



TAMPEREEN TEKNILLINEN YLIOPISTO

Hydrauliikka ja automatiikka

30.4.2009

VILLE KINNANEN
HARDWARE IN THE LOOP -SIMULOINNIN KÄYTTÖ
TUOTEKEHITYKSESSÄ

IHA-3300 / ELE-3350 Mekatroniikan erityiskysymyksiä

Tutkielma

SISÄLLYS

1.	Johdanto	1
2.	HIL -simuloinnin periaate	2
2.1.	HIL -simulointijärjestelmän komponentit.....	4
2.2.	HIL -simuloinnin edut.....	6
2.3.	HIL -simuloinnin rajoitukset.....	6
3.	HIL -simuloinnin hyödyntäminen.....	8
3.1.	HIL -simuloinnin käytännön toteutus	8
4.	Sovellutusesimerkkejä	11
4.1.	HIL -simulointi skootterin moottorinohjauksen kehityksessä	11
4.2.	HIL -simulointi UAV:n autopilotin kehityksessä	13
5.	Yhteenveto	16
	Lähteet.....	17

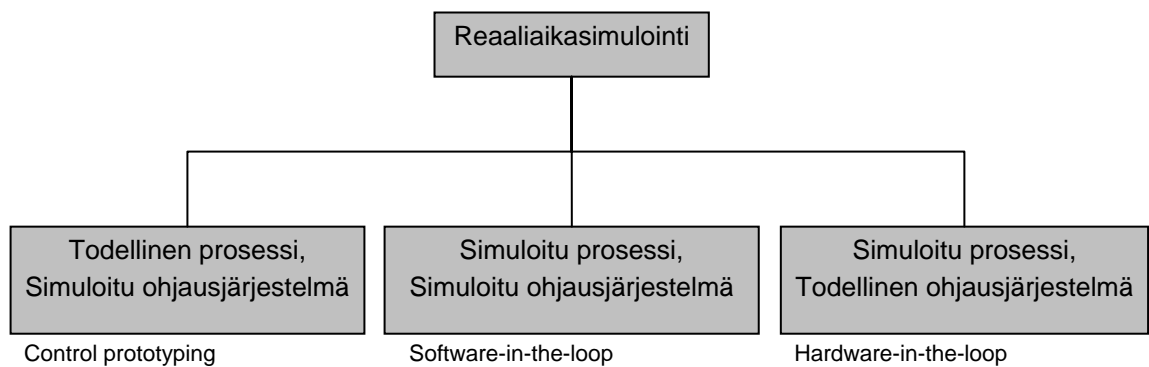
1. JOHDANTO

Reaaliaikasimulointia hyödynnetään yhä enemmän mekatronisten sulautettujen järjestelmien suunnittelussa. Simuloinnin lisääntymiseen vaikuttavat monet tekijät, kuten sulautettujen järjestelmien lisääntyminen, kompleksisuuden kasvu sekä mikrokontrollerien ja tietokoneiden suorituskyvyn parantuminen. Erityisen tärkeäksi simulointi tulee, kun pyritään samanaikaisesti suunnittelemaan sekä sulautetun järjestelmän ohjelmistoa, että sen hardwarea ilman useita iterointikertoja. Tämä taas on välttämätöntä, jotta saavutetaan tavoiteltava tuotteen nopea lanseeraus markkinoille sekä siihen suhteessa alhaiset tuotekehityskustannukset. Lisäksi laajojen ja uniikkien sulautettujen järjestelmien käytännön testaaminen tuotekehitysvaiheessa on hyvin vaikeaa.

Tässä tutkielmassa esitellään erään reaaliaikasimuloinnin muodon, Hardware-in-the-loop (HIL) -simuloinnin periaate, sen edut ja rajoitukset sekä syitä valita HIL -simulointi osaksi tuotekehitysprojektia. Lisäksi käydään läpi, kuinka HIL -simulointi voidaan toteuttaa ja tutustutaan sen asettamiin vaatimuksiin ohjelmistolle ja laitteistolle. Lopuksi tutustutaan vielä HIL -simuloinnin toteutus esimerkkeihin.

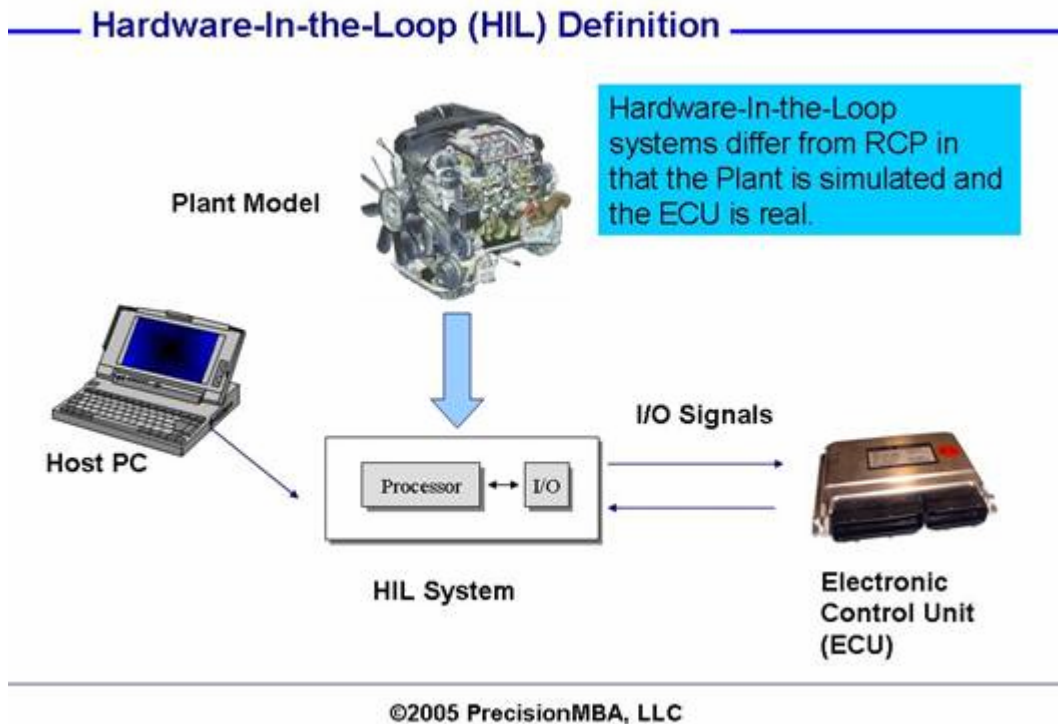
2. HIL -SIMULOINNIN PERIAATE

Hardware in the loop -simulointi on yksi reaaliaikasilmoinnin muoto. Reaaliaikasilmointi jaetaan tyypillisesti kolmeen osa-alueeseen toteutustavan mukaisesti. Jako suoritetaan sen mukaan, ovatko prosessi ja ohjausjärjestelmä toteutettuja ohjelmallisesti vai ovatko ne todellisia järjestelmässä käytettyjä komponentteja. Reaaliaikasilmoinnin osa-alueet näkyvät kuvassa 1.



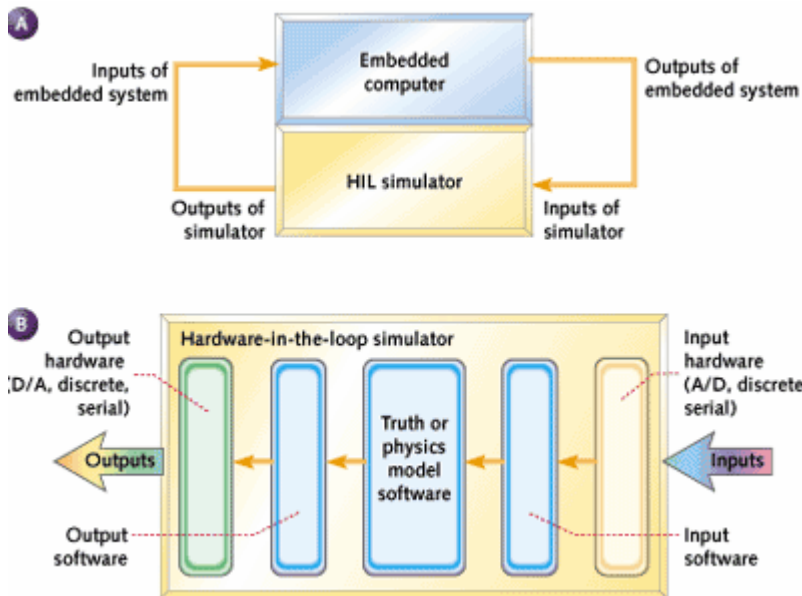
Kuva 1. Reaaliaikasilmoinnin jako lähdeittä mukailien [1]

HIL -simuloinnissa simulointijärjestelmään on kytketty todellisen järjestelmän osa tai komponentti. Tämä komponentti on yleensä ohjausyksikkö (autoteollisuudessa ECU, ilmailussa FADEC), mutta myös muita todellisen järjestelmän osia saatetaan käyttää. Syynä muiden järjestelmän todellisten komponenttien käyttöön saattaa olla esimerkiksi liian vaikea mallinnettavuus tai reaaliaikasilmoitavuus. Toisaalta myös HIL -simuloinnissa yksi tai useampia ohjausjärjestelmän osia voidaan toteuttaa simuloimalla. Yleinen hardware-in-the-loop simuloinnin määrittelevä järjestelmä on esitetty kuvassa 2. Siinä järjestelmän ”koneisto” on simulointimalli ja ohjausyksikkö on todellinen komponentti.



Kuva 2. Hardware-in-the-loop -simuloinnin määrittelevä järjestelmä [4]

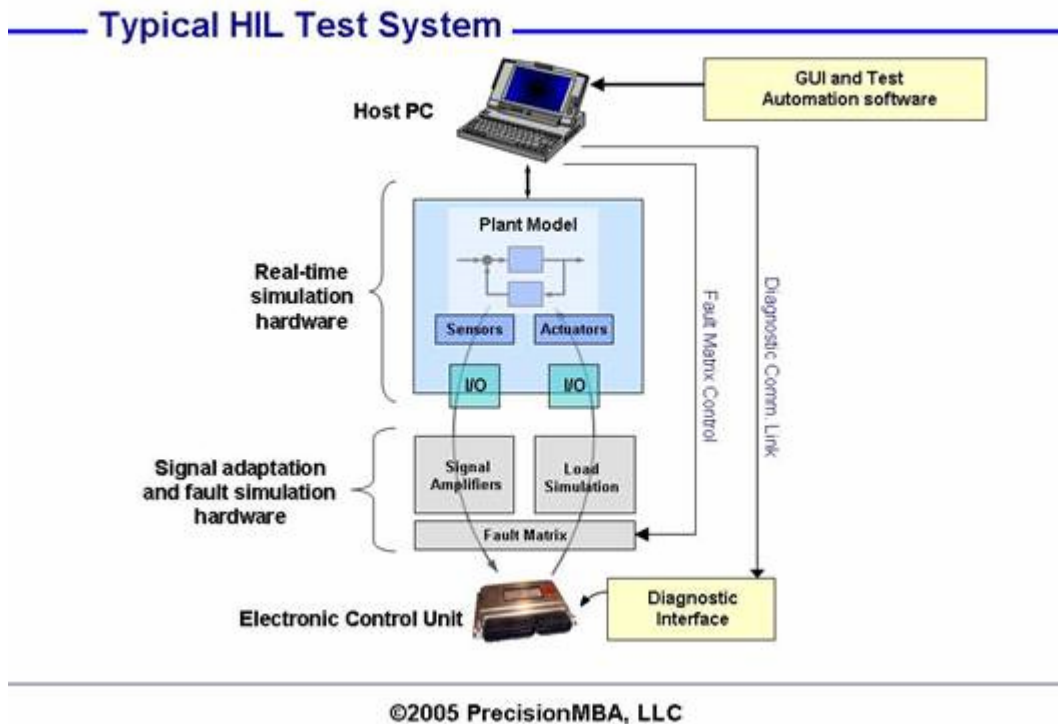
Hardware-in-the-loop -järjestelmän tarkoitus on tarjota kaikki todellisen järjestelmän sähköiset toiminnot ja vasteet ohjausyksikön testaukseen ja harjoittamiseen. Ohjausyksikkö toimii siis aivan kuten ohjatessaan todellista järjestelmää. HIL -simuloinnille saadaan tätä taustaa vasten kolme erityispiirrettä, jotka erottavat sen ohjausjärjestelmän testaamisesta tietokoneella. Ensimmäiseksi ohjelmallisesti tehdyssä toteutuksessa sisään- ja ulostulot ovat vain viivoja tietokoneen ruudulla, kun taas HIL -simuloinnissa ne ovat oikeita signaaleja, tämä on esitettyinä kuvassa 3. Toisekseen HIL -simulointia suoritetaan aina reaaliajassa ja kolmanneksi HIL -simuloinnin sulautettu ohjelma toimii juuri sillä alustalla, jota lopputuotteessakin tullaan käyttämään - työaseman sijaan.



Kuva 3. Yksinkertainen HIL -järjestelmä esitettynä lohkokaaviona (A) sekä HIL -simulaattorin sisäinen toteutus (B) [3].

2.1. HIL -simulointijärjestelmän komponentit

HIL -järjestelmälle voidaan määritellä kahdeksan komponenttia [4], joista järjestelmä koostuu. Jokaisella komponentilla on oma tehtävänsä, jota se suorittaa simulaattoria käytettäessä. Järjestelmän komponentit ja niiden tehtävät, jotka näkyvät kuvassa neljä, ovat seuraavat:



Kuva 4. Tyypilliset HIL -järjestelmän komponentit. [4]

Malli (Plant Model) esittää mitä tahansa, mitä ohjausjärjestelmä ohjaa, esim. moottori, ajoneuvo tai lentokone. Malli pitää usein tehdä itse, mutta jonkin verran yleisimpien komponenttien valmiita malleja esim. Simulink-toteutuksina on saatavilla. Mallin tarkkuuden merkitys kehitystyössä pitää arvioida sovelluskohtaisesti.

Anturimallit (Sensors) ja toimilaitteet (Actuators) sisällytetään usein malliin, mutta ne voivat olla myös erillinen osa sitä. Anturien ja toimilaitteiden mallinnus on haasteellista mm. niiden epälinearisuuksista johtuen. Hitaan vasteen anturimallit voidaankin toteuttaa myös dynaamisina. Joissakin sovelluksissa antureina ja toimilaitteina käytetään myös todellisia komponentteja, tämä kuitenkin saattaa vaatia huomattavasti muuta lisätyötä rajapintojen ja mahdollisten lisälaitteiden suunnittelun muodossa.

Reaaliaika I/O on tärkeä osa järjestelmää reaaliaikaisuusvaatimuksen vuoksi. Usein HIL -järjestelmissä mallin alustana käytetään sulautettujen järjestelmien tietokoneita, sillä Windows -käyttöjärjestelmällä varustettu PC ei ole reaaliaikainen. On kuitenkin mahdollista käyttää PC:tä alustana, jolloin tarvitaan erillinen reaaliaikakäyttöjärjestelmä. Toinen mahdollisuus on käyttää PC:llä erillisiä FPGA I/O -kortteja, jolloin HIL -simulointi on mahdollista.

Kuormat (Loads) HIL- järjestelmässä voivat olla aitoja tai simuloituja. Kuormat suunnitellaan kulloistakin järjestelmää vastaaviksi.

Virheiden kytkentä ja mahdolliset virhetilanteet (Fault Matrix) kannattaa suunnitella erilliseksi osakseen, koska merkittäviä HIL -simuloinnin käyttöalueita ovat virhetilanteiden analysointi sekä ohjausjärjestelmän virheentunnistuskyvyn tutkiminen.

Tällaisia virheitä voivat olla esim. virheelliset tai puuttuvat kytkennät sekä anturien ja toimilaitteiden toimintahäiriöt.

Isäntätietokoneella (Host PC) voidaan käyttää järjestelmän graafista käyttöliittymää, suorittaa automaattisia testiohjelmia ja ylläpitää diagnostiikkaa ohjaimen tilasta. Tietokone voidaan kytkeä reaaliaikajärjestelmään esim. ethernetin tai sarjaportin välityksellä.

Testit automatisoiva ohjelma lasketaan järjestelmän kahdeksanneksi osaksi. Se mahdollistaa isäntätietokoneella testiohjelman automatisoinnin, jolloin testit voidaan esiohjelmoida suoritettavaksi tietyssä järjestyksessä.

2.2. HIL -simuloinnin edut

HIL -simuloinnilla on useita etuja, joista yleisimpinä voidaan mainita seuraavat [1]:

- Ohjausjärjestelmän testaaminen ja kehittäminen ilman todellista prosessia
- Ohjausjärjestelmän testaaminen erityisen vaativissa olosuhteissa (alhainen / korkea lämpötila, suuri kiihtyvyys, sähkömagneettiset häiriöt)
- Järjestelmäkokonaisuuden vikojen testaus (anturit, toimilaitteet, ohjelmistot)
- Vaarallisten tilanteiden testaaminen
- Toistettavuus
- Erilaisten ihminen-kone -rajapintojen testaamisen helppous
- Kustannusten ja ajan säästö

Erityisesti HIL -simuloinnin edut tulevat esiin laajojen sulautettujen järjestelmien ohjaimien testauksessa. Jo pelkästään kaikkien erilaisten sisään- ja ulostulojen kombinaatioiden testaaminen ja analysoiminen saattaa olla valtava työ, puhumattakaan tyypillisestä tilanteesta, jossa ulostulo ei ole vain yksinkertainen funktio kyseisen ajanhetken sisääntuloista vaan sen sijaan siihen vaikuttaa määrättyllä tavalla myös aiempien sisääntulojen kombinaatio. Tällöin haasteeksi muodostuu myös se, kuinka edes määritellään kulloisessakin tilanteessa ”laillinen” ja tosiasiasa mahdollinen sisääntulojen testivektori! Lisäksi järjestelmä saattaa sisältää komponentteja, jotka jo itsessään ovat sulautettuja järjestelmiä, eikä näiden tutkiminen todennäköisesti edes onnistu vain rajoitetuilla ajanhetkillä. Tämä on tyypillistä nykyisissä sulautetuissa järjestelmissä ja myös ratkaisujen täytyy tällöin olla reaaliaikasovelluksia. [3]

2.3. HIL -simuloinnin rajoitukset

HIL -simulaattoria kehitettäessä saattaa esiintyä ongelmia, joista osa liittyy yleisesti simulointiin ja osa on erityisesti HIL -simuloinnille tyypillisiä. Yleinen simuloinnin ongelma on riittävän luotettavan ja yleispätevän mallin muodostaminen. Lopullinen malli onkin aina kompromissi simulointitarkkuuden ja mallin kehittämiseen käytetyn

ajan välillä. Mitä tarkempiin ja laajempien järjestelmien malleihin mennään, sitä enemmän myös mallin raskaus alkaa vaikuttaa.

HIL -simuloinnissa järjestelmät ovat tyypillisesti monimutkaisia ja hyvien tulosten saavuttamiseksi malleista pyritään tekemään mahdollisimman tarkkoja. Tällöin ongelmaksi saattavat muodostua viiveet laskennassa ja niiden vaikutus reaaliaikaisuuteen. HIL -simuloinnissa tulee eteen myös konkreettisempia ongelmia: kuinka ohjain ja muut todellisen järjestelmän komponentit sammutetaan, kun HIL -simulaattori pysäytetään? Laitteethan saattavat jatkaa toimintaansa muusta mallista riippumatta. Mitä tällaisessa pysäytystilanteessa tosiasiaassa tapahtuu? Toiseksi HIL -simulointi ei kerro meille, mitä käytettyjen sulautetun järjestelmän komponenttien sisällä ajon aikana tapahtuu. Se siis ei korvaa mikroprosessoriemulaattoria, logiikka-analysaattoria eikä ohjelmistodebuggeria. [3] Simulointimalli ainoastaan lukee ohjaimen ulostuloja, esittää järjestelmää ja lähettää informaatiota ohjaimelle. Jos simuloinnissa menee jotakin pieleen, ei käytettävissä olevista tiedoista välttämättä pystytä päättämään, mitä osaa ohjelmasta oltiin suorittamassa tai mitkä sisäisten muuttujien tilat vian sattuessa olivat.

HIL -simulointia rajoittaa myös vaatimus nopeista I/O -rajapinnoista sekä tarve käyttää suorituskykyisiä reaaliaikalaitteistoja, mikä usein johtaa valmiin laitteiston korkeaan hintaan. [6] Joissakin tapauksissa vaativan laitteiston rakentamiseen saattaa myös kulua paljon aikaa, jolloin yhdestä yleensä simuloinnilla saavutettavasta edusta joudutaan luopumaan.

3. HIL -SIMULOINNIN HYÖDYNTÄMINEN

Toisinaan sulautettua järjestelmää kehitettäessä tehokkain tapa toteuttaa tuotekehitysprosessi on kytkeä kehitetty ohjausjärjestelmä heti ohjaamaan todellista laitteistoa. Toisissa tapauksissa taas HIL -simuloinnin käyttö on kannattavampaa. Kehitystyön ja testauksen tehokkuuden tutkiminen perustuu tyypillisesti seuraavien kolmen muuttujan arviointiin: kustannus, kesto (aika) ja turvallisuus. Kustannusta tutkittaessa pyritään mittaamaan vaadittujen työkalujen ja työpanoksen aiheuttama taloudellinen kustannus. Tuotekehitysprosessin kestoaika taas vaikuttaa tuotteen markkinoille lanseeraamisen ajankohtaan ja tätä kautta myös sillä on huomattava kustannusvaikutus. Myös kehitysmetodin ja laitteiston turvallisuuden vaikutus pyritään riskianalyysissä analysoimaan kustannuksiin verrannollisena yhtenäisen arviointikriteerin saavuttamiseksi. [5]

Hardware-in-the-Loop -simuloinnin käyttö on todennäköisimmin perusteltu valinta ainakin projekteissa, joissa laajalla sulautetulla järjestelmällä on tiukka kehitysaikataulu, kuten ajoneuvo-, ilmailu- ja sotateollisuus, jotka eivät kaikissa tapauksissa mahdollista täysimittaisia testejä prototyypilaitteistolla.

3.1. HIL -simuloinnin käytännön toteutus

HIL -simulaation toteuttaminen aloitetaan listaamalla sulautetun järjestelmän signaalit [3]. Tämä tehdään listaamalla ohjaimen kaikki sisään- ja ulostulot, niiden alueet, resoluutiot ja näytteenottotaajuudet. Vastaavasti mallin sisään- ja ulostulot tulevat olemaan ohjaimen signaalien ”peilikuva” (Kuva 5). Lisäksi pitää määritellä simuloinnin arkkitehtuuri, eli mitkä osat todellisesta järjestelmästä ohjaimen lisäksi mahdollisesti otetaan mukaan. Todellisten antureiden tai toimilaitteiden lisääminen järjestelmään saattaa lisätä saatujen simulointitulosten luotettavuutta.

Table 1 Signal list for the autopilot example

Signal Name	Range (V)	Range (physical units)	Resolution (units)	Sample Rate (Hz)	Input or Output
Airspeed	0-10	0-200 (m/s)	0.1 m/s	5	Input
Commanded_airspeed	0-10	0-250 (m/s)	0.25 m/s	10	Input
Pitch_rate	0-10	-2-2 (rad/s)	0.004 rad/s	10	Input
Pitch	0-10	0-2p (rad)	0.006 rad	10	Input
Norm_accel	0-10	-10-10 (g)	0.02 g	10	Input
Elev_cmd	0-5	-25-25 (deg)	0.05 deg	10	Output

Table 2 Signal list for a HILS to test the autopilot

Signal Name	Range (V)	Range (physical units)	Resolution (units)	Sample Rate (Hz)	Input or Output
Airspeed	0-10	0-250 (m/s)	0.06 m/s	50	Output
Commanded_airspeed	0-10	0-300 (m/s)	0.08 m/s	50	Output
Pitch_rate	0-10	-2.5-2.5 (rad/s)	0.001 rad/s	50	Output
Pitch	0-10	0-2p (rad)	0.002 rad	50	Output
Norm_accel	0-10	-15-15 (g)	0.008 g	50	Output
Elev_cmd	0-5	-30-30 (deg)	0.02 deg	50	Input

Kuva 5. Esimerkki ohjaimen signaaleista (Table 1) ja HIL -mallin signaaleista (Table 2) lentokoneen autopilottia simuloitaessa [3], vrt. kuva 3.

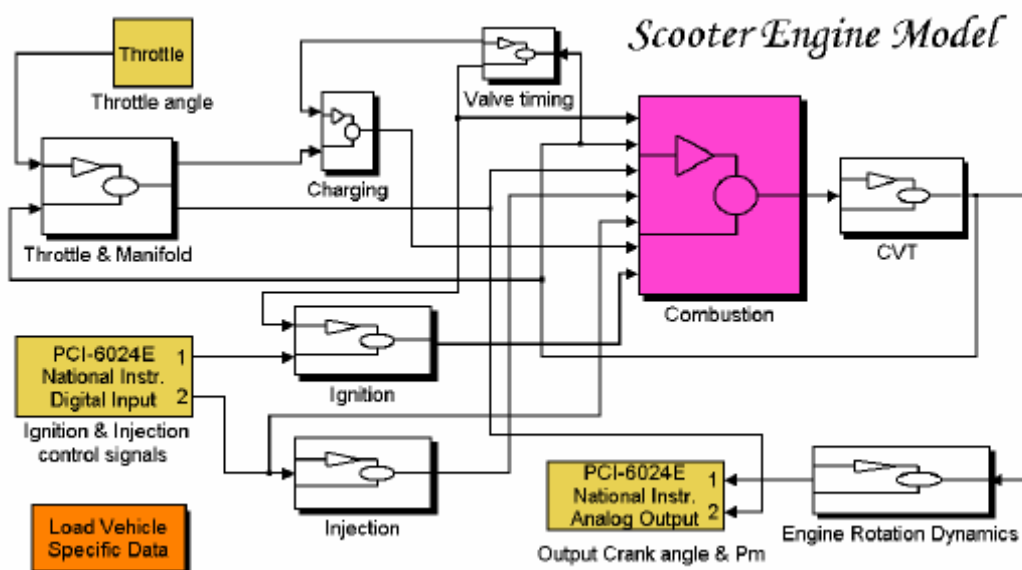
Seuraavaksi valitaan käytettävä simulointialusta, ohjelmisto sekä I/O-toteutus. HIL -järjestelmää toteutettaessa on vain vähäisesti tarjolla valmiita ratkaisuja. Mahdollisia toteutuslustoja ovat dSpace sekä National Instrumentsin Labview, jotka toimivat DSP ja I/O -korttien kanssa tarjoten mahdollisuuden korkeamman tason ohjelmointiin sekä mm. MATLAB ja Simulink -liitynnät. Näissäkin simulointiohjelma pitää itse tehdä, mutta verrattuna C-ohjelmointiin korkeamman tason kielet helpottavat työtä. [3] Käytäntö on osoittanut, että HIL -järjestelmää rakennettaessa järjestelmämallin suorituksen pitää olla viidestä kymmeneen kertaa nopeampi kuin sulautetun ohjaimen. [3] Myös analogisten signaalien digitoinnissa näytteenottotaajuuden tulee laskostumisen välttämiseksi olla vähintään kaksinkertainen signaalin korkeimman taajuuden suuruinen.

Laitteistoa rakennettaessa kannattaa ottaa huomioon myös sen uudelleenkäytettävyys tulevissa projekteissa. Erityisesti anturoinnin ja toimilaitteiden mallinnuksessa sekä I/O:n toteutuksessa tulevien projektien huomioiminen siten, että osioita voidaan pienin muutoksin käyttää uudelleen, säästää huomattavasti kustannuksissa. [3]

4. SOVELLUTUSESIMERKKEJÄ

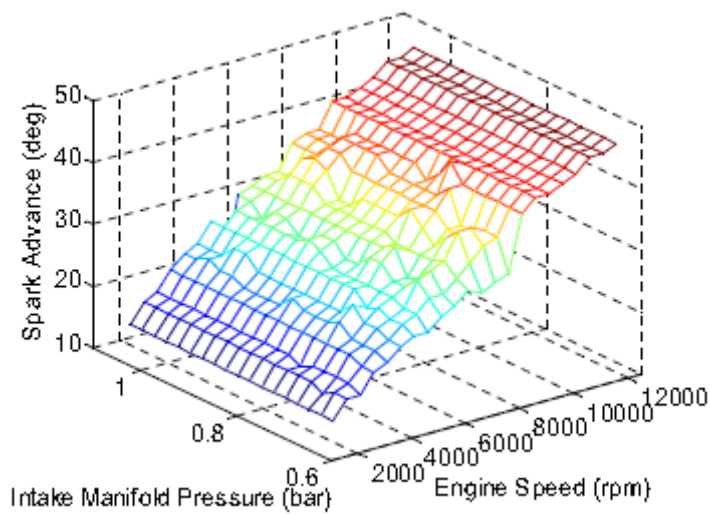
4.1. HIL -simulointi skootterin moottorinohjauksen kehityksessä

Taipein teknillisessä yliopistossa tehdyssä tutkimuksessa [7] kehitettiin HIL -simulaattori skootterin moottorinohjauksen testaukseen. 125-kuutioisen ilmajähdytetyn nelitahtimoottorin (Yamaha XC125) malli toteutettiin Matlab/Simulink -ympäristössä ja sitä käytettiin xPC reaaliaika-alustalla. Moottorinohjaimena käytettiin Motorolan MC 68376 ohjainpiiriä, joka ohjasi virtuaalimoottorin ruiskutusta sekä sytytyksen ajoitusta. I/O -korttina ja rajapintana mallin ja ohjaimen välillä oli käytössä National Instrumentsin PCI-6024E -kortti. Moottorin malli koostui polttoaineen ruiskutuksen ja virtauksen mallista, ilman virtauksen / täytöksen mallista, palamistapahtuman mallista, kitkamallista sekä moottorin pyörimisen dynamiikkamallista. Toteutettu moottorin malli liityntöineen näkyy kuvassa 6.

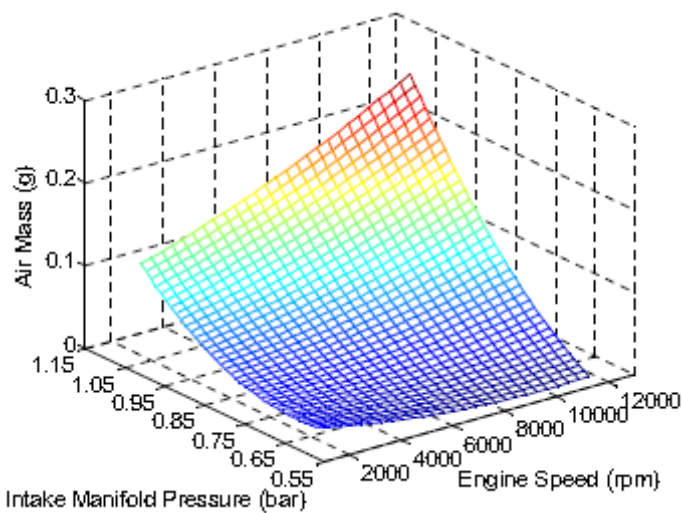


Kuva 6. Skootterin moottorin HIL -toteutus Matlab/Simulink -ympäristössä

Moottorinohjauksessa sytytysennakko parhaan vääntömomentin saavuttamiseksi oli toteutettu kokeellisesti määritetyllä look-up taulukolla imusarjan paineen ja moottorin kierrosluvun funktiona (kuva 7). Vastaavasti polttoaineenruiskutukselle määriteltiin imusarjan paineen ja moottorin kierrosluvun funktiona vaadittu ilmamassa (kuva 8), josta vaadittu polttoaineenruiskutus saadaan halutun täytössuhteen mukaisesti.

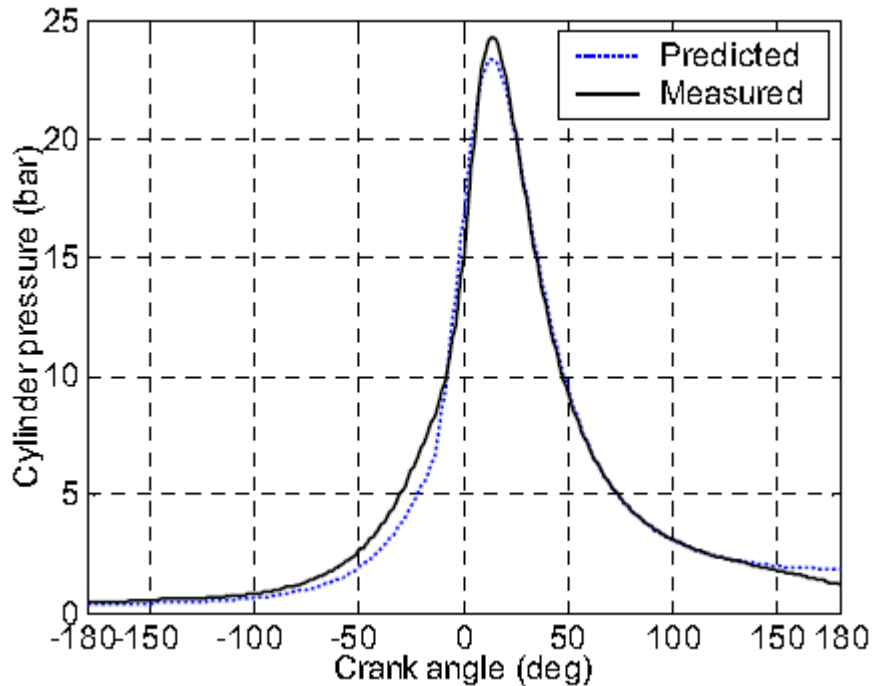


Kuva 6. Lookup-taulukko sytytysennakolle parhaan vääntömomentin saavuttamiseksi.



Kuva 7. Lookup-taulukko vaaditun ilmamassan määrittämiseksi

Moottorin malli verifioitiin koemittauksilla tasakierrosluvulla 8000 rpm sekä täydellä kaasulla kiihdytettäessä moottorin kierrosluvulta 3500 rpm kierrosluvulle 8500 rpm. Esimerkkinä mallin toimivuudesta kuvassa 8 sylinterin paine työkierron aikana kierrosluvulla 8000 rpm.

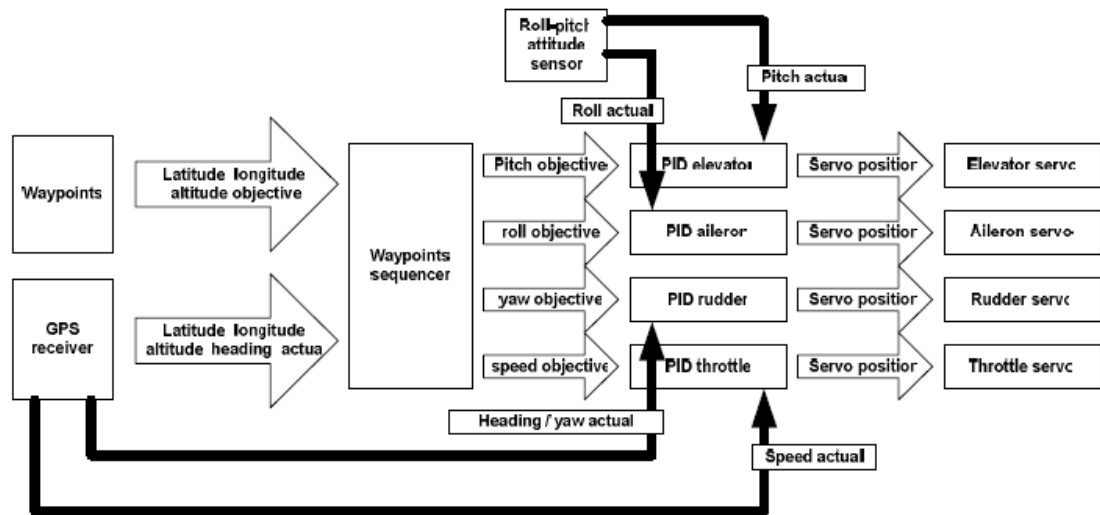


Kuva 8. Mallin ennustama ja mitattu sylinterin paine työkierron aikana kierrosnopeudella 8000 rpm.

Toteutettua HIL -simulaattoria sovellettiin kahteen kehityskohteeseen. Mallin avulla tutkittiin sytytysennakon vaikutusta sylinterin paineisiin ja pyrittiin optimoimaan sytytysennakkoa. Toisessa kohteessa hyödynnettiin tätä tutkimusta kytkemällä skootterin dynamiikkamalli simulaattoriin ja tutkittiin skootterin nopeuksia, kiihtyvyyksiä ja polttoaineenkulutusta. Tutkimus onkin hyvä esimerkki siitä, mitä Hardware in the Loop -simulointi mahdollistaa tuotekehityksessä.

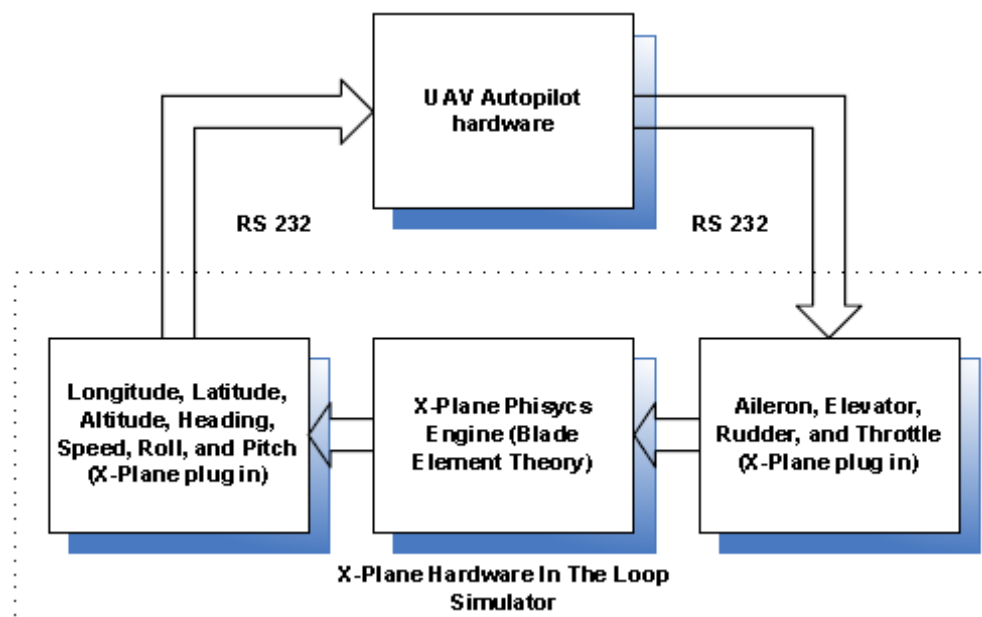
4.2. HIL -simulointi UAV:n autopilotin kehityksessä

Bandungin sähkö- ja informaatiotekniikan instituutissa Intiassa [8] tutkittiin HIL -simuloinnin soveltuvuutta UAV:n (miehittämätön ilma-alus) autopilotin kehittämiseen. HIL -simulointi soveltuu kohteeseen erityisen hyvin, sillä kenttäkokeiden toteuttaminen prototyyppiosilla on vaikeaa ja mahdolliset ohjausjärjestelmän viat voivat johtaa kalliiden laitteiden vahingoittumiseen.



Kuva 9. UAV:n autopilotin säätöjärjestelmä

Autopilotin säätöjärjestelmä (kuva 9) koostuu kahdesta osasta, joista toinen laskee gps-reittipisteitä ja toinen osa sisältää PID-säätimet koneen nopeudelle ja ohjainpinoille. HIL -simulaattorin malli (kuva 10) tehtiin vastaamaan todellista lentokonetta ja siihen sisällytettiin aluksen nopeudensäätö ja siivekkeet sekä korkeus- ja sivuperäsimet. Toteutuslueksaksi valittiin X-Plane -simulointiohjelmisto, sillä sen avulla voidaan tehdä fysiikkamalli suoraan lentokoneen ja sen ohjainpintojen geometriasta. X-Plane osaa tämän jälkeen automaattisesti laskea ilman virtauksen ja ohjainpintojen vaikutukset. I/O -liityntänä käytettiin X-Planen pluginia ja kommunikointi autopilotin kanssa toteutettiin RS232 sarjaporttiliitynnällä.



Kuva 10. UAV Autopilotin HIL -simulointi

HIL- simulaattoria käytettiin maa-aseman ohjelmistokehityksessä sekä säätäjän kehitystyössä. Simulaattorin avulla viritettiin PID-säätäjän parametrit yritys ja erehdys menetelmällä. Analyyttinen parametrien määrittäminen olisi työn tekijöiden mielestä ollut kyseiselle laitteistolle huomattavasti vaikeampaa matemaattisen mallin monimutkaisuudesta johtuen. Simuloitaessa huomattiin myös säätimen virransyötön reguloinnin epästabiilisuus, mikä käytännön testeissä olisi pahimmillaan saattanut aiheuttaa koneen maahansyöksyn. Kyseisessä tutkimuksessa HIL-simulointiin käytetty aika vastasi 150 tuntia käytännön lentotestejä ja tämän jälkeen tarkoituksena on kokeilla autopilotin toimintaa oikeassa ympäristössä.

5. YHTEENVETO

Tässä tutkielmassa esiteltiin Hardware in the loop -simulointia, joka on reaaliaikaisimuloinnin muoto. Sitä käytetään hyvin yleisesti esim. auto- ja lentokoneteollisuudessa. HIL -simulointia suositaan sen mahdollistaessa todellisen ohjausjärjestelmän testaamisen ilman todellisten prototyyppien rakentamista. HIL -simulointi asettaa kehitettävälle järjestelmälle tiukat vaatimukset reaaliaikaisuuden suhteen, ja on moniin muihin simulointitapoihin verrattuna jonkin verran haastavampi sekä usein myös kalliimpi toteuttaa. Hyvin toteutetulla HIL -järjestelmällä on saatavissa myös laadukkaita simulointituloksia. Sulautettua järjestelmää suunniteltaessa on kuitenkin tärkeää tapauskohtaisesti analysoida perusteet simuloinnin käytölle sekä simulointitavan valintaan vaikuttavat tekijät halutun tuloksen saavuttamiseksi.

LÄHTEET

- [1] Robert H. Bishop. 2006. Mechatronics: an introduction. 1. painos. CRC Press. 312 s.
- [2] Tae-Hyeong Lim, Hyun-Chul Cho, Hong-Sun Lee, Soon-Yong Yang. 2005. Development of Hardware in the Loop System (HILS) for Hydraulic Excavator. Seminaariesitelmä. 6 s.
- [3] Martin Gomez. 2001. Hardware-in-the-Loop Simulation. Embedded.com [WWW]. [Viitattu 30.4.2009]. Saatavissa: <http://www.embedded.com/story/OEG20011129S0054>
- [4] PrecisionMBA, LLC. 2005. Hardware-In-the-Loop. [WWW]. [Viitattu 29.4.2009]. Saatavissa: http://www.precisionmba.com/hardware_in_the_loop.htm
- [5] Wikimedia Foundation, Inc. 2009. Hardware-in-the-loop Simulation. [WWW]. [Viitattu 29.4.2009] Saatavissa: <http://en.wikipedia.org/wiki/Hardware-in-the-loop>
- [6] Marcel Groothuis. 2004. Distributed HIL simulation for Boderc. Masters Thesis, University of Twente. 124 s.
- [7] Yuh-Yih Wu, Bo-Chiuan Chen, Feng-Chi Hsieh, Ming-Lung Huang ja Ying-Huang Wu. 2006. Development of Hardware-In-the-Loop Simulation for Scooter Engine Control. Konferenssiesitelmä. 11 s.
- [8] Widyawardana Adiprawita, Adang Suwandi Ahmad, Jaka Sembiring. 2007. Hardware In The Loop Simulation for Simple Low Cost Autonomous UAV Autopilot System Research and Development. Konferenssiesitelmä. 4 s.