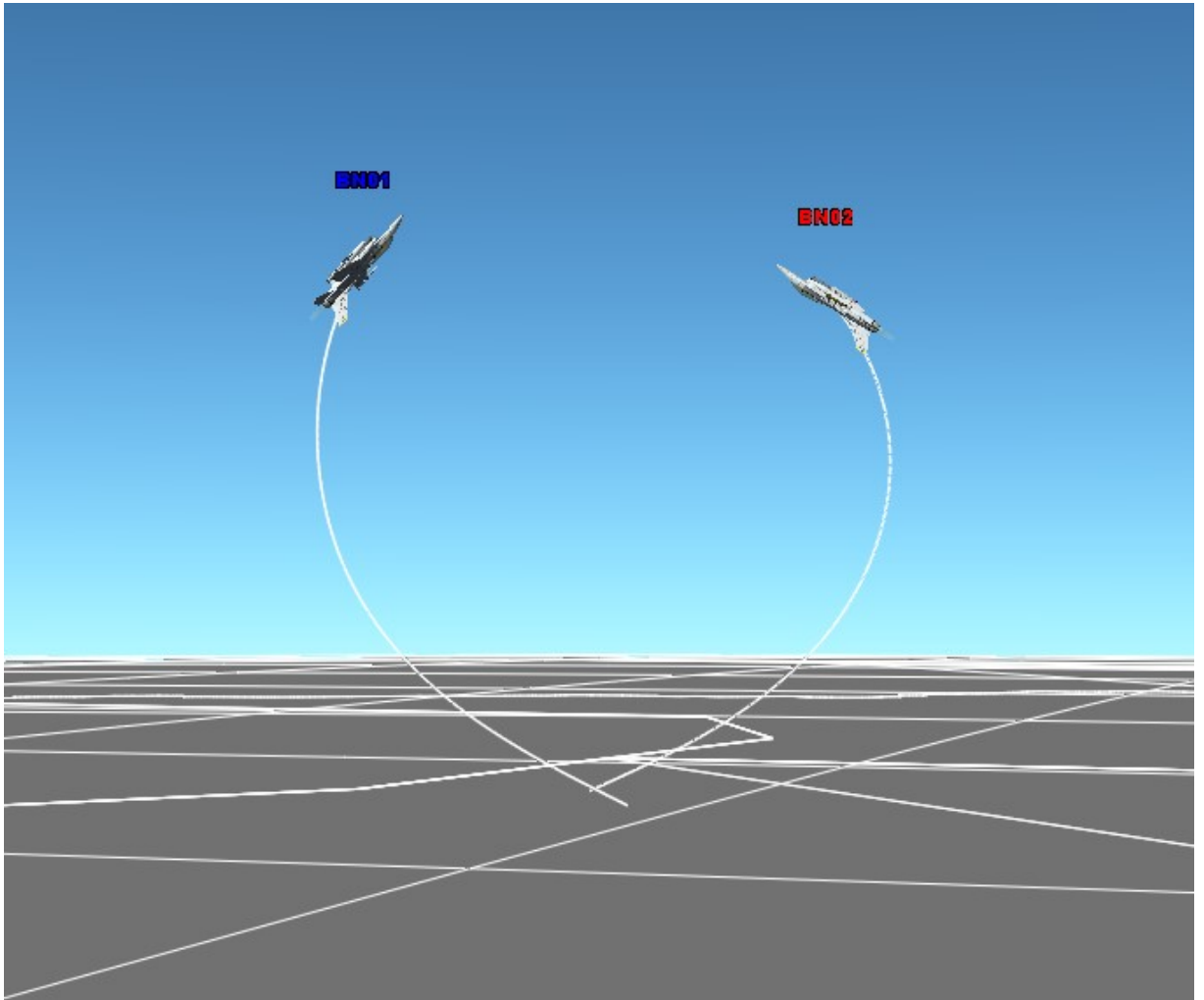
Numeerisen tarkastelun lisäksi simulaatioita on tarkasteltu visuaalisesti TADS-ohjelmalla, joka on Ilmavoimissa käytössä oleva tehtävienpurku-työkalu. Ohjelma mahdollistaa lentotehtävän tarkastelun jälkikäteen kolmiulotteisessa virtuaaliympäristössä.

Visuaalisesti tarkasteltuna tutkijan oman arvion mukaan liikehtely tehostuu huomattavasti sukupolvien edetessä. Sukupolven 3 973 parhaan yksilön liikehtely näyttää tarkoituksenmukaisemmalta, kun vastustajana on kyvykäs yksilö. Vähemmän koulutettua yksilö vastaan liikehtelyssä on puutteita, eikä tekoäly osaa hyödyntää vastustajan tekemiä virheitä. Tämä on osaltaan merkki tekoälyn ylisovittumisesta suoriutumaan juuri kyseisen sukupolven populaatiota vastaan. Myös taulukon 9 sarakkeen ”Pun #3973 pisteet” numeeriset arvot viittaavat yksilön #3973 kelpoisuuden paranevan, kun vastustaja on kelpoisempi. Visuaalisen tarkastelun perusteella voidaan todeta tekoälyn kykenevän tyypillisimpiin helppoihin kaartotaisteluliikkeisiin yhden ympyrän ja kahden ympyrän taisteluissa.



Kuva 20. Hävittäjät kahden ympyrän vaakakaarrossa #3973 vs. #3974 (kuvankaappaus TADS).

Kahden ympyrän taistelulla (kuva 20) tarkoitetaan tilannetta, jossa hävittäjät kaartavat ympyrän muotoisilla lentoradoilla samansuuntaisesti. Tällaisessa kaartotaistelutilanteessa tärkeintä on optimoida hävittäjän kulmanopeus ja näin saavuta etulyöntiasema vastustajaan nähden. Ympyrät voivat olla maantieteellisesti päällekkäin, erillään tai lomittain, kuten kuvassa 20.

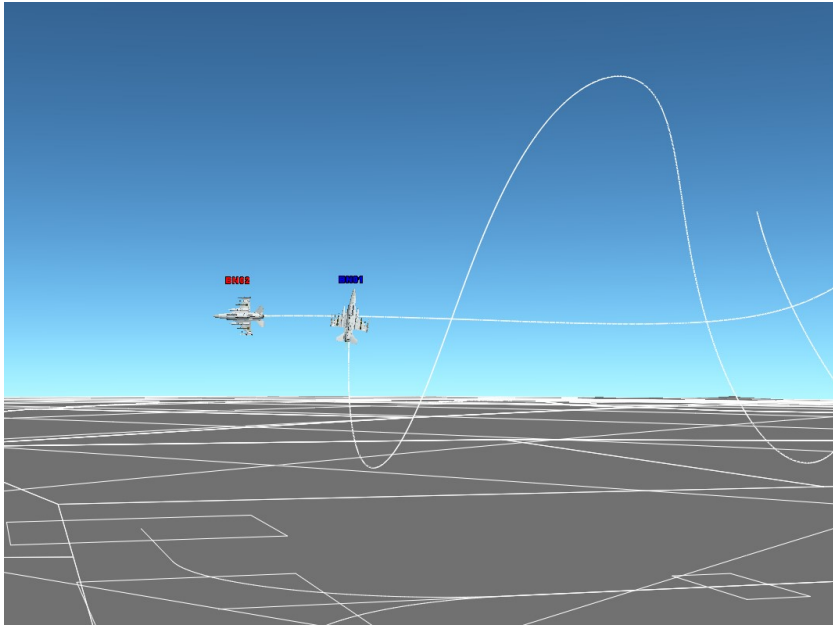


Kuva 21. Hävittäjät yhden ympyrän silmukkataistelussa #3973 vs. #3974 (kuvankaappaus TADS).

Yhden ympyrän taistelussa (kuva 21) hävittäjät kiertävät ympyrän muotoisia ratoja vastakkaisiin suuntiin. Tässä tilanteessa pyritään minimoimaan kaartosäde ja näin pääsemään vastustajan kaartosäteen sisäpuolelle edulliseen asemaan.

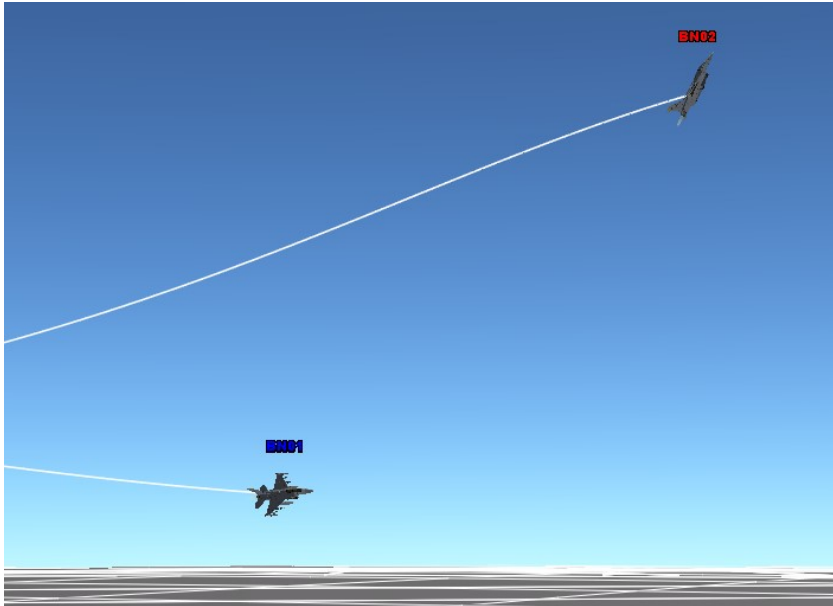
Erilaisilla kaartotaisteluliikkeillä pyritään saavuttamaan oikea-aikainen kulmanopeuden tai kaartosäteen optimointi kuhunkin taistelutilanteeseen, jotta saavutetaan etulyöntiasema vastustajaan nähden. Visuaalisen tarkastelun perusteella tekoälyn liikehtelystä on tunnistettavissa tyypillisiä helppoja liikkeitä kuten silmukka, tynnyri ja sakset.

Silmukalla tarkoitetaan pystysuuntaista ympyrän muotoista liikettä ja sen etuja ovat mm. ylöspäin suuntautuvalla osalla hidastuva vauhti, jonka seurauksena kaartosäde pienenee. Erityisesti alaspäin suuntautuvalla liikkeen osalla pystyasennosta hävittäjä on helposti ja nopeasti kääntävissä mihin tahansa ilmansuuntaan kohti vastustajan hallinta-asemaa. Kuvassa 21 molemmat hävittäjät ovat aloittaneet pystyliikehtelyn ja ovat silmukoidensa nousevilla osilla.



Kuva 22. Sininen hävittäjä ohiajattaa punaisen hävittäjän tynnyriliikkeellä #3973 vs. #3974 (kuvankaappaus TADS).

Tynnyri lennetään tyypillisesti tilanteessa, jossa hävittäjän nopeus ei enää riitä täydelliseen silmukkaan pystysuunnassa. Tämän seurauksena liike on taitettava horisonttiin nähden viistoon kulmaan, jotta vältytään liian pieneltä ilmanopeudelta ja ohjaamattomalta lentotilalta. Tynnyri mahdollistaa hävittäjän tehokkaan suuntaamisen ilman maantieteellisen paikan suurta muutosta. Kuvassa 22 punainen hävittäjä kaartaa horisonttiin nähden vaakatasossa. Samaan aikaan sininen hävittäjä on suorittanut tynnyriliikkeen punaisen hävittäjän lentoradan ympäri. Sininen hävittäjä on näin ollen ohiajattanut punaisen ja saavuttanut edullisen aseman tämän siipilinjan takana.



Kuva 23. Punainen hävittäjä saksissa pyrkii ohiajattamaan sinisen hävittäjän (kuvankaappaus TADS).

Saksilla puolestaan tarkoitetaan liikettä, jossa minimoidaan ilma-aluksen nopeus ja eteenpäin suuntautuva liike. Saksilla pyritään ohiajattamaan vastustaja ja näin pääsemään tämän siipilinjaa taakse edulliseen asemaan. Kuvassa 23 punainen hävittäjä lentää pienellä nopeudella ja suurella kohtauskulmalla minimoidakseen maantieteellisen liikkeensä yrittäessään ohiajattaa sinistä hävittäjää, joka liikehti edelleen suurella nopeudella vaakatasossa.

Visuaalisessa tarkastelussa ei havaittu tilanteita, joissa hävittäjät olisivat onnistuneet tähtäämään vastustajaa kohti tarkasti. Yhtä tilannetta lukuun ottamatta ei saavutettu vakiintunutta hallinta-asemaa. Lisäksi tekoälyn ohjaamien hävittäjien liikehtelyn suuntaaminen oli puutteellista.

Tutkijan omat havainnot visuaalisesta tarkastelusta korreloivat hyvin kuvan 18 kehityskäyrän ja taulukon 9 pisteytyksen kanssa. Iteraatioiden myötä tekoäly kehittyi liikehtimään merkittävästi tehokkaammin ja erityisesti välttämään virheliikkeitä ja maahan törmäämistä. Iteraatioiden lopputuloksena luotu sukupolven 3 874 paras yksilö osaa liikehtiä tehokkaasti taistelun aloituksessa ja kahden ympyrän taistelussa. Liikkeiden suuntaamisessa suhteessa vastustajaan ja taistelutilanteen etenemiseen on suuria puutteita. Tämä näkyy epäloogisina liikehtelygeometrian suuntina, minkä seurauksena hävittäjät eivät kykene vakiinnuttamaan paikkaansa hallinta-asemassa suhteessa vastustajaan.

4.2. Asiantuntija-arviot

Asiantuntijoille on esitetty taulukon 9 mukaiset kaartotaistelusimulaatiot visuaalisesti TADS-tehtävänpurkujärjestelmällä. Asiantuntijoita on pyydetty arvioimaan tekoälyn ohjaamien hävittäjien suoriutumista kaartotaistelussa numeerisesti 1-5 välillä (1 huonoin ja 5 paras arvio). Lisäksi asiantuntijat arvioivat tekoälyn suoritusta sanallisesti kuten arvioisivat oikeaa lento-oppilasta lentotehtävän jälkeen. Asiantuntijoita on pyydetty asettamaan suoritukset paremmuusjärjestykseen. Asiantuntijat eivät tienneet etukäteen heille esitettyihin simulaatioihin osallistuvien neuroverkkoyksilöiden kyvykkyyttä tai sukupolvea. Asiantuntijoina on käytetty Ilmavoimien hävittäjälentolaivue 11:n ilmataisteluopettajia.

Taulukko 10. Sukupolven 3 973 paras yksilö muita sukupolvia (sin.) vastaan – Ilmataisteluopettajien antamien suoritusarviointien keskiarvo.

Sin.sukupolvi	Sin. KA. Pisteet	Huomio
#1	0,9*	Simulaatio keskeytyy sinisen törmätessä maahan aikaan 24.23
#10	1,9	Simulaatio keskeytyy sinisen epätavallisen lentotilan vuoksi aikaan 153.96
#100	1,4	Simulaatio keskeytyy sinisen törmätessä maahan aikaan 146.92
#1000	2,6	
#3974	3,4	

* Suoritusarvion arvo ”1-” tulkitaan numeerisesti 0,75, minkä vuoksi keskiarvo jää arvon 1,0 alle.

Asiantuntija-arvioita kerättiin yhteensä neljä ja ne ovat saatavissa kokonaisuudessaan liitteessä 3. Kaikki asiantuntijat arvioivat heille esitettyjen tekoäly-yksilöiden paremmuusjärjestyksen seuraavasti: #3974, #1000, #10, #100, #1 (paras ensimmäisenä). Yksilön #10 arvioiminen yksilöä #100 paremmaksi selittyy yksilön #100 maahan törmäämisellä, jota asiantuntijat pitävät merkittävästi suurempana virheenä kuin yksilön #10 epätavallista lentotilaa, johon simulaatio katkeaa. Asiantuntijoiden arvio paremmuusjärjestyksestä sekä taulukon 10 keskiarvopisteet korreloivat hyvin edellisen alaluvun tulosten kanssa, mikä yhdessä kuvissa 18 ja 19 esitettyjen suoritusasokäyrien kanssa osoittavat tekoälyn kehittyvät paremmaksi tehtävässään iteraatioiden myötä.

Asiantuntijoiden sanalliset arviot ovat yhteneviä numeeristen arvioiden sekä paremmuusjärjestyksen kanssa, mikä tukee näkemystä tekoälyn kehityksestä iterointien myötä. Ensimmäisen sukupolven parhaasta yksilöstä #1 (peitenimeltään ”Markku”) sanotaan seuraavaa:

” Vaakakierre ei ole taistelu- vaan taitolentoliike. Aloituksesta kuuluu lähteä 1 tai 2 ympyrään, ei suoraan alas ja törmätä maahan. Toki maali varmasti ampui meidät ennen törmäystä. ”

” Markku oli löytänyt vain poikittaisohjauksen. Takaisin alkeiskoulutukseen. ”

” Lensi suoraa maahan ”

” -3x Ail roll, decent into ground”

Yksilön #1 saamat sanalliset arviot ovat sävyllään negatiivisia, mikä on odotettua, sillä kyseessä on ensimmäisen sukupolven yksilö, jonka parametrit ovat täysin satunnaisia eikä niitä ole vielä optimoitu.

Yksilön #1 saamaan palautteeseen verrattuna viimeisen sukupolven parhaan yksilön #3974 (”Antti”) saama palaute on huomattavasti rohkaisevampaa, vaikka asiantuntijat arvioivatkin toimintaa hyvin kriittisesti pitäen vertailukohtana oikeita hävittäjäohjaajaoppilaita. Yksilöstä #3974 sanotaan mm. seuraavaa:

” Alussa pitää vetää tiukemmin pystyyn. Punaiselle annettiin turhaan 2 ammuntopaikkaa. Tämän jälkeen liikehdintä liikaa leadille -> ajaututtiin punaisen eteen. Lopussa päästiin lag:n / oikean ennakon kautta lähes hallinta-asemaan. Tee se aiemmin. ”

” Liikkeiden suuntaaminen heikkoa. Jäi useassa kohdassa maalin etusektoriin. Ennakkokaarrot puutteellisia. Hakeutui pääsääntöisesti 2 ympyrän geometrioihin ja lopulta pääsi hallinta-asemaan. ”

” Antti suosi 2-ymp. taktiikkaa. Pystysuunnassa liikkeen suuntaaminen oikein tuottaa haasteita. 2-ymp. -ennakkokaarto onnistuu satunnaisesti. ”

Yllä esitettyjen sanallisten arvioiden perusteella voidaan todeta yksilön #3974 toiminnassa olevan edelleen merkittäviä puutteita verrattuna oikeaan hävittäjäohjaajaan. Palaute koskee kuitenkin kaartotaistelun suoritustekniikoita pitäen sisällään myös muutamia positiivisia kommentteja. Yksilön #1 tapauksessa moitteet kohdistuivat lentoturvallisuuteen. Tämän perusteella voidaan todeta tekoälyn kehittyneen liikehtimään turvallisesti, vaikka taktisessa tehokkuudessa on vielä puutteita.

Asiantuntijoita pyydettiin lisäksi arvioimana yksilön #3973 (”Matti”) toimintaa. #3973 toimi vastustajana kaikissa visuaalisen tarkastelun kaartotaistelukohtaamisissa. Yksilöstä #3973 sanotaan seuraavaa:

”Aloitukset tehokkaita ja tiukkoja. Parempaa suuntaamista pääsääntöisesti kuin muilla, mutta ei ratkaisuhalukuutta: Aina stalemate.”

” Matin käytöksessä ei merkittäviä muutoksia eri vastustajien välillä, mutta kohtaamisten sisällä toiminta ja geometrianhallinta muuttuivat järkevistä surkeiksi kuten sinisellä.”

” Vastustajan vaihtuessa Matti ei osannut soveltaa liikehtelyä eikä hyödyntää vastustajan virheitä vaan lensi ennalta oppimiaan temppuja.”

” Failed to capitalize on blue mistakes. LV placement. Lost to Antti after 3 min, so okay until then.

Näiden sanallisten arvioiden perusteella yksilö #3973 ei kyennyt ratkaisemaan taistelua, vaikka osa kaartotaisteluista tapahtuikin selkeästi vähemmän koulutusta saaneita yksilöitä vastaan. Tämä on osaltaan merkki tekoälyn ylisovittumisesta mihin erityisesti kommentti *”... Matti ei osannut soveltaa liikehtelyä eikä hyödyntää vastustajan virheitä vaan lensi ennalta oppimiaan temppuja.”* viittaa. Asiantuntijoiden sanalliset arviota ovat saatavissa kokonaisuudessaan liitteessä 3.

5. POHDINTAA

Tässä tutkimuksessa geneettiselle algoritmille asetettu optimointitehtävä on erittäin haastava, sillä optimoitavia parametreja ovat tutkimuksen neuroverkon kertoimet, joita on 1512. Tämän seurauksena etsintäavaruus on 1512 ulotteinen hyperkuutio, josta on pyrittävä etsimään optimaalinen piste $3024 \cdot 10^{18}$ pisteen joukosta. Tämä ei kuitenkaan yksiselitteisesti ole mahdollista, sillä ongelman luonteesta johtuen etsintäavaruus on dynaaminen ja simulaation edetessä optimaalinen ratkaisu vaihtaa jatkuvasti paikkaansa etsintäavaruudessa. Tutkimuksessa käytetty myötäsyöttöneuroverkko on luonteeltaan staattinen, joten se ei kykene seuraamaan simulaation aikana liikkuvaa optimiratkaisua, vaan sen on tyydyttävä globaaliin kompromissiin.

Neuroverkon ja geneettisen algoritmin teorialuvussa kuvattuja hyperparametreja muuttamalla ja testaamalla on mahdollista parantaa tekoälyn suorituskykyä ja oppimisnopeutta jonkin verran. Edellisessä kappaleessa mainittujen seikkojen lisäksi muistin puute ja siitä johtuva kykenemättömyys tilanteen kehittymisen arviointiin ovat merkittäviä haasteita, joiden ratkaiseminen vaatii suuria muutoksia tekoälyalgoritmin rakenteeseen.

Vaihtoehtoja tekoälyalgoritmin suorituskyvyn parantamiseen on useita. Neuroverkko on esimerkiksi mahdollista rakentaa modulaariseksi käyttäen useita toisiinsa kytkettyjä verkkoja. Tällaisessa verkkorakenteessa ensimmäiselle verkolle voidaan antaa tehtäväksi eri tyyppisten taistelutilanteiden tunnistaminen. Tunnistamisen jälkeen verkko voi valita tilanteeseen sopivimman aliverkon, joka on optimoitu juuri kyseisen kaltaiseen taistelutilanteeseen. Erilliselle neuroverkolle voidaan myös antaa tehtäväksi lopputuloksen arvioiminen hetkittäisen tilanteen perusteella, mikä mahdollistaa erilaisten toimintavaihtoehtojen reaaliaikaisen arvioimisen. Lisäksi verkkorakenteeseen on mahdollista luoda takaisinkytkentöjä ja niihin neuroneita, jotka ajavat muistin asemaa. Tämä aikadimensio antaa neuroverkolle kattavamman tilannekuvan päätöksentekoa varten, ja todennäköisesti parantaa verkon suorituskykyä. Tämän kaltainen kehittynyt verkkorakenne vastaa osittain dynaamisen etsintäavaruuden aiheuttamaan haasteeseen.

Neuroverkkorakenteen muuttamisen lisäksi on mahdollista muuttaa koulutusmenetelmää tai käyttää erilaisten koulutusmenetelmien yhdistelmiä. Tässä tutkimuksessa koulutusmenetelmänä käytettiin vahvistusoppimista, joka toteutettiin geneettisellä algoritmilla. Geneettinen algoritmi pyrki optimoimaan neuroverkon kertoimia vain sukupolvien välillä, eli neuroverkot ovat staattisia simulaatioiden aikana. Ohjatun oppimisen menetelmiin kuuluvaa takaisinvirtausfunktioita [12] voidaan käyttää myös simulaation aikana, mikäli ongelmaa riittävän tarkasti kuvaava kelpoisuusfunktio kyetään luomaan. Takaisinvirtausfunktio voi analysoida verkon suoritusta simulaation jokaisen aikahypyn aikana ja hienosäätää ohjaajana käytettävän neuroverkon kertoimia. Onnistuneesti toteutettuna neuroverkko kykenee tällöin seuraamaan etsintäavaruudessa liikkuvaa optimaalista ratkaisua.

Yleinen haaste neuroverkkojen hyödyntämisessä käytännön sovelluksissa on kattavan laadukkaan datan puute. Sama ongelma on läsnä tämän tutkimuksen viitekehyksessä, sillä todelliseen maailmaan siirrettynä tämän tutkimuksen skenaariosta tulee hyvin haasteellinen. Haasteita ovat muun muassa omien sekä vastustajan lentoarvojen mittaamisen toteuttaminen sekä mittaustulosten häiriöt ja katveet. Toisin sanoen simulaatiomaailmassa syötedata on puhdasta ja kattavaa, kun todellisessa maailmassa se on puutteellista ja siinä on kohinaa. Datan puutteita ja kohinaa on toki mahdollista simuloida, mutta tämän pitäisi vastata todellista maailmaa liki täydellisesti, jotta vältetään ylisovittumisen ongelmalta neuroverkkoa koulutettaessa. Todelliseen maailmaan tämän tutkimuksen koetta ei voida siirtää valtaviin resurssivaatimusten vuoksi: kokeessa simuloitiin 317 920 lentosuoritusta. Niissä hävittäjä tuhoutui maahan törmäämisen tai epätavallisen lentotilan vuoksi 29 645 kertaa, mikä on täysin kestämatöntä todellisia hävittäjiä käytettäessä. Taloudellisin tapa lienee kouluttaa tekoälyä synteettisellä datalla simulaatioympäristössä tasolle, joka mahdollistaa koulutuksen jatkamisen hienosäädöin todellisessa ympäristössä turvallisesti. Tässäkin on omat haasteensa, sillä vaikka valmiiksi koulutetun neuroverkon käyttäminen on laskentatehon näkökulmasta melko taloudellista, vaatii koulutuksen jatkaminen sovelluksen käyttönoton jälkeen merkittävän määrän laskentatehoa.

Yllä mainituista haasteista huolimatta koetulos yllätti tutkijan positiivisesti ja tekoäly saavutti 3 974 sukupolven jälkeen suoritustason jollaista tutkija odotti näkevänsä vasta 10 000 sukupolven jälkeen. Tutkija yritti ennen koetta helpottaa arvionsa mukaan haastavaa ongelmaa vähentämällä neuroverkon kontrolloimia parametreja: verkon ei annettu muuttaa hävittäjän tehoasetusta tai sivuperäsimen asentoa. Kokeen jälkeen voidaan todeta tämän todennäköisesti tarpeettomasti rajoittaneen neuroverkon toimintavaihtoehtoja. Tämä on hyvä esimerkki kappaleessa 2.2 esitetystä induktiivisesta vinoumasta, joka vaikuttaa neuroverkon oppimiseen.

5.1. Jatkotutkimustarpeet

Tutkimuksesta nousee useita jatkotutkimustarpeita, joista ensimmäisenä mainittakoon tutkimuksen kokeessa käytetyn tekoälyn hyperparametrien optimointi kokeellisesti. Tässä tutkimuksessa jouduttiin tyytymään käytettävissä olevien resurssien vuoksi tutkijan parhaaseen arvioon kustakin parametrasta. Tutkimuksen koetta on mahdollista jatkojalostaa kokeilemalla erityyppisiä neuroverkkoja tekoälyn runkona, etsiä parhaiten ongelmaan sopivia arvoja verkon hyperparametreille sekä kokeilla eri tyyppisiä evoluutioalgoritmeja ja näiden parametreja parhaan mahdollisen koulutustehokkuuden saavuttamiseksi. Tämän tutkimuksen kokeessa neuroverkot eivät oppineet suorittamaan vakaata tykkitähtäystä, mikä nostaa esille kysymyksen: Onko kokeen tekoälyn parametreja (neuroverkon rakenne, syötteen, evoluutioalgoritmin tyyppi, kelpoisuusfunktio, simulaatioiden pituus vs. iterointien määrä jne.) muuttamalla mahdollista luoda algoritmi, jota iteroimalla tekoäly oppii tähtäämään tarkasti?

Lisäksi tutkimuksen koeasetelmaa voidaan jatkojalostaa integroimalla simulaatioon ilmataiste-
luohjusmalli, joka mahdollistaa ilmataistelun näköetäisyyden ulkopuolella. Kokeessa käytettyä neuroverkkoa sekä geneettistä algoritmia voidaan hyödyntää sellaisenaan kyseisessä skenaariossa. Ainoastaan kelpoisuusfunktion parametreja tulisi muuttaa tutkittavaa skenaariota paremmin tukeviksi. Ohjusmalliksi voidaan valita geneerinen julkisen lähdemateriaalin malli, tai käyttää tarkempaa turvaluokiteltua ARS (*AMRAAM Raytheon Simulation*)¹ AIM-120-ohjusammuntasimulaatiota.

Yleisellä tasolla tutkija näkee tarpeelliseksi kartoittaa systemaattisesti Puolustusvoimissa tekoälyn avulla automatisoitavia prosesseja. Lisäksi on kartoitettava mahdollisia uusia tekoälyn avulla toteutettavia prosesseja, joilla kokonaistaloudellisuutta voidaan parantaa. Kartoittamisen jälkeen prosessien automatisointi tai tehostaminen on mahdollista toteuttaa osana Puolustusvoimien T&K-toimintaa esimerkiksi opinnäytetöinä. Luonnollisesti tämän tutkimuksen kaltainen todellisen hävittäjän kaartotaistelun automatisointi ei ole käytännössä mahdollista käytettävissä olevin resurssein, mutta tutkimuksen kokeen tulokset antavat näkökulmaa teoreettisista mahdollisuuksista tekoälyn hyödyntämisessä.

¹ Ilmavoimissa käytössä oleva Raytheonin julkaisema turvaluokiteltu AIM-120 -ohjussimulaatio-ohjelma.

6. JOHTOPÄÄTÖKSET

Tutkimuksen tehtävänä oli selvittää, pystytäänkö neuroverkko kouluttamaan ohjaamaan hävittäjää kaartotaistelusimulaatiossa evoluutioalgoritmia hyödyntäen. Tutkimustehtävästä johdettiin päätutkimuskysymys:

- Voidaanko hävittäjäohjaaja korvata neuroverkkoon ja evoluutioalgoritmiin perustuvalla tekoälyllä kaartotaistelusimulaatiossa?

Tutkimustehtävän tueksi päätutkimuskysymyksestä johdettiin seuraavat alakysymykset, joihin vastaamalla saatiin vastaus päätutkimuskysymykseen:

- Mitä ovat neuroverkko ja evoluutioalgoritmi ja miten ne toimivat?
- Miten neuroverkkoon ja evoluutioalgoritmiin perustuvaa tekoälyä voidaan käyttää hävittäjäohjaajana kaartotaistelusimulaatiossa?
- Onko kaartotaistelusimulaatioita iteroitaessa havaittavissa nouseva trendi tekoälyn suoritustasossa?

Ensimmäiseen alakysymykseen vastattiin luvussa kaksi. Toiseen alakysymykseen vastattiin luvussa kolme, tukeutuen toisen luvun teoreettiseen viitekehykseen. Kolmanteen alakysymykseen vastasivat tutkimuksen tulokset, jotka on esitelty luvussa neljä. Tässä luvussa alakysymysten vastauksista johdetaan vastaus päätutkimuskysymykseen.

Luvun 4 tulosten perusteella voidaan todeta hävittäjäohjaajan olevan korvattavissa neuroverkkoon ja evoluutioalgoritmiin perustuvalla tekoälyllä kaartotaistelusimulaatiossa. Eräajon tuloksissa nähdään kvantitatiivisen tarkastelun perusteella selkeä, syväoppimiselle tyypillinen, logaritmista kuvaajaa mukaileva nouseva trendi tekoälyn suoritustasossa iteroitien edetessä. Kvalitatiivisesti visuaaliseen tarkasteluun perustuvat asiantuntijalausunnat viittaavat myös nousevaan trendiin.

Kokeen tekoäly perustui neuroverkkoon, joka on deterministinen kuvaus syötteistä tuloksiin eli funktio. Tätä funktiota optimoitiin geneettisellä algoritmilla, joka on stokastista ohjelmointia hyödyntävä evoluutioalgoritmi, eli se hyödyntää suurta määrää laskentatehoa ratkaisun optimoimiseksi yrityksen ja erehdyksen kautta. Neuroverkko ja geneettinen algoritmi on esitelty tarkemmin luvussa 2.2. Luvussa 3 on esitelty neuroverkon ja geneettisen algoritmin parametrit sekä menetelmät, joilla tekoäly on asetettu hävittäjäohjaajan korvaajaksi simulaatiossa.

3 974 iteraation jälkeen tekoälyn suoritustaso ei vielä vastannut koulutetun hävittäjäohjaajan suoritustasoa. Väitettä tukevat liitteessä 3 esitetyt asiantuntijoiden sanalliset arviot. parhaan yksilön ja populaation suoritustasoa kuvaavat käyrät kuvissa 18 ja 19 antavat kuitenkin kulmaker-toimen ja käyrien arvojen erotuksen osalta viitteitä tekoälyn kehityspotentialista iterointimää-rää kasvatettaessa. Tekoälyn parametreja hienosäätämällä ja iterointia jatkamalla on siis mah-dollista parantaa tekoälyn suoritustasoa jopa hävittäjäohjaajaa vastaavalle tasolle.

Merkittävin haaste tekoälyn kehitykselle kokeessa oli ylisovittuminen, joka näkyy eniten kou-lutettujen neuroverkkoyksilöiden suoritustason heikentymisenä, kun vastustajan on joku muu kuin kyseisen yksilön sen hetkisen populaation edustaja. Ylisovittumisen ongelmaa voidaan ratkaista käyttämällä evoluutioalgoritmina alaluvussa 2.3 esitettyä biografiapohjaista optimoin-tia, joka pitää sisällään useita eri populaatioita, joiden välillä tapahtuu muuttoliikettä. Tämä lisää kaartotaistelutilanteisiin monipuolisuutta populaatioiden diversiteetin kasvaessa, mikä vä-hentää ylisovittumisen riskiä.

Tutkimuksen tuloksista johdettu johtopäätös, jonka mukaan hävittäjäohjaaja on korvattavissa tekoälyllä, voidaan perustella tämän tutkimuksen tulosten perusteella pitävän paikkansa aino-astaan simulaatiossa, jossa tekoälyllä on jatkuvasti käytettävissä täydellinen tilannetieto. Tut-kimuksessa ei ole todennettu tekoälyn kehittymispotentialia paremmin todellista maailmaa vastaavassa skenaariossa, jossa tilannetieto on vajaata ja sisältää kohinaa sekä virheitä.

Lisäksi on huomioitava kokeessa käytetyn tekoälyn olevan hyvin alkeellinen, sillä se pitää si-sällään ainoastaan yhden myötäsytöneuroverkon, jota optimoidaan yksinkertaisella geneetti-sellä algoritmilla. Tästä huolimatta tekoälyn kehityksen nopeus ja iteraatioiden myötä saavu-tettu suoritustaso yllättivät tutkijan positiivisesti.

Konseptitasolla kokeessa käytettyjen tekoälyn menetelmien, neuroverkkojen ja evoluutioalgo-ritmien, voidaan todeta olevan toimivia välineitä hävittäjän liikehtelyn optimoimisessa kaarto-taistelusimulaatiossa, joka on haastava online-säätöä vaativa optimointitehtävä

LÄHTEET

- [1] J. Poropudas ja K. Virtanen, ”Analyzing air combat simulation results with dynamic Bayesian networks,” tekijä: *Proceedings of the 2007 Winter Simulation Conference*, Washington DC, 2007.
- [2] T. Raivio, H. Ehtamo ja R. P. Hämäläinen, ”Aircraft trajectory optimization using nonlinear programming,” tekijä: *System Modelling and Optimization*, Helsinki, Helsinki University of Technology, 1995, s. 435-441.
- [3] J. Karelaiti, K. Virtanen ja T. Raivio, ”Game Optimal Support Time of a Medium Range Air-to-Air Missile,” *Journal of Guidance, Control and Dynamics*, osa/vuosik. 29, nro 5, s. 1061-1069, 2006.
- [4] ”Alpha Dogfight Trials Go Virtual for Final Events,” Defence Advanced Research Project Agency, 7.8.2020. [Online]. Available: www.darpa.mil/news-events/2020-08-07. [Haettu 27 05 2021].
- [5] ”Malli,” Skepsis, [Online]. Available: <https://www.skepsis.fi/ihmeellinen/malli.html>. [Haettu 4 1 2022].
- [6] B. L. Stevens, F. L. Lewis ja E. N. Johnson, *Aircraft Control and Simulation*, 3th edition toim., New Jersey: John Wiley & Sons, inc., 2016.
- [7] S. Hirsjärvi, P. Remes ja P. Sajavaara, Tutki ja Kirjoita, Keuruu: Otava, 2007.
- [8] H. Vilkkä, Tutki ja Mittaa, Helsinki: Tammi, 2014.
- [9] L. T. Nguyen, M. E. Ogburn, W. P. Gilbert, K. S. Kibler, P. W. Brown ja P. L. Deal, ”Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability,” NASA, Hampton Virginia, 1979.
- [10] Y. Huo, ”Model of F-16 Fighter Aircraft -Equation of Motion.,” System University of Southern California, Los Angeles.
- [11] Y. Huo, M. Mirmarini, P. Ioannou ja R. Colgren, ”Adaptive Linear Quadratic Design with Application to F-16 Fighter Aircraft,” tekijä: *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2004.
- [12] J. D. Kelleher, *Syväoppiminen - Kuinka tekoäly toimii*, Helsinki: Terra Cognita Oy, 2020.
- [13] S. Mirjalili, *Evolution Algorithms and Neural Networks - Theory and Application*, Springer, 2019.
- [14] ”United States customary units,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/United_States_customary_units. [Haettu 06 09 2021].
- [15] CSC, ”Supertietokone Puhti on avattu tutkijoiden käyttöön,” CSC, 2.9.2019. [Online]. Available: <https://www.csc.fi/-/supertietokone-puhti-on-avattu-tutkijoiden-kayttoon>. [Haettu 28 09 2021].
- [16] Intel, ”APP Metrics for Intel Microprocessors,” 24.2.2020. [Online]. Available: <https://www.intel.com/content/dam/support/us/en/documents/processors/APP-for-Intel-Core-Processors.pdf>. [Haettu 24 9 2021].
- [17] J. Tähtinen, E. Laakkonen ja M. Broberg, *Tilastollisen aineiston käsittelyn ja tulkinnan perusteet*, Turku: Turun yliopiston kasvatustieteiden laitos, 2020.
- [18] ”Runge-Kutta -menetelmät,” [Online]. Available: <https://fi.wikipedia.org/wiki/Runge-Kutta-menetelmät>. [Haettu 21 8 2021].

LIITTEET

Liite 1: F-16-lentomallin matemaattiset kaavat

Liite 2: Taulukot F-16-hävittäjän aerodynaamisten voimien kertoimista

Liite 3: Asiantuntija-arviot

Liite 4: Lähdekoodi ja eräajon tulokset

LIITE 1: F-16-lentomallin matemaattiset kaavat

Voimien kaavat:

$$u = V \cos a \cos \beta$$

$$v = V \sin \beta$$

$$w = V \sin a \cos B$$

$$V = \sqrt{u^2 + v^2 + w^2}$$

$$\dot{u} = rv - qw - g \sin \theta + \frac{1}{m} (\bar{q} S C_{x,t} + T)$$

$$\dot{v} = pw - ru + g \cos \theta \sin \phi + \frac{\bar{q} S}{m} C_{y,t}$$

$$\dot{w} = qu - pv + g \cos \theta \cos \phi + \frac{\bar{q} S}{m} C_{z,t}$$

$$\dot{V} = \frac{u\dot{u} + v\dot{v} + w\dot{w}}{V}$$

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right)$$

$$\beta = \sin^{-1} \left(\frac{v}{V} \right)$$

$$\dot{\alpha} = \frac{u\dot{w} - w\dot{u}}{(v \cos \beta)^2}$$

$$\dot{\beta} = \frac{V \cos \beta \dot{v} - v \cos \beta \dot{V}}{(v \cos \beta)^2}$$

Kinematiikka:

$$\dot{\phi} = p + \tan \theta (q \sin \phi + r \cos \phi)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi$$

$$\dot{\psi} = \frac{q \sin \phi + r \cos \phi}{\cos \theta}$$

Momentit:

$$\begin{aligned}\dot{p} &= \frac{I_y - I_z}{I_x} qr + \frac{I_{xz}}{I_x} (\dot{r} + pq) + \frac{\bar{q}Sb}{I_x} C_{L,t} \\ \dot{q} &= \frac{I_z - I_x}{I_y} pr + \frac{I_{xz}}{I_y} (r^2 - p^2) + \frac{\bar{q}S\bar{c}}{I_y} C_{M,t} - h_E r \\ \dot{r} &= \frac{I_x - I_y}{I_z} pq + \frac{I_{xz}}{I_z} (\dot{p} - qr) + \frac{\bar{q}Sb}{I_z} C_{N,t} - h_E q\end{aligned}$$

Paikkatieto:

$$\begin{aligned}\dot{N}_{dis} &= V \cos \alpha \cos \beta \cos \theta \cos \psi + V \sin \beta (\sin \phi \cos \psi \sin \theta - \cos \phi \sin \psi) \\ &\quad + V \sin \alpha \cos \beta (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \dot{E}_{dis} &= V \cos \alpha \cos \beta \cos \theta \sin \psi + V \sin \beta (\sin \phi \sin \psi \sin \theta - \cos \phi \cos \psi) \\ &\quad + V \sin \alpha \cos \beta (\cos \phi \sin \theta \sin \psi + \sin \phi \cos \psi) \\ \dot{h}_{dis} &= V \cos \alpha \cos \beta \sin \theta - V \sin \beta \sin \phi \cos \theta - V \sin \alpha \cos \phi \cos \theta\end{aligned}$$

Kertoimet:

$$\rho = 2,377 \times 10^{-3} (1,0 - 0,703 \times 10^{-5} h)^{4,14}$$

$$t = \begin{cases} 519(1,0 - 0,703 \times 10^{-5} h) & h < 35000,0 \\ 390,0 & h \geq 35000,0 \end{cases}$$

$$\begin{cases} \bar{q} = \frac{1}{2} \rho V^2 & \text{dynaaminen paine} \\ p_s = 1715 \rho t & \text{staattinen paine} \end{cases}$$

$$M = \frac{V}{\sqrt{1,4 \times 1716,3 \times t}}$$

$$C_{X,t} = \frac{\bar{c}}{2V} C_{Xq}(\alpha_d) q + C_x(\alpha_d, \delta_E)$$

$$C_{Y,t} = C_y(\beta_d, \delta_A, \delta_R) + \frac{b}{2V} [C_{Yr}(\alpha_d) r + C_{Yp}(\alpha_d) p]$$

$$= -0,02\beta_d + \frac{b}{2V} [r C_{Yr}(\alpha_d) - C_{Yp}(\alpha_d) p] + 0,021 \frac{\delta_A}{20,0} + 0,086 \frac{\delta_R}{30,0}$$

$$\begin{aligned}
C_{Z,t} &= C_z(\alpha_d, \beta_d, \delta_E) + \frac{\bar{c}}{2V} C_{Zq}(\alpha_d)q \\
&= C_{z,1}(\alpha_d, \beta_d) + \frac{\bar{c}}{2V} C_{Zq}(\alpha_d)q - 0,19 \frac{\delta_E}{25,0} \\
C_{L,t} &= C_l(\alpha_d, \beta_d, \delta_A, \delta_R) + \frac{b}{2V} [rC_{Lr}(\alpha_d) + C_{Lp}(\alpha_d)p] \\
&= C_{l,1}(\alpha_d, \beta_d) + \frac{b}{2V} [C_{Lr}(\alpha_d)r + C_{Lp}(\alpha_d)p] + C_{l,2}(\alpha_d, \beta) \frac{\delta_A}{20,0} + C_{l,3}(\alpha_d, \beta_d) \frac{\delta_R}{30} \\
C_{M,t} &= \frac{\bar{c}}{2V} C_{Mq}(\alpha_d)q + C_{Z,t}(X_{cgr} - X_{cg}) + C_m(\alpha_d, \delta_E) \\
C_{N,t} &= C_n(\alpha_d, \beta_d, \delta_A, \delta_R) + \frac{b}{2V} [C_{Nr}(\alpha_d)r + C_{Np}(\alpha_d)p] - \frac{\bar{c}}{b} C_{Y,t}(X_{cgr} - X_{cg}) \\
&= C_n(\alpha_d, \beta_d) + \frac{b}{2V} [C_{Nr}(\alpha_d)r + C_{Np}(\alpha_d)p] - \frac{\bar{c}}{b} C_{Y,t}(X_{cgr} - X_{cg}) + C_{n,2}(\alpha_d, \beta_d) \frac{\delta_A}{20,0} \\
&\quad + C_{n,3}(\alpha_d, \beta_d) \frac{\delta_R}{30}
\end{aligned}$$

LIITE 2: Taulukot F-16-hävittäjän aerodynaamisten voimien kertoimista

Taulukot mukailevat *Aircraft Control and Simulation* -kirjan liitteen A taulukoita [6, s. 714-722]. Tässä liitteessä taulukot esitetään Python -ohjelmointikieleen yhteensopivina.

Taulukko 1. F-16-lentomallin kaavoissa käytettävät vaimennuskertoimet

$d[0] = CXq$
 $d[1] = CYr$
 $d[2] = CYP$
 $d[3] = CZq$
 $d[4] = Clr$
 $d[5] = Clp$
 $d[6] = Cm q$
 $d[7] = Cnr$
 $d[8] = Cnp$

[-0.267, -0.110, 0.308, 1.34, 2.08, 2.91, 2.76, 2.05, 1.50, 1.49, 1.83, 1.21],
 [0.882, 0.852, 0.876, 0.958, 0.962, 0.974, 0.819, 0.483, 0.590, 1.21, -0.493, -1.04],
 [-0.108, -0.108, -0.188, 0.110, 0.258, 0.226, 0.344, 0.362, 0.611, 0.529, 0.298, -0.227],
 [-8.80, -25.8, -28.9, -31.4, -31.2, -30.7, -27.7, -28.2, -29.0, -29.8, -38.3, -35.3],
 [-0.126, -0.026, 0.063, 0.113, 0.208, 0.230, 0.319, 0.437, 0.680, 0.100, 0.447, -0.330],
 [-0.360, -0.359, -0.443, -0.420, -0.383, -0.375, -0.329, -0.294, -0.230, -0.210, -0.120, -0.100],
 [-7.21, -5.40, -5.23, -5.26, -6.11, -6.64, -5.69, -6.00, -6.20, -6.40, -6.60, -6.00],
 [-0.380, -0.363, -0.378, -0.386, -0.370, -0.453, -0.550, -0.582, -0.595, -0.637, -1.02, -0.840],
 [0.061, 0.052, 0.052, -0.012, -0.013, -0.024, 0.050, 0.150, 0.130, 0.158, 0.240, 0.150]

Taulukko 2. Aerodynaaminen kerroin C_x

[-0.099, -0.081, -0.081, -0.063, -0.025, 0.044, 0.097, 0.113, 0.145, 0.167, 0.174, 0.166],
 [-0.048, -0.038, -0.040, -0.021, 0.016, 0.083, 0.127, 0.137, 0.162, 0.177, 0.179, 0.167],
 [-0.022, -0.020, -0.021, -0.004, 0.032, 0.094, 0.128, 0.130, 0.154, 0.161, 0.155, 0.138],
 [-0.040, -0.038, -0.039, -0.025, 0.006, 0.062, 0.087, 0.085, 0.100, 0.110, 0.104, 0.091],
 [-0.083, -0.073, -0.076, -0.072, -0.046, 0.012, 0.024, 0.025, 0.043, 0.053, 0.047, 0.040]

Taulukko 3. Aerodynaaminen kerroin C_z

[0.770, 0.241, -0.100, -0.416, -0.731, -1.053, -1.366, -1.646, -1.917, -2.120, -2.248, -2.229]

Taulukko 4. Aerodynaaminen kerroin C_l

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [-0.001, -0.004, -0.008, -0.012, -0.016, -0.022, -0.022, -0.021, -0.015, -0.008, -0.013, -0.015],
 [-0.003, -0.009, -0.017, -0.024, -0.030, -0.041, -0.045, -0.040, -0.016, -0.002, -0.010, -0.019],
 [-0.001, -0.010, -0.020, -0.030, -0.039, -0.054, -0.057, -0.054, -0.023, -0.006, -0.014, -0.027],
 [0.000, -0.010, -0.022, -0.034, -0.047, -0.060, -0.069, -0.067, -0.033, -0.036, -0.035, -0.035],
 [0.007, -0.010, -0.023, -0.034, -0.049, -0.063, -0.081, -0.079, -0.060, -0.058, -0.062, -0.059],
 [0.009, -0.011, -0.023, -0.037, -0.050, -0.068, -0.089, -0.088, -0.091, -0.076, -0.077, -0.076]

Taulukko 5. Aerodynaaminen kerroin $C_{l,2}$

[-.041, -.052, -.053, -.056, -.050, -.056, -.082, -.059, -.042, -.038, -.027, -.017],
 [-.041, -.053, -.053, -.053, -.050, -.051, -.066, -.043, -.038, -.027, -.023, -.016],
 [-.042, -.053, -.052, -.051, -.049, -.049, -.043, -.035, -.026, -.016, -.018, -.014],
 [-.040, -.052, -.051, -.052, -.048, -.048, -.042, -.037, -.031, -.026, -.017, -.012],
 [-.043, -.049, -.048, -.049, -.043, -.042, -.042, -.036, -.025, -.021, -.016, -.011],
 [-.044, -.048, -.048, -.047, -.042, -.041, -.020, -.028, -.013, -.014, -.011, -.010],
 [-.043, -.049, -.047, -.045, -.042, -.037, -.003, -.013, -.010, -.003, -.007, -.008]

Taulukko 6. Aerodynaaminen kerroin $C_{l,3}$

[.005, .017, .014, .010, -.005, .009, .019, .005, .000, -.005, -.011, .008],
 [.007, .016, .014, .014, .013, .009, .012, .005, .000, .004, .009, .007],
 [.013, .013, .011, .012, .011, .009, .008, .005, .000, .005, .003, .005],
 [.018, .015, .015, .014, .014, .014, .014, .015, .013, .011, .006, .001],
 [.015, .014, .013, .013, .012, .011, .011, .010, .008, .008, .007, .003],
 [.021, .011, .010, .011, .010, .009, .008, .010, .006, .005, .000, .001],
 [.023, .010, .011, .011, .011, .010, .008, .010, .006, .014, .020, .000]

Taulukko 7. Aerodynaaminen kerroin C_m

[.205, .168, .186, .196, .213, .251, .245, .238, .252, .231, .198, .192],
 [.081, .077, .107, .110, .110, .141, .127, .119, .133, .108, .081, .093],
 [-.046, -.020, -.009, -.005, -.006, .010, .006, -.001, .014, .000, -.013, .032],
 [-.174, -.145, -.121, -.127, -.129, -.102, -.097, -.113, -.087, -.084, -.069, -.006],
 [-.259, -.202, -.184, -.193, -.199, -.150, -.160, -.167, -.104, -.076, -.041, -.005]

Taulukko 8. Aerodynaaminen kerroin C_n

[.0, .0, .0, .0, .0, .0, .0, .0, .0, .0, .0, .0],
 [.018, .019, .018, .019, .019, .018, .013, .007, .004, -.014, -.017, -.033],
 [.038, .042, .042, .042, .043, .039, .030, .017, .004, -.035, -.047, -.057],
 [.056, .057, .059, .058, .058, .053, .032, .012, .002, -.046, -.071, -.073],
 [.064, .077, .076, .074, .073, .057, .029, .007, .012, -.034, -.065, -.041],
 [.074, .086, .093, .089, .080, .062, .049, .022, .028, -.012, -.002, -.013],
 [.079, .090, .106, .106, .096, .080, .068, .030, .064, .015, .011, -.001]

Taulukko 9. Aerodynaaminen kerroin $C_{n,2}$

[-.041, -.052, -.053, -.056, -.050, -.056, -.082, -.059, -.042, -.038, -.027, -.017],
 [-.041, -.053, -.053, -.053, -.050, -.051, -.066, -.043, -.038, -.027, -.023, -.016],
 [-.042, -.053, -.052, -.051, -.049, -.049, -.043, -.035, -.026, -.016, -.018, -.014],
 [-.040, -.052, -.051, -.052, -.048, -.048, -.042, -.037, -.031, -.026, -.017, -.012],
 [-.043, -.049, -.048, -.049, -.043, -.042, -.042, -.036, -.025, -.021, -.016, -.011],
 [-.044, -.048, -.048, -.047, -.042, -.041, -.020, -.028, -.013, -.014, -.011, -.010],
 [-.043, -.049, -.047, -.045, -.042, -.037, -.003, -.013, -.010, -.003, -.007, -.008]

Taulukko 10. Aerodynaaminen kerroin $C_{n,3}$

[-.018, -.052, -.052, -.052, -.054, -.049, -.059, -.051, -.030, -.037, -.026, -.013],
[-.028, -.051, -.043, -.046, -.045, -.049, -.057, -.052, -.030, -.033, -.030, -.008],
[-.037, -.041, -.038, -.040, -.040, -.038, -.037, -.030, -.027, -.024, -.019, -.013],
[-.048, -.045, -.045, -.045, -.044, -.045, -.047, -.048, -.049, -.045, -.033, -.016],
[-.043, -.044, -.041, -.041, -.040, -.038, -.034, -.035, -.035, -.029, -.022, -.009],
[-.052, -.034, -.036, -.036, -.035, -.028, -.024, -.023, -.020, -.016, -.010, -.014],
[-.062, -.034, -.027, -.028, -.027, -.027, -.023, -.023, -.019, -.009, -.025, -.010]

LIITE 3: Asiantuntija-arviot

Asiantuntija-arvioissa arvioidaan pääasiakirjan taulukon 7 mukaisten tekoälyn ohjaamien hävittäjien suoritustasoa. Tekoäly-yksilöt on nimetty peitenimillä niiden sukupolven piilottamiseksi seuraavasti:

#1	- Markku
#10	- Henri
#100	- Pekka
#1000	- Eero
#3973	- Matti (Toimii vastustajana kaikille muille)
#3974	- Antti

Liitteen seuraavilla sivuilla asiantuntijoiden täyttämät arviointilomakkeet.

ID	Score(1-5)	Notes	Ranking
Markku	1	Lensi suoraa maahan	1: Antti 2: Eero 3: Henri 4: Pekka 5: Markku
Antti	4	Liikkeiden <u>suuntaaminen</u> heikkoa. Jäi useassa kohdassa maalin etusektoriin. Ennakkokaarrot puuttellisia. Hakeutui pääsääntöisesti 2 ympyrän geometrioihin ja lopulta pääsi hallinta-asemaan.	
Henri	3	Aloituksessa vaarallinen kaarron vaihto, jonka jälkeen ongelmia geometrioiden valinnassa. Ei osannut toimia saksi-tyyppisessä liikehtelyssä. Energian hallinta puuttellista. <u>Departure</u> .	
Eero	3	Energianhallinnassa merkittäviä puutteita. Ei käsitystä jatkogeometrioiden suunnittelusta (ympyrä ylös, pudotti nokan maalin etusektoriin).	
Pekka	2	Aloitus viivästyi + iso nopeus -> päättyi uhatuksi. Vaarallisia kaarronvaihtoja -> koko ajan uhattuna. Punainen aloitti vastaavanlaiset typeryydet jolloin sininen pääsi uhkaamaan. <u>Maahan törmäys</u> .	
Matti		Matin käytöksessä ei merkittäviä muutoksia eri vastustajien välillä, mutta kohtaamisten sisällä toiminta ja geometriahallinta muuttuivat järjekevistä surkeiksi kuten sinisellä.	

General notes:

ID	Score(1-5)	Notes	Ranking
Markku	1	Markku oli löytänyt vain poikittaisohjauksen. Takaisin alkeiskoulutukseen.	1: Antti 2: Eero 3: Henri 4: Pekka 5: Markku
Antti	3,5	Antti suosi 2-ymp. taktiikkaa. Pystysuunnassa liikkeen suuntaminen oikein tuottaa haasteita. 2-ymp- ennakkokaarto onnistuu satunnaisesti.	
Henri	2	Ei täysin itsetuhoinen, mutta ymmärrys kaartotaistelusta täysin hukassa. Päätyy departureen.	
Eero	3	Ronski käsiala. Liikkeen suuntamisessa vielä isoja haasteita, varsinkin ennakkokaarron hyödyntämisessä. Selkeä 1-ympyrän kaveri.	
Pekka	1,5	Parempi kuin Markku, mutta Henriä huonompi ymmärrys kaartotaistelun perusteista.	
Matti	3,5	Vastustajan vaihtuessa Matti ei osannut soveltaa liikehtelyä eikä hyödyntää vastustajan virheitä vaan lensi ennalta oppimimaan temppuja.	

General notes:

Kukaan ei oppinut kaarronvaihtoja ja muuttamaan toimintaa yhden ja kahden ympyrän tilanteiden välillä.

ID	Score(1-5)	Notes	Ranking
Markku	1-	-3x Ail roll, decent into ground -Accure of other guy?	1: Antti 2: Eero 3: Henri 4: Pekka 5: Markku
Antti	~3	-1C High +Alt advantage at SSM +Loops -Nose off time ~50 ??? roller -1+19 ail rolls -> 2C equal (bandit error) +Out of plane -LV placement +2C offensive (took 3 min to get offensive)	
Henri	2-	1C High -Nose off to denfensive -Poor LV placement (Bandit won't capitalise) -Nose high ballistic, but fixes it -Departs	
Eero	2+	1C High +Aoa excursion for shot -Came off and go denfensive, bandit doesn't capitalise -LV placement -Won't turn w/ bandit, especially to 6 o'clock -Kept turning in front of bandit	
Pekka	1+	1C High +alt adv but no def maneuver -departs almost, then very low -won't turn back in to bandit -poor aoa + g control -LV placement -was defensive but wouldn't stay there -HITS GROUND	
Matti	3	-Failed to capitalize on blue mistakes -LV placement -Lost to Antti after 3 min, so okay until then	

General notes:

- All had poor lift vector placement and did not predict where the other would go
- If they were offense, they tended to make a mistake by turning away

ID	Score(1-5)	Notes	Ranking
Markku	1	Vaakakierre ei ole taistelu- vaan taitolentoliike. Aloituksesta kuuluu lähteä 1 tai 2 ympyrään, ei suoraan alas ja törmätä maahan. Toki maali varmasti ampui meidät ennen törmäystä.	1: Antti 2: Eero 3: Henri 4: Pekka 5: Markku
Antti	3	Alussa pitää vetää tiukemmin pystyyn. Punaiselle annettiin turhaan 2 ammutapaikkaa. Tämän jälkeen liikehdintä liikaa leadille -> ajaututtiin punaisen eteen. Lopussa päästiin lag:n / oikean ennakon kautta lähes hallinta-asemaan. Tee se aiemmin.	
Henri	1	Jäit pystyliikeestä turhaan maalin eteen. Suuntaaminen ei nyt hallussa, vaan lähtee keskimäärin 180° väärään paikkaan. Ajoit liian ahnaasti ja vaarallisesti lopussa lähelle - > DEP.	
Eero	2	Älä tiputa ylhäältä nokkaa ensimmäisenä, koska toisesta silmukasta ajetaan suoraan maalin eteen. Oman ammutapaikan sait koska maali ajoi tyhmästi, ja tämän jälkeen taas olit defensive. Lopussa ajoit paremmin neutraalina. Suuntausta pitää vielä harjoitella.	
Pekka	1	Aloitukset myöhässä ja liian löysä. Vedit alas kohti maalia liian suurella ennakolla, jonka oikaisussa liikaa vetoa -> koneen Dep. Tämän jälkeen tehotonta ajamista molemmilta ja ground kill.	
Matti	3	Aloitukset tehokkaita ja tiukkoja. Parempaa suuntaamista pääsääntöisesti kuin muilla, mutta ei ratkaisuhalukuutta: Aina stalemate.	

General notes:

Koneet eivät osaa hakeutua ammunnan (=lead) jälkeen lagille tai ylipäätään kohti control zonea. Off / Def asetelma vaihtuu jatkuvasti & epähallitusti.

LIITE 4: Lähdekoodi ja eräajon tulokset

Lähdekoodin tiedostot sekä eräajon tulokset ovat saatavissa tutkielman liitteenä 4.1 GitHub-verkkosivustolta (<https://github.com/MikkoKyto/F-16-Dogfight-AI>) tai tutkijalta erikseen pyydettyessä (mikko.kyto@gmail.com) seuraavan listan mukaisesti:

adc.py	- standardi-ilmakehämalli
cl.py	- F-16 mallin kallistusmomenttikertoimen (<i>rolling moment</i>) taulukko ja interpolointifunktio
cm.py	- F-16 mallin pystykallistusmomenttikertoimen (<i>pitching moment</i>) taulukko ja interpolointifunktio
cn.py	- F-16 mallin suentakallistusmomenttikertoimen (<i>yawing moment</i>) taulukko ja interpolointifunktio
control.py	- Neuroverkon tuottamat ohjainmuuttujat F-16 mallille
conversions.py	- Erinäisiä ohjelmissa tarvittavia muunnoksia
cx.py	- F-16 mallin x-akselin suuntaisten aerodynaamisten voimien kertoimien taulukko ja interpolointifunktio
cy.py	- F-16 mallin y-akselin suuntaisten aerodynaamisten voimien kaava
cz.py	- F-16 mallin z-akselin suuntaisten aerodynaamisten voimien kertoimien taulukko ja interpolointifunktio
damping.py	- F-16 mallissa tarvittavien vaimennuskertoimien taulukot ja interpolointifunktio
dlda.py	- F-16 mallin siivekkeiden tuottaman kallistusvoiman taulukko ja interpolointifunktio
dldr.py	- F-16 mallin sivuperäsimen tuottaman kallistusvoiman taulukko ja interpolointifunktio
dnda.py	- F-16 mallin siivekkeiden tuottaman suentakallistusvoiman taulukko ja interpolointifunktio
dndr.py	- F-16 mallin sivuperäsimen tuottaman suentakallistusvoiman taulukko ja interpolointifunktio
f_16.py	- Täyden epälineaarisen F-16 mallin pääohjelmakoodi
main.py	- Simulaation pääohjelmakoodi
neuralnet.py	- Neuroverkko ja geneettinen algoritmi
pdot.py	- F-16 moottorimallin tehoasetuksen muutosnopeuden funktio
rotations.py	- Simulaatiossa tarvittavia koordinaatistomuutoksia
rtau.py	- ”pdot” -funktion käyttämä ”rtau” -funktio

scoring.py	- Kelpoisuusfunktiot
tgrear.py	- Tehovivun vaihteiston mallinnus
thrust.py	- F-16 moottorin työntövoimamallin taulukot ja interpolointifunktio
savA.txt	- Kunkin sukupolven neuroverkkojen kelpoisuuden keskiarvo
savB.txt	- Kunkin sukupolven parhaan neuroverkkoyksilön kelpoisuus
savBN.txt	- Kunkin sukupolven paras neuroverkkoyksilö (suuri tiedosto saatavissa tutkijalta pyydettäessä)
savP.txt	- Viimeisen sukupolven populaatio