

Rakettitieteen jalanjäljillä

Osa 6: Haisunäätä suuntaa taivaalle

Sampo Niskanen

SATS:n raketijaoksen aloittaessa hybridirakettiprojektinsa avautui aivan uusia mahdollisuuksia asentaa hyötykuormaa raketin kyytiin. Jo ensimmäisen Haisunäädän kyytiin tuli useita kameroita, paikannusjärjestelmiä ja radiolähettämiä. Kun Haisunäädän isompaa veljeä Iso-Haisua alettiin rakentaa, ihmeteltiin, mitä kaikkea käytettävissä olevalla tilalla voitaisiin tehdä.

Yksi esiin nousseista ideoista oli yrittää yksinkertaista raketin ohjausta — pyörimisen hallintaa. Raketin akselinsa ympäri pyöriminen ei sinällään haittaa raketin lentoa, mutta videokuvaa se haittaa. Tavoite oli siis säätää jotain ohjainpintaa niin, että se pysäyttää raketin pyörimisen.

Ohjausta varten pyöriminen täytyy mitata jollain tapaa. Perinteisesti pyörimisnopeuden mittaamiseen käytetään gyroa. Pelkäsimme kuitenkin, että hybridimoottorin aiheuttama voimakas värinä tekisi gyron toimimattomaksi — joskin olemme sittemmin kuulleet gyrojen valmistajilta vakuutuksia ettei värinä nykyisiä gyroja haittaa. Halusimme myös testata jotain muuta.

Diplomityötäni ja OpenRocketin validointilentoja varten Antti J. Niskanen oli rakentanut lentotietokoneen, joka tallensi magnetometrin arvoja. Näistä pystyi mittaamaan raketin pyörimisnopeuden ja vertaamaan sitä simuloituun arvoon. Päädyimme tätä myötä käyttämään magnetometrejä myös Iso-Haisun pyörimisnopeuden reaaliaikaiseen mittaamiseen.

Raketin ensimmäiseen versioon rakennettiin kaksi ohjaussiivekettä, jotka näkyvät kuvassa raketin yläosassa. Myöhempiä lentoja varten nämä poistettiin, ja yhteen pääsiivekkeeseen asennettiin sen sijaan ohjainpinta jota pystyi kääntämään. Hyötykuormaan rakennettiin Arduino-pohjainen lentotietokone mittaamaan pyörimissuuntaa ja ohjaamaan siivekettä.

PID-säätimet

Yhden vapausasteen hallinta on helppo toteuttaa PID-säätimellä. Jokin anturi mittaa haluttua suuretta (pyörimisnopeutta) ja säädin päivittää ohjausjärjestelmää, joka vaikuttaa suureeseen (ohjainpintaa). Säätimen ei tarvitse olennaisesti tietää mitään ohjattavasta järjestelmästä — se vain yrittää saada mitatun arvon vastaamaan tavoitearvoa.

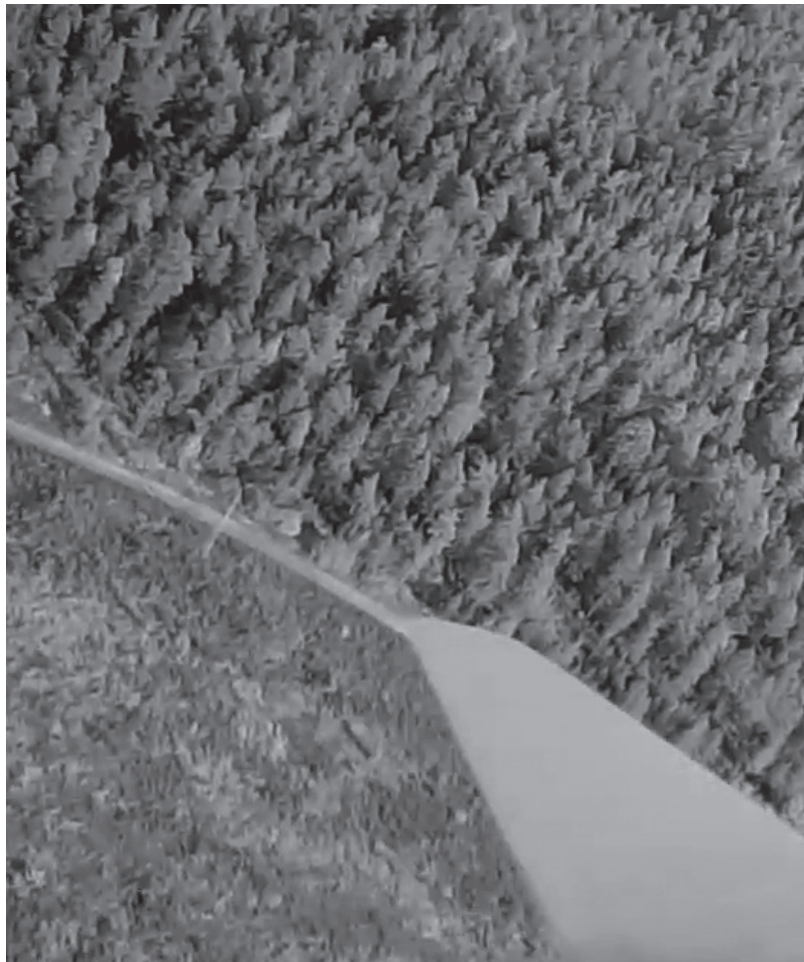
PID-säätimen syöte on hetkellinen virhe $e(t)$, eli hetkellisen mittaustuloksen ja tavoitearvon erotus. Ulostulo puolestaan riippuu kolmesta säätimen laskemasta arvosta: virheen hetkellisestä arvosta (proportional, P), sen integraalista ajanhetkestä 0 alkaen (integral, I) sekä hetkellisestä derivaatasta (derivative, D). Nämä kolme arvoa kerrotaan vakioilla K_p , K_i ja K_d ja summataan yhteen, jolloin saadaan säätimen ulostulo $u(t)$:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Jotta PID-säädin toimisi oikein, nämä kolme vakiota täytyy saada sopiviksi. Säädin ei ole herkkä vakioiden tarkoilta arvoille, mutta niiden pitää olla suunnilleen oikeat. Väärillä arvoilla säädin saattaa jäädä oskilloimaan, toimia liian hitaasti tai jopa säätää väärään suuntaan.

Jo varsin alkuvaiheessa päätimme jättää derivaatan D laskematta ja käyttämättä. Derivaatan laskeminen mitatuista arvoista kasvattaa voimakkaasti mittausvirheitä. PID-säädin toimii oikein myös ilman derivaattaa (ns. PI-säädin), derivaatan hyödyntäminen vain nopeuttaa tavoitearvon saavuttamista.

Säätimeen piti silti saada jotkin arvot K_p ja K_i . Näiden laskeminen analyttisesti raketin ominaisuuksista olisi ollut erittäin haastavaa ja työlästä. Tässä OpenRocket tuli avuksi.



Haisunäätä menossa kohti korkeuksia. Kuva: SATS.

Säätöä OpenRocketilla

Olin alusta pitäen rakentanut OpenRocketin olemaan laajennettavissa omalla koodilla. Samalla kun ohjelmisto simuloi raketin lentoa, se kutsuu jokaisella aika-askeleella ns. kuuntelijakoodia (simulation listener). OpenRocket hyödyntää sisäisesti monessa asiassa kuuntelijakoodia, mutta käyttäjä voi syöttää tähän myös omaa koodia.

Kun Iso-Haisu oli mallinnettu OpenRocketilla, kirjoitin koodipätkän, joka mallinsi PID-säätimen toimintaa. Jokaisella aika-askeleella OpenRocket kutsui myös tätä koodia, joka laski sen hetkisen PID-säätimen ulostulon, ja asetti raketin ohjainpinnan uuteen asentoon. Koodi mallinsi myös servon äärellisen reaktionopeuden, eli ohjainpinta kääntyi sekunnin murto-osia ohjauuskäskyn jälkeen.

Tällä järjestelyllä pystyi testaamaan eri K_p ja K_v arvoja, ja tutkimaan niiden vaikutusta lentorataan. Kallistamalla pääsiivekkeitä mallissa hieman raketin sai pyörimään, mikä ohjausjärjestelmän piti pysäyttää. Pyörimishäiriöitä pystyi myös tuottamaan toisella kuuntelijakoodilla.

Jo diplomityötä tehdessä olin havainnut, että simulaation pyörimisen mallinnus ei ollut aivan kohdallaan. Simulaatio tuotti pyörimisnopeudeksi vain puolet oikealla raketilla mitatusta pyörimisnopeudesta. Tämä mallinnus riitti kuitenkin sopivien säätöparametrien haarukoimiseksi, joita saattoi sen jälkeen vielä muuttaa vähän konservatiivisempaan (hitaampaan) suuntaan.

Tämä ei kuitenkaan ollut loppu OpenRocketin hyödyntämiselle. Seuraavaksi oli vuorossa itse lentotietokoneen ohjelmointi, jonka pitäisi toimia oikein jo ensimmäisellä

lennolla. Ongelma on, ettei ohjelmaa pysty testaamaan aidossa olosuhteissa ennen lennätystä.

Hardware-in-the-loop

Lentotietokoneen toiminto on periaatteessa yksinkertaista: lue sisään arvot magnetometreistä, tee tarvittavat laskutoimitukset, kirjoita ulos uusi säätöarvo ohjainservolle, ja odota seuraavaa ajanhetkeä.

Ohjelmistoa kirjoittaessani siirsin magnetometrin lukuruutiinit ja servon ohjauskomennot omiin funktioihin ja ehdollistin ne. Tietyllä parametrilla käännettynä ohjelma ei luekaan arvoja Arduinon analogisisääntulosta, vaan komentoriviltä, johon se myös kirjoittaa servon ohjauuskäskyn. Tällöin OpenRocketin kuuntelijakoodin sai kytkettyä varsinaiseen ohjausohjelmistoon, ja testattua ohjelmistoa "aidolla" lentodatalla — jota simuloitiin sitä mukaa kuin sitä tarvittiin.

Testausprosessi ei kuitenkaan päättynyt tähän, vaan seuraava askel oli tuottaa ohjelmistoversio, jota ajettiin lopullisella fyysisellä lentotietokoneella, joka luki arvot ja antoi ohjauuskäskyt USB-väylän yli tietokoneelta. Näin saatiin testattua ohjelmisto ja algoritmit aidolla datalla, aidossa ympäristössä.

Tämä toiminto osoittautui korvaamattomaksi avuksi, kun se paljasti yhden mystisimmistä bugeista joita olen joutunut selvittämään. Ohjelma laski väärin ja sai outoja tuloksia, vaikka kuinka varmistin koodin olevan oikein. Usean tunnin selvittämisen jälkeen selvisi, että Arduinokääntäjä, joka tuli oletusarvoisesti Ubuntun mukana, sisälsi bugin, minkä vuoksi ohjelma laski kertolaskut väärin!

Toinen selvinnyt bugi oli, että yhden laskuissa käytetyn numerotyyppin koko oli Arduinossa eri kuin mitä se oli tietokoneessa ajettaessa. Tämä aiheutti ylivuodon, mikä tuotti määrättyissä tilanteissa virheellisiä tuloksia. Tätäkin bugia olisi ollut erittäin vaikeaa löytää muutoin.

Tämä todistaa miten hyödyllinen apuväline avoimen lähdekoodin rakettisimulaattori voi olla edistyneille rakettiharrastajille. Esimerkiksi RockSimilla vastaava testaaminen olisi ollut mahdotonta.

Muutama viikko sitten Portland State Universityn opiskelijat ottivat yhteyttä ja ilmoittivat käyttäneensä OpenRocketia harjoitustyössään, jossa olivat myös tehneet hardware-in-the-loop testausta OpenRocketilla. He olivat laajentaneet ohjelmistoa tarvittavassa määrin testatakseen oman lentotietokoneensa toimintaa simuloitua dataa vasten. Laajennukset tulevat todennäköisesti pian osaksi OpenRocketia.



Iso-Haisun lennot ja tulevaisuus

Iso-Haisun ohjausjärjestelmä on kolmella eri lennollaan toiminut vaihtelevasti. Ensimmäisellä lennolla ohjausjärjestelmä toimi varsin hyvin suurimman osan lennosta, mutta aivan loppuvaiheessa raketti alkoi oskilloida voimakkaasti edestakaisin. Toisella lennolla, jossa ohjauspinta oli siirretty pääsiivekkeen osaksi, järjestelmä näytti toimivan hienosti. Kolmannella lennolla taas raketti pyöri villisti koko lennon ajan.

Mahdollisia syitä tähän on etsitty ja löydetty monia. Ensimmäinen epäily oli kamera, joka oli teipattu pääsiivekkeen kylkeen, aiheuttaen epäsymmetriaa. Sen jälkeen selvisi, että pääsiivekkeiden edessä olevat ohjaussivekkeet voivat aiheuttaa ilmapirtaan niin paljon pyörteitä, että osuessaan pääsiivekkeisiin se kumoaa ohjausvaikutuksen — minkä seurauksena ohjauspinta siirrettiin alas.

Vasta kolmannen lennon jälkiselvittelyissä tajusimme kriittisen virheen lentotietokoneen laskenta-algoritmissa. Raketin pyöriessä magnetometreista tuleva signaali on sini-muotoista. Lentotietokone derivoi nämä signaalit, korottaa ne toiseen potenssiin ja summaa ne. Tämä tuottaa tuloksen, joka on verrannollinen pyörimisnopeuteen.

Mitä järjestelmässä ei ollut huomioitu oli, että raketin kallistuessa magnetometrien mittaaman magneettikentän voimakkuus muuttuu. Jos raketti kallistuu magneettiviivoja vastaan, amplitudi voi moninkertaistua — jolloin myös lentotietokoneen mittaama pyörimisnopeus moninkertaistuu! Tilannetta vain pahentaa se, että Suomessa maan magneettikenttä on noin 75° kulmassa maahan nähden, jolloin pienelläkin kallistumisella voi olla suuri vaikutus kentän voimakkuuteen.

Kun tarkastelimme Iso-Haisun lentoratoja, ne näyttävät olevan yhtenevät havainnon kanssa. Aina, kun ohjausjärjestelmä on mennyt pieleen, raketti on kallistunut etelän suuntaan. Seuraavia lennätöksiä varten ohjelmistoa täytyykin joko korjata ottamaan tämä huomioon, tai vaihtaa laitteisto käyttämään gyroa pyörimisnopeuden mittamiseen.

Tästäkin huolimatta olemme osoittaneet ohjausjärjestelmän toimivaksi, vaikkakin laskenta-algoritmeissa on vielä korjattavaa. Jatkoa varten on myös pohdittu mahdollisuutta tavoitella tiettyä pyörimis-

asentoa sen sijaan että vain pyöriminen pysäytetään. Tämä saattaisi helpottaa ohjausta, koska kääntyminen pois halutusta asennosta pystytään havaitsemaan heti, ilman mitään viiveitä. □

Artikkelisarjan aiemmissa osissa olen kertonut OpenRocket-ohjelmiston kehitysvaiheista ja raketin aerodynaamisten ominaisuuksien laskennasta. Myöhemmissä osissa on luvassa tietoa mm. moniulotteisesta optimoinnista sekä tulevaisuuden suunnitelmista.

OpenRocket-ohjelmiston sekä diplomityön saa ladata osoitteesta <http://openrocket.sourceforge.net/> Artikkelisarjan aiemmat osat on luettavissa SATS:n sivuilta osoitteesta <http://www.sats-saff.fi/>



Iso-Haisu laukaisutelineessä. Kuva: Timo Toivanen.