

Tartu Ülikool
Matemaatika-informaatikateaduskond
Arvutiteaduse instituut
Infotehnoloogia eriala

REAALAJASÜSTEEMI NÕUDED JA DISAIN:

Auto püsikiiruse hoidja

Autor: Rait Ool
Autor: Mart Vellak
Juhendaja: Aivo Reinart

Tartu 2010

Sisukord

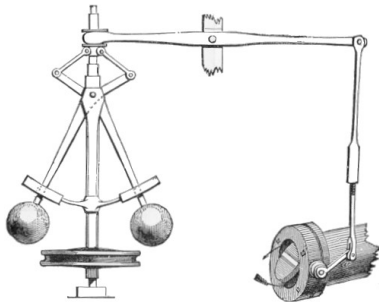
1. Sissejuhatus.....	3
1.1 Ajalugu.....	3
1.2 Ulatus.....	3
1.3 Mõisted ja lühendid.....	3
2. Üldkirjeldus.....	4
3. Spetsiifilised nõuded.....	5
3.2 Klassid/Objektid.....	6
3.2 Kasutuslood.....	6
3.3 Reaalajalised nõuded.....	8
4. Disain.....	9
4.1 UML diagramm.....	9
4.1 Ajaliste seoste analüüs.....	10
4.2 Koostööskeemid.....	10
4.4 Jõudlustesti korraldamise plaan.....	12
Lisad.....	14

1. Sissejuhatus

Püsikiiruse hoidja on süsteem, mis automaatselt kontrollib mootorsõiduki kiirust. Süsteem võtab üle kiirendamise ja pidurdamise süsteemi, et tagada ühtlane kasutaja poolt valitud kiirus. Antud kodutöö eesmärgiks on kirjeldada lühidalt püsikiiruse hoidja tööpõhimõtet. Keskendume põhiliselt adaptiivsele püsikiiruse hoidjale, mis ühtlasi hoiab ohutut vahemaad eesoleva sõidukiga.

1.1 Ajalugu

Kõige esimese püsikiiruse hoidja leiutati väidetavalt firma Peerless Motors'i poolt aastal 1912. See töötas tsentrifugaaljõu toimet. Seda süsteemi illustreerib joonis 1. Moderne kiirusehoidja leiutati 1945. aastal Ralph Teetori poolt ning esimest korda auto peal oli 1958. aastal.



Joonis 1. Tsentrifugaaljõu toimet töötav püsikiiruse hoidja.

1.2 Ulatus

Kiirusehoidja abil tehakse arvuti toimet juhitaavaks mootorsõiduki püsikiirus. Kuna antud töös käsitleme ka adaptiivset kiirusehoidjat, siis on süsteemi integreeritud nii kiirendamise kui ka pidurdamise võimalused. Tegemist on juba leiutatud masstoodas oleva seadmega, seega anname lihtsalt ülevaate süsteemi toimimisest.

1.3 Mõisted ja lühendid

ACC – Adaptiivne püsikiiruse hoidja (ingl. k. *adaptive cruise control*)

DAC – digital-analog konverter (ingl. k. *digital-analog converter*)

2. Üldkirjeldus

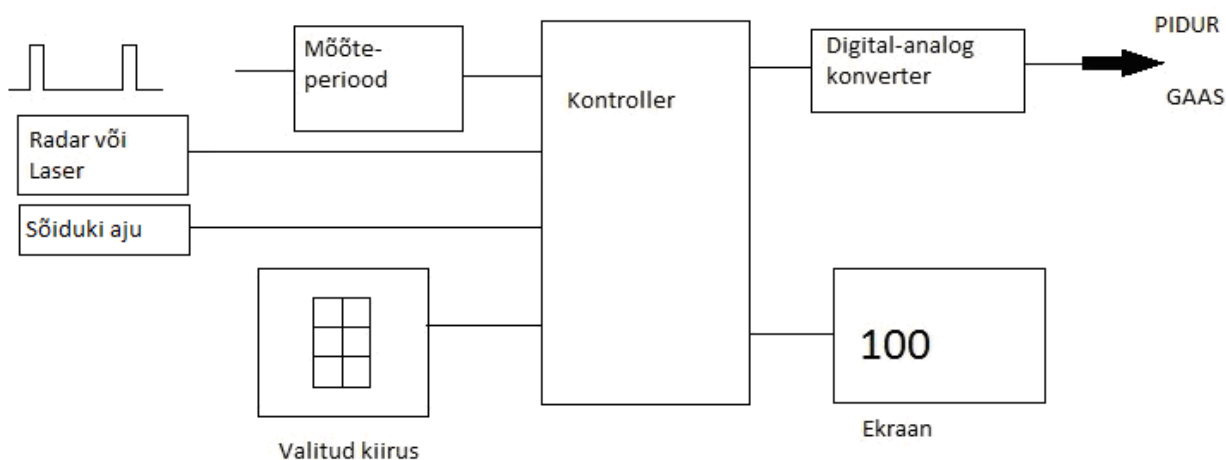
Joonisel 2 on kujutatud üldist ACC süsteemi. Kontrollrist vasakul asuvad sisendid ning paremal on väljundid.

Sisendid:

- Mõõteperiood on ajavahemik, mille järel kontrollitakse sõiduki hetkekiirust ja kaugust eesmisest objektist.
- Radari või laseri abil määratakse ära eesmise objekti kiirus ja/või kaugus.
- Sõiduki ajast loetakse sõiduki hetkekiirus ning see informeerib kontrollit gaasi ja piduri manuaalsest kasutamisest.
- Valitud kiirus annab kontrollile teada, millist kiirust vaja hoida.

Väljundid:

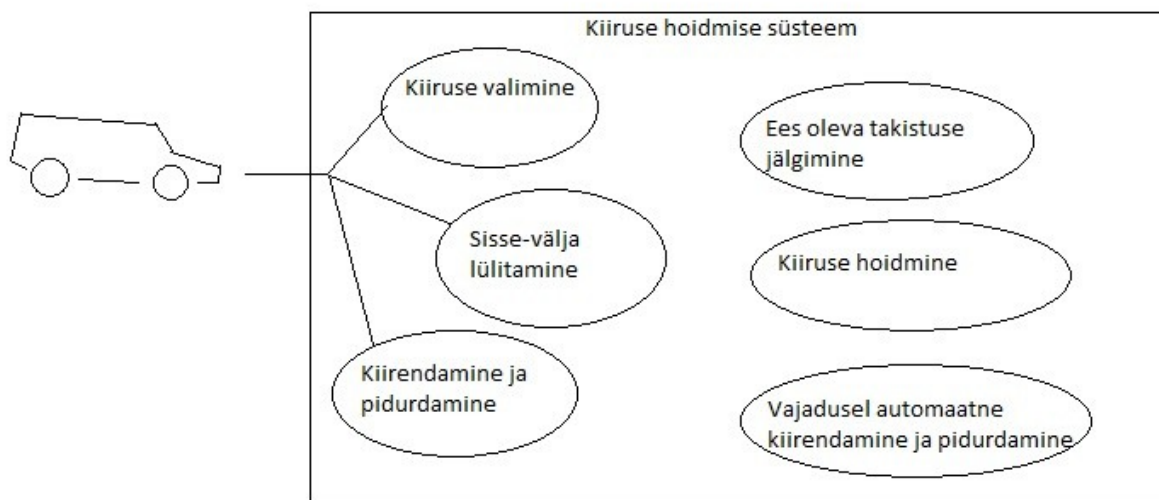
- DAC muudab kontrolleri poolt edastatud info auto juhtimisseadmetele arusaadavaks.
- Ekraanil esitatakse auto valitud kiirus.



Joonis 2. Üldine kiirusehoidja skeem

3. Spetsiifilised nõuded

3.1 Kasutuslugude üldskeem



Joonis 3. Kasutuslugude üldskeem

Kiiruse valimine: autojuht valib kiiruse, mida süsteem peab säilitama. V.a. juhul kui ees on teine auto, siis ACC hoiab ohutut vahemaad olenemata seatud kiirusest.

Sisse-välja lülitamine: vaikimisi pole süsteem aktiveeritud. Vastava nupu abil saab autojuht süsteemi käivitada.

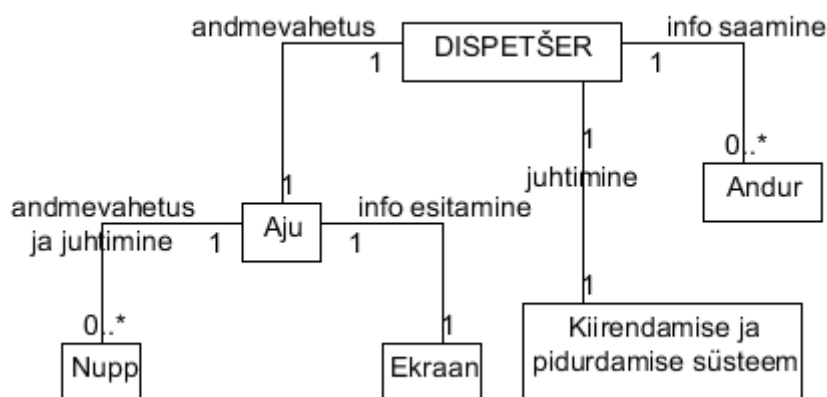
Kiirendamine-aeglustamine: juht saab ise manuaalselt kiirendada (mööda sõita) ning pärast toimingut taastab ACC endise oleku. Pidurdamisel lõpetab ACC töötamise ja võimalusel koheselt soovitud kiirust valida või jätkata eelmist režiimi.

Ees oleva takistuse jälgimine: ACC määrab ees oleva sõiduki kiiruse ja ohutu vahemaa ning hoiab seda olenemata valitud kiirusest.

Kiiruse hoidmine: ACC saab infot sõiduki hetkekiiruse kohta sõiduki ajult ning vastavalt sellele automaatselt kiirendab või pidurdab.

Vajadusel automaatne kiirendamine ja pidurdamine: kui hetkekiirus on väiksem valitud kiirusest, siis süsteem automaatselt kiirendab nõutud kiiruseni. Kui teele tuleb takistus, siis ACC automaatselt pidurdab sõltumata valitud kiirusest.

3.2 Klassid/Objektid



Joonis 4. ACC klassidiagramm

Andur: eesoleva takistuse kauguse ja kiirusega seotud protsessid.

Nupp: süsteemi oleku muuttustega seotud protsessid.

Ekraan: juhi teavitamisega seotud protsessid.

Sõiduki aju: sõiduki enda info (kiiruse) saamine.

Kiirendamise ja pidurdamise süsteem: protsessid seoses sõiduki kiiruse muutmisega.

3.2 Kasutuslood

ACC-l on 4 erinevat kasutuslugu.

- Süsteemi käivitamine ja kiiruse valimine
- Ette tuleb takistus ja süsteem pidurdab sõltumata sisestatud kiirusest
- Juht pidurdab manuaalselt
- Juht kiirendab manuaalselt

Need neli kasutuslugu on püsikiirushoidja põhilised funktsioonid, mida see peab tavaolukordades täitma. Kaks esimest on täisautomaatsed tegevused ja viimased nõuavad ka juhivoolset sekkumist.

1. Süsteemi käivitamine ja kiiruse valimine:

- Juht käivitab sõiduki
- Juht käivitab kiirusehoidja
- Valib nupust kiiruse
- Dispetšer saab nupu käest info
- Dispetšer saab ajult hetkkiiruse
- Vastavalt sellele reguleerib kiirendamist/pidurdamist
- Väljastab ekraanile valitud kiiruse

2. Ette tuleb sõiduk ja süsteem pidurdab sõltumata sisestatud kiirusest

- Juht käivitab süsteemi ja valib kiiruse (vaata 1. kasutuslugu)
- Ees sõidab teine sõiduk
- Dispetšer saab anduritelt infot sõiduki kohta
- Vastavalt andurilt saadud infole aeglustatakse sõidukit
- Süsteem hoiab ohutut vahet eesoleva sõidukiga
- Sõiduki eest kadumisega taastub normaalne töö

3. Juht pidurdab manuaalselt

- Juht käivitab süsteemi ja valib kiiruse (vaata 1. kasutuslugu)
- Juht pidurdab sõidukit manuaalselt
- Süsteem lülitab ACC puhkeolekule
- Juhile antakse täielik kontroll sõiduki üle
- Juhil on võimalus valitud kiirust jätkata vajutades resume nuppu
- Süsteem kiirendab valitud kiirusele (läheldes esimesest kasutusloost)

4. Juht kiirendab manuaalselt

- Juht käivitab süsteemi ja valib kiiruse (vaata 1. kasutuslugu)
- Juht kiirendab manuaalselt
- ACC hetkel ei tee midagi
- Kui kiirendamine lõppenud, siis ACC taastab algse kiiruse

3.3 Reaalajalised nõuded

Süsteem töötab ranges reaalajas. Seega peavad olema mõningad punktid täidetud:

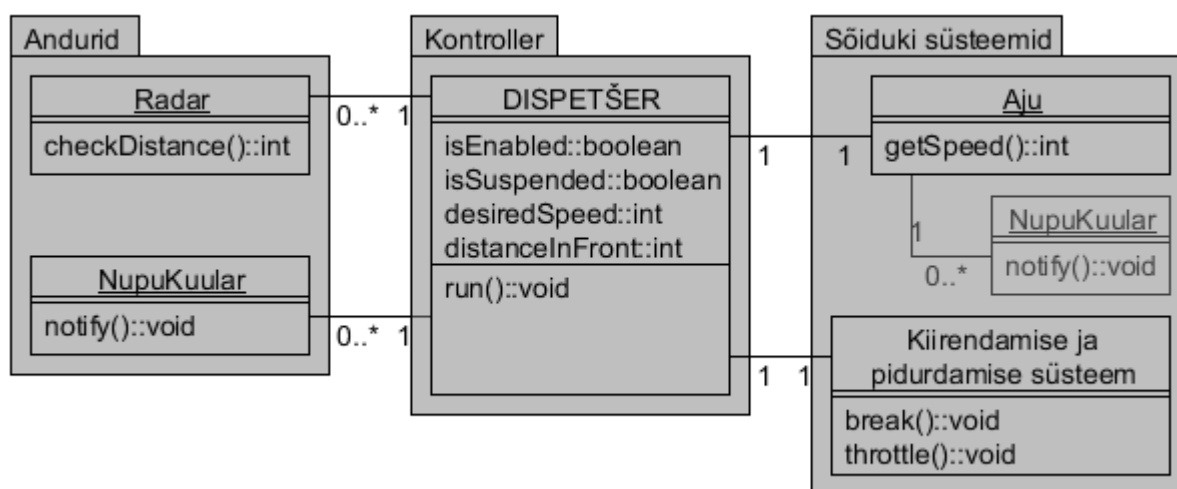
- Kosteaaeg on rangelt nõutud. Ning ei tohiks ühelgi juhul ületada kindlat piiri.
- Ohutus peab olema alati tagatud, kuna inimesed on tihedas suhtluses antud süsteemiga.
- Pidurdusprotsessidel on suurem prioriteet kui kiirendamisprotsessil.
- Vigade avastamine peab olema automaatne ning kiire.
- Vead jagatakse ohtlikuks ja mitte, ohtlikud on seotud pidurdussüsteemi või kiirendamissüsteemiga.
- Vea avastamisel tuleb kasutajat teavitada.
- Ohtliku vea avastamisel lõpetab süsteem koheselt töö ja juht saab sõiduki üle täieliku kontrolli.

Kõik protsessid töötavad katkestuste abil.

4. Disain

4.1 UML diagramm

Süsteemi saab sisuliselt 3-ks erinevaks osaks jaotada. Esimene osa on andurid, mis koosneb radarist ja nupukuularist. Radari ülesandeks on jälgida, mis toimub autost eespool ja sellest teavitada kontrolleri. Nupukuular saadab juhipoolt saadud info kontrolleri, et see oskaks vastavalt käituda. Süsteemi teine osa on kontrolleri, mille ülesandeks on püsikiirushoidja juhtimine, saades infot anduritelt ja see edasi anda sõiduki süsteemidele. Kolmas osa on sõidukisüsteemid. See koosneb sõiduki ajast ja kiiruse kontrollimise süsteemist. Aju ülesanne on jagada infot sõiduki kiiruse kohta, kiiruse kontrollimise süsteem peab vastavalt kontrolleri poolt saadud infole kas kiirendama või pidurdama.



Joonis 5. Süsteemi UML diagramm

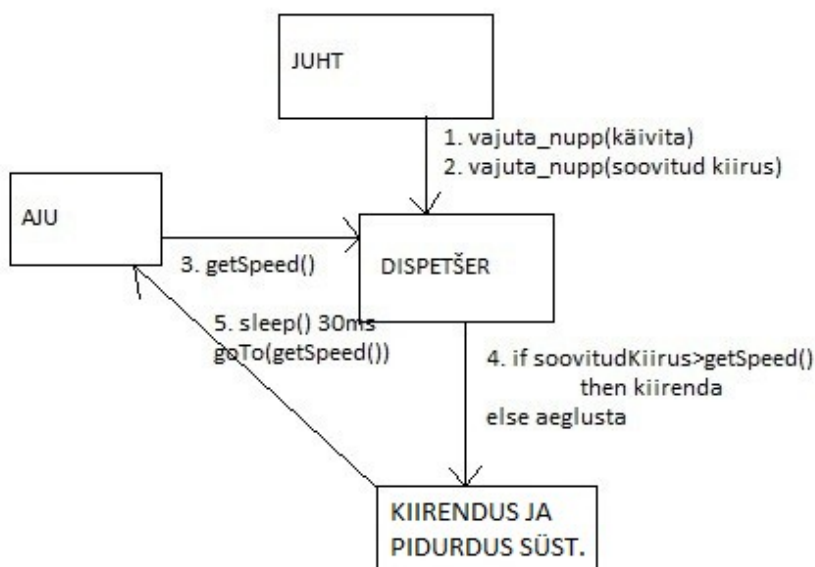
4.1 Ajaliste seoste analüüs

Kuna osadel autodel aktiveeritakse soovitud kiiruse muutmisega kiiruse püsihoidja süsteem, siis antud näites käsitlemegi sellist olukorda. Võtame ette järgmise olukorra: süsteem käivitatakse, teele ilmub objekt, radarilt saadakse esimene vahekaugus ning teise vahekauguse saamisel ei tule radarilt vastust. Järelikult süsteem peaks end välja lülitama ja kasutajat teavitama.

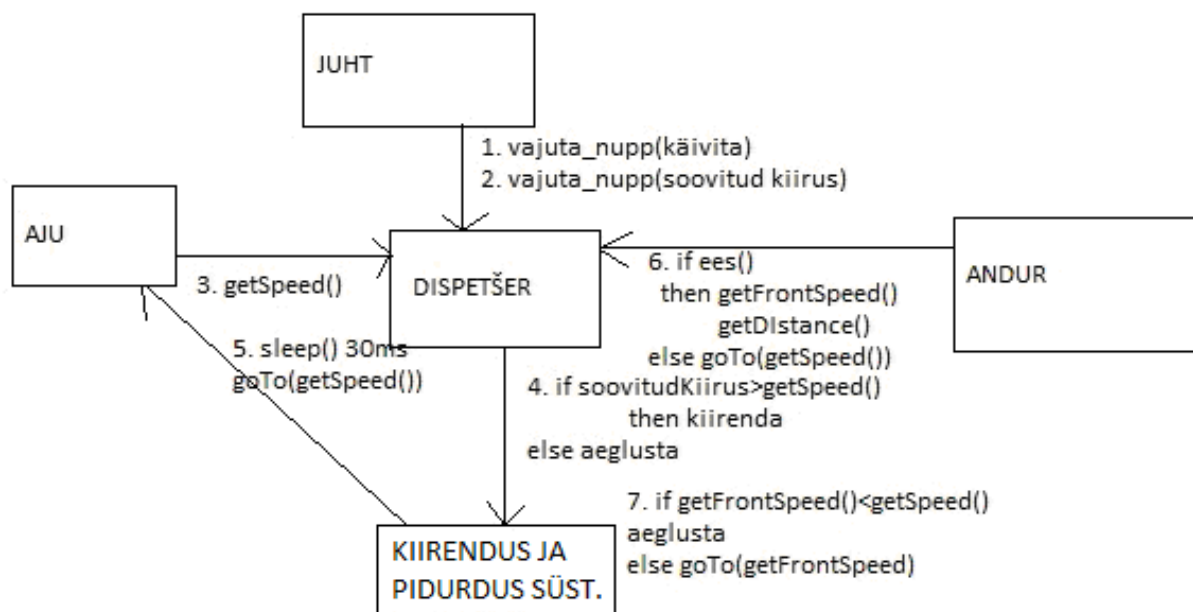
Aktiveerimise nupule vajutamise ja selle käsu rakendumiseks kulugu meil 20ms ning seejärel käivitatakse kogu süsteemi 40 ms jooksul. Radar saadab kontrolleriile kauguse eesmisest objektist ning 10 sekundi pärast tõlgendab dispetšer tulemuse ja küsib radarilt uut mõõtmist, et süsteem saaks eesmise sõiduki suhtelise kiiruse arvutada. 60 ms jooksul ei tule radarilt vastust, seega programm peab seda tõlgendama rikkena. Kontrolleri saadab ekraanile veateate ning see rakendub 10 ms pärast. Seejärel lülitatakse kogu süsteem 40 ms jooksul välja. Järelikult kogu selle protsessi peale läheb kokku $20 + 40 + 10 + 60 + 10 + 40 = 180$ ms.

Tavaliselt peaks radar teise mõõtekauguse saatma 10 ms jooksul ning edasise käsu tõlgendamine ja selle edasisaatmine toimub 20 ms vältel. Seega töötava süsteemi puhul kulub ainult $20 + 40 + 10 + 20 + 10 = 100$ ms.

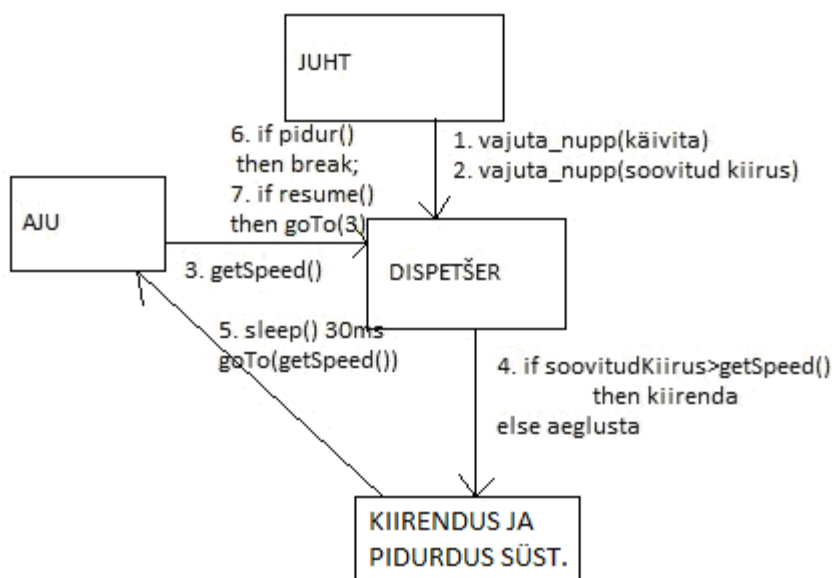
4.2 Koostööskeemid



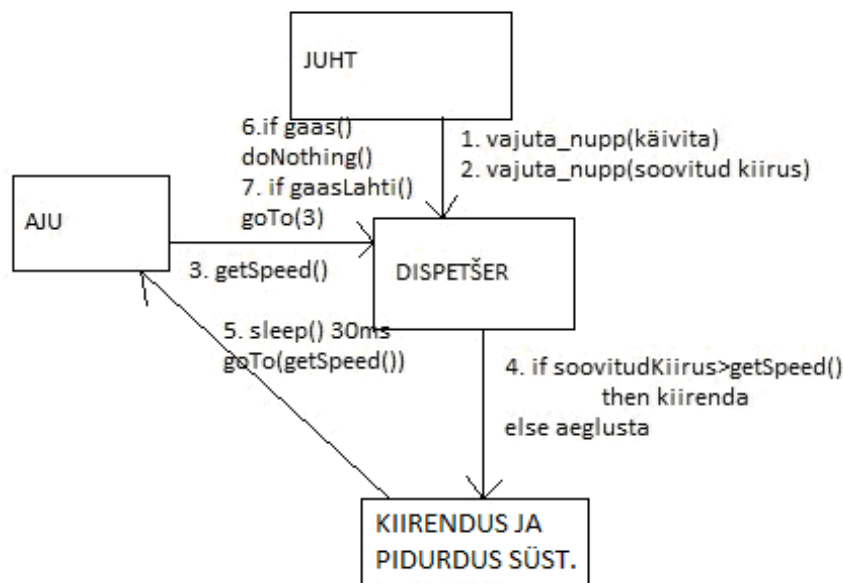
Joonis 6. Koostööskeem 1. kasutusloole



Joonis 7. Koostööskeem 2. kasutusloole



Joonis 8. Koostööskeem 3. kasutusloole



Joonis 9. Koostööskeem 4. kasutusloole

4.4 Jõudlustesti korraldamise plaan

Lifitsüsteemi valideerimine toimub demonstratsioonide, inspeksioonide, analüüside, simulatsioonide ja täppismõõtmiste abil. Tuleb läbi töötada kõik kasutusjuhud ning lisaks vaadata, kuidas süsteem töötab vigade tekkimisel (näiteks olukord, kus dispetšer ei saa kiirendamise ja pidurdamise süsteemi kontrollida). Süsteemi valideerimiseks kasutatakse antud ACC prototüüpe, paigaldatakse test-sõidukitesse ning testitakse 3 kuud.

Täppismõõtmisteks kasutatakse vastavaid mõõtmisinstrumente. Testitakse erinevaid liftisüsteemi andureid, et veenduda nende täpsuses. Simulatsiooniks kasutatakse kas vababaraalist simulaatorit või luuakse vastav simulaator ise, kui olemasolevad programmid ei vasta soovitud tingimustele.

Kasutatud allikad

1. Cruise control - Wikipedia, the free encyclopedia, URL = http://en.wikipedia.org/wiki/Cruise_control (külastatud 10. aprill 2010)
2. Centrifugal governor - Wikipedia, the free encyclopedia, URL = http://en.wikipedia.org/wiki/Centrifugal_governor (külastatud 10. aprill 2010)
3. Adaptive Cruise Control System Overview, URL = http://sunnyday.mit.edu/safety-club/workshop5/Adaptive_Cruise_Control_Sys_Overview.pdf (külastatud 10. aprill 2010)

Lisad

Demonstratsioon ACC süsteemi simulatsioonist: http://www.youtube.com/watch?v=uiByqe5r_Dw

Demonstratsioon ACC süsteemi simulatsioonist riist- ja tarkvara kombineerimisel: http://www.youtube.com/watch?v=X4GiAg2_lOI