

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKATEADUSKOND  
ARVUTITEADUSE INSTITUUT  
Infotehnoloogia eriala

# **Orbitaaldünaamika teegi OREKIT kasutamine satelliitside kontaktide optimaalseks planeerimiseks**

Bakalaureusetöö (6 EAP)

Autor: Mart Vellak

Juhendajad: Urmas Kvell,

Lauri Kimmel,

Tõnis Eenmäe

Autor: .....	“.....“ mai	2013
Juhendaja: .....	“.....“ mai	2013
Juhendaja: .....	“.....“ mai	2013
Juhendaja: .....	“.....“ mai	2013
Lubatud kaitsmisele		
Professor: .....	“.....“ .....	2013

Tartu 2013

# Sisukord

Sissejuhatus .....	3
1. Eesti tudengisatelliit ESTCube-1 .....	5
2. Satelliidi jälgimine .....	7
2.1. Satelliidi jälgimise tarkvara .....	8
2.2. Erinevad ülennu strateegiad .....	13
2.3. Satelliitide ülendlude optimeerimine .....	17
3. Metoodika .....	18
3.1. Testimisplaan .....	18
3.2. Üks satelliit ja mitu maajaama juhtum .....	18
3.3. Kontakti arvutamine .....	19
3.4. MCS süsteem .....	20
3.5. Satelliidi ülendlu iseloomustavad parameetrid .....	22
4. Tulemused .....	24
4.1. Kontakti arvutamise moodul .....	24
4.2. Testimine .....	25
5. Kokkuvõte .....	31
Summary .....	33
Viited .....	35
Lisa 1. Kontakti arvutamise mooduli kood .....	37
Lisa 2. Kontakti arvutamise mooduli testiplaan .....	38

## Sissejuhatus

7. mai varahommikul, kui kanderaketist VEGA eraldus edukalt satelliit ESTCube-1, sai Eesti ametlikult kosmoseriigiks. Meie kosmosemissiooni edukaks opereerimiseks kasutame satelliitsidejaamu, mille juhtimiseks on kasutusel vabavaralisel raamistikul Hummingbird põhinev missioonijuhtimistarkvara (MCS – ingl k. Mission Control Software). Raamistik Hummingbird on laialdase kasutusala tarkvaraplatvorm seadmete töö jälgimiseks ja juhtimiseks. Paraku on raamistikust puudu satelliitside jaoks oluline komponent, mis võimaldaks ette planeerida sidekontakte satelliitide ja maajaamade vahel.

Käesoleva bakalaureuse töö peamiseks eesmärgiks on luua tarkvaramoodul, mis võimaldab leida missiooni opereerimiseks ja planeerimiseks vajalikud kontaktiajad satelliitsidejaamade ja satelliitide vahel koos neid kontakte iseloomustavate parameetritega. See võimaldaks operaatoritel optimaalselt ära kasutada kvaliteetsed sideaega. Loodav lahendus on eriti oluline madalal Maa orbiidil olevate satelliitide puhul, näiteks satelliidil ESTCube-1, sest sellistel orbiitidel olevate satelliitide arv kasvab väga kiiresti ja samas on kvaliteetne sideaeg nende satelliitidega väga piiratud. Sidekontaktide planeerimise mooduli lisamine vabavaralisse raamistikku Hummingbird lihtsustaks oluliselt teistel ülikoolidel ja väikefirmadel oma satelliidimissioonide läbiviimist Maa lähedastel orbiitidel.

Lisaks kontaktiaegade leidmisele on operaatorite igapäevaseks väljakutseks piiratud satelliitsidejaamade ressursside optimaalne kasutamine mitmete satelliitide missioonide läbiviimiseks. Optimeerimist vajavad juhud võib jaotada kolmeks: kui kasutada on üks satelliit ja mitu maajaama, üks maajaam ja mitu satelliiti või mitu maajaama ja mitu satelliiti. Antud töös keskendutakse ühe satelliidi ja mitme maajaama sidekontaktide optimeerimisele, käsitledes teoreetilisi lahendusi, mis võtavad arvesse sideoperaatorite vajadusi.

Antud bakalaureusetöö eesmärgid on:

- luua tarkvaramoodul, mis arvutab maajaama ja satelliidi sidekontakti ajad,
- kavandada tarkvaramooduli osa, mis arvutab sideseansi kvaliteeti iseloomustavad parameetrid,
- saavutada vähemalt 80% koodi kattuvus ühiktestidega ja viia läbi katsetused reaalse maajaama ja satelliidiga,
- pakkuda välja võimalikud optimeerimisstrateegiad ühe satelliidi ja mitme maajaama puhul.

Bakalaureusetöö on jaotatud neljaks peatükiks. Esimeses peatükis antakse ülevaade Eesti esimesest satelliidist ESTCube-1 ja selle missioonist. Teine peatükk annab ülevaate varem loodud satelliitide jälgimise tarkvarast ning optimeerimise probleemist. Kolmandas peatükis on toodud töö metoodika. Neljandas peatükis antakse ülevaade töö tulemustest. Lisadena on esitatud programmi kood ja testiplaan.

# 1. Eesti tudengisatelliit ESTCube-1

ESTCube-1 on Eesti esimene satelliit, mis jõudis 7. mai varahommikul maalähedasele orbiidile Euroopa Kosmoseagentuuri kanderaketiga VEGA. ESTCube-1 on 10 x 10 x 10 cm mõõtmetega kuupsatelliit, mis kaalub ligikaudu 1 kg [1]. Oma missioonil peab satelliit täitma mitmeid erinevaid funktsioone, mille jaoks on satelliidil eraldi alamsüsteemid:

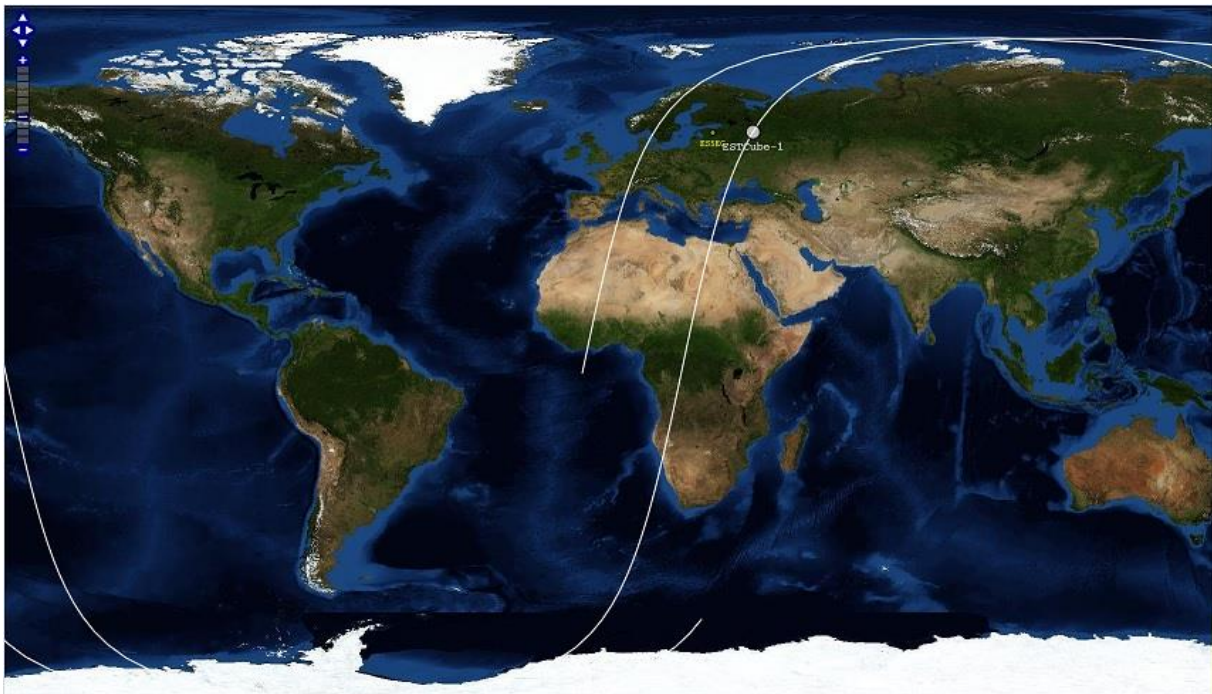
- ADCS – satelliidi asendi määramise ja kontrollimise süsteem
- CDHS – käskude ja andmehalduse süsteem
- CAM – pardal olev kaamera süsteem
- COM – raadioside süsteem
- EPS – elektrienergia haldamise süsteem
- PL – päikesepurje eksperimendimoodul
- STR – satelliidi raam ja külgmised paneelid

Satelliidi missiooni peamiseks eesmärgiks on esimest korda kosmoses katsetada elektrilist päikesepurje [2,3]. Elektriline päikeseपुरi koosneb pikkadest traatidest, mis keritakse kosmoses lahti ja mis püüavad päikese tuult, et satelliidi orbiiti muuta. Päikeseपुरi annaks võimaluse kosmoses ringi liikuda praktiliselt ilma kütust kasutama, võimaldades pikemaid ja kaugemaid kosmosereise [4].



*Joonis 1. Tudengisatelliit ESTCube-1. Autor: Riina Varol.*

Lisaks satelliidi alamsüsteemidele on missiooni edukaks läbiviimiseks vajalik ka maapealne tugi, milleks on sidejaam ja missioonijuhtimistarkvara MCS. Madalal Maa orbiidil lendab satelliit ESTCube-1 ööpäevas üle Eesti umbes 10 korda, millest päevasel ajal toimuvad 4-5. Missiooni operaatoritele on kasutatavad ainult päevasel ajal toimuvad ülelennud, mille keskmiseks pikkuseks on umbes 10-15 minutit. Ühe orbiidi periood on ligikaudu 90 minutit. Seega on ühe ööpäeva jooksul operaatorite jaoks kvaliteetset sideaega üsna vähe. MCS tarkvara võimaldab moodustada ülemaailmset maajaamade võrgustikku, mille abil oleks võimalik tunduvalt suurendada kvaliteetset sideaega satelliidiga. See võimaldaks saada suuremal hulgal andmepakette satelliidilt ja anda paremat tagasisidet missiooni käekäigu kohta. [2,3] Orbiidi liikumine Eesti suhtes on nähtav joonisel 2.



*Joonis 2. ESTCube-1 orbiidi liikumine Eesti suhtes. Valged jooned tähistavad ESTCube-1 orbiite ja kollane täpp on Tartus asetsev maajaam.*

## 2. Satelliidi jälgimine

Maajaama ja satelliidi vahelise sidekontakti toimumise aja teadasaamiseks on vaja välja arvutada satelliidi selline asukoht, kust maajaam on võimeline satelliidiga suhtlema. See tähendab seda, et maajaama ja satelliidi vahel on võimalik luua raadiosideseanss. Teadusharu, mis uurib kosmoses liikuvate kehade dünaamikat ja neile mõjuvaid jõude nimetatakse orbitaaldünaamikaks [5].

Satelliidiga seotud arvutuste jaoks on vajalik teada infot satelliidi kohta. Sellist infot hoitakse andmeformaadis TLE (two-line element set). TLE iseloomustab satelliiti ja kasutades vastavaid mudeleid on selle abil võimalik välja arvutada satelliidi positsioon kindlal ajahetkel. TLE-formaat on loodud NORAD (Põhja-Ameerika õhuruumi kaitsestaap) poolt ning seda kasutavad kõik satelliitide operaatorid.[6] Joonisel 2 on toodud TLE formaadi näide. TLE koosneb nimereast ja kahest andmerekast.

```
COMPASS-1
1 32787U 08021E 13110.14342084 .00001651 00000-0 19822-3 0 6007
2 32787 97.7577 170.2044 0016229 132.2140 261.6619 14.85233133269351
```

Joonis 2. Kuupsatelliidi COMPASS-1 orbiidi parameetrid antud TLE-formaadis.[7]

Tabel 1. COMPASS-1 TLE esimese rea elementide kirjeldused ja näidisparameetrid joonisel 2 kujutatud TLE põhjal.[6]

Väli	Veerg	Kirjeldus	Näidisparameetrid
1.1	01	Andmerekas number	1
1.2	03-07	Satelliidi number	32787
1.3	08	Klassifikatsioon	U
1.4	10-11	Rahvusvaheline tähis (stardi aasta kaks viimast numbrit)	08
1.5	12-14	Rahvusvaheline tähis (stardi number aastas)	021
1.6	15-17	Rahvusvaheline tähis (stardi osa)	E
1.7	19-20	Ajastu aasta (aastaarvu viimased kaks numbrit)	13
1.8	21-32	Ajastu (päev aastas ja murdosa päevast)	110.14342084
1.9	34-43	Esimest järku tuletis keskmisest liikumisest	.00001651
1.10	45-52	Teist järku tuletis keskmisest liikumisest	00000-0
1.11	54-61	BSTAR	19822-3
1.12	63	Number 0	0
1.13	65-69	Elemendi number	600
1.14	69	Kontrollsumma	7

Tabel 2. COMPASS-1 TLE teise rea elementide kirjeldused ja näidisparameetrid joonisel 2 kujutatud TLE põhjal.[6]

Väli	Veerg	Kirjeldus	Näidisparameetrid
2.1	01	Andmerea number	2
2.2	03-07	Satelliidi number	32787
2.3	09-16	Orbiidi kaldenurk (kraadides)	97.7577
2.4	18-25	Tõususõlme otsetõus (kraadides)	170.2044
2.5	27-33	Ekstsentrilisus	0016229
2.6	35-42	Perigee argument (kraadides)	132.2140
2.7	44-51	Keskmine kõrvalekalle (kraadides)	261.6619
2.8	53-63	Keskmine kiirus (orbiidid päevas)	14.85233133
2.9	64-68	Orbiidinumber	26935
2.10	69	Kontrollsumma	1

Orbiidi parameetrid muutuvad aja jooksul, seega väljastatud TLE andmed on aeguvad. Vastavalt vajadusele annab NORAD iga satelliidi kohta välja uued TLED ja sellega seoses tuleb TLEsid uuendada ka satelliitide jälgimise programmides. Antud bakalaureusetöös kirjeldatud satelliitide jälgimissüsteemides kasutatakse NORAD SGP4/SDP4 orbiidi mudeleid, mis on kõige täpsemad avalikuks kasutamiseks mõeldud mudelid [8]. Kuna satelliitide orbiidid pole täpselt ellipsi kujuga, siis kujutletavate orbiitide loomiseks ongi tarvis vastavaid mudeleid. SGP4/SDP4 mudelid on mõeldud kasutamiseks koos NORAD poolt väljastava TLEga [8].

Kuna orbitaaldünaamika teeki on juba varem loodud, siis antud töös kasutatakse satelliidi liikumisega seotud kalkulasioonide jaoks teeki OREKIT. OREKIT on avatud lähtekoodiga Java teek, mis on ühtlasi ka kõige laiemalt kasutatavam. Antud teeki on kasutatud Euroopa Kosmose Agentuuri poolt ATV (Automaatne transportmoodul) missiooni jaoks Rahvusvahelisse Kosmosejaama. OREKIT võimaldab kasutada erinevaid elemente nagu orbiidid, raamid ja asukohad ning pakub ka algoritme vastavate elementide käsitlemiseks. Samuti võimaldab OREKIT samm-sammult vajalikku kosmoseaparaati mööda tema orbiiti edasi liigutada, mille abil saab arvutada suvalisel ajahetkel kosmoseaparaadi asukoha kindla maapealse punkti suhtes. Kosmoseaparaadi asukoha teadmine suvalisel ajahetkel aitab arvutada kontakti aegasid ja kontakti iseloomustavaid parameetreid.[9]

## 2.1. Satelliidi jälgimise tarkvara

Varemalt on loodud mitmeid programme satelliitide jälgimiseks, kuid ükski neist pole keskendunud missioonide juhtimisele. MCSi suurimaks eeliseks on see, et tegemist on ühtse süsteemiga, mille abil saab satelliite jälgida, maajaamasid juhtida ja nende abil satelliitidele andmepakette edastada.



PREDICT on avatud lähtekoodiga reaal-ajas satelliitide jälgimise ja orbiidi ennustamise programm, mis on kirjutatud operatsioonisüsteemi Linux jaoks. PREDICTi suurimaks puuduseks on see, et tegemist on käsurea tarkvaraga, kuigi praeguseks on juba loodud ka kliendi rakendus, mis saavad andmeid läbi PREDICTi UDP liidese. Joonisel 3 on näidatud PREDICT kasutajaliidest. Programm on võimeline arvutama satelliidi ülelennu aega ja vaadata ka juba toimunud ülelendude aegu. Satelliidi jälgimise funktsioone on kaks – kas korraga jälgitakse ühte satelliiti või mitut, kuid jälgivaid maajaamu saab korraga olla vaid üks. PREDICT pole küll võimeline iseseisvalt maajaama, antenne ja raadioid juhtima, kuid selle jaoks on loodud PREDICTiga ühilduvad maajaama juhtimise utiliidid, näiteks PIC/TRACK Antenna Rotator Controller. PREDICT on oma võimaluste poolest küllaltki laialt kasutatav, seda on kasutanud oma missioonidel nii NASA (USA Riiklik Aeronautika- ja Kosmosevalitsus) kui ka ESA (Euroopa Kosmoseagentuur).[10]

Satellite	Az	El	LatN	LonW	Range		Satellite	Az	El	LatN	LonW	Range	
OSCAR-7	271	-45	-6	172	10911	D	RS-20	147	-58	-62	355	11669	N
OSCAR-11	275	-70	-28	210	12659	D	RS-22	193	-57	-73	117	11480	D
PACSAT	300	-56	4	203	11463	D	PCSAT	47	+4	53	45	2846	D
LUSAT	115	-33	-8	13	8151	D	HAMSAT	85	-57	-14	319	11463	N
ITAMSAT	335	-49	31	225	10715	D	OSCAR-57	327	-32	53	195	8217	D
OSCAR-27	133	+3	22	55	2998	D	OSCAR-58	249	-31	-3	136	7798	D
OSCAR-29	240	-75	-49	216	13643	D	NOAA-14	28	-15	69	1	5473	D
OSCAR-32	207	-65	-70	181	12449	D	NOAA-15	278	-73	-29	217	13027	D
OSCAR-50	342	-32	64	212	7804	D	NOAA-17	189	-67	-81	211	12608	D
OSCAR-51	209	-66	-68	182	12385	D	UARS	311	-28	48	165	6757	D
CUTE-1	131	-34	-20	23	8429	D	HUBBLE	28	-68	-1	273	12424	N
RS-15	336	-53	19	232	12710	D	ISS	285	-30	27	153	7002	D

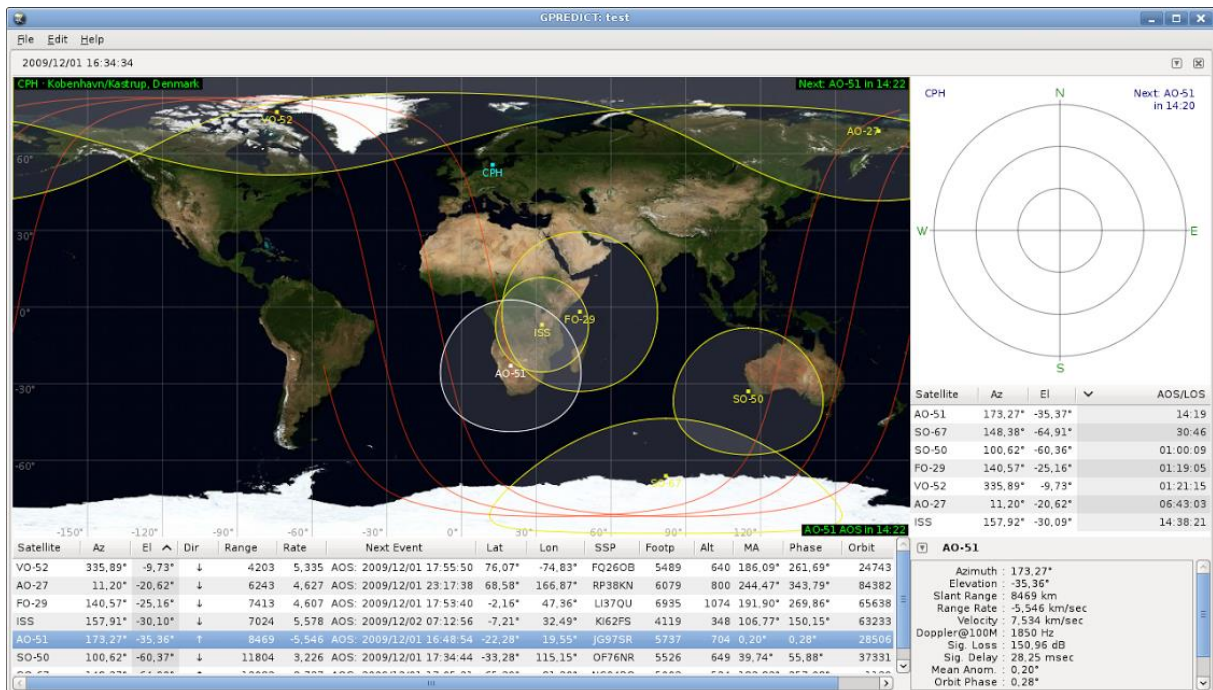
  

Upcoming Passes		
Sun	ISS	Moon
253.89 Az	ISS on Mon 15May06 20:02:43 UTC	41.48 Az
+46.63 El	CUTE-1 on Mon 15May06 20:06:40 UTC	-74.92 El
	RS-15 on Mon 15May06 20:18:46 UTC	

Joonis 3. Satelliidi jälgimise käsurea programm PREDICT, mille abil on võimalik jälgida mitut satelliiti korraga, kasutades ühte maajaama ja arvutada välja kontakti aegu mitmele satelliidile. Mitme satelliidi jälgimisel näidatakse igat satelliiti eraldi real ja iga satelliidi kohta iseloomustavat informatsiooni.[10]

GPredict on vabavaraline tarkvara satelliitide jälgimiseks ja saabuvate ülelendude ennustamiseks, see põhineb varem kirjeldatud programmil PREDICT. Erinevalt PREDICTist on antud

programmil asjalik graafiline liides, mis teeb programmi käsitlemise tunduvalt lihtsamaks ja selgemaks. Lisaks on GPredict saadaval nii Linuxi, Mac OS X kui ka Windowsi jaoks. GPredicti eripäraks on see, et kasutajal on võimalik jälgida samaaegselt mitut satelliiti ja mitut maajaama. Võrreldes PREDICTiga on GPredictile sisse kirjutatud ka maajaama raadio ja antenni pööraja kontrollid autonoomseks ühe satelliidi jälgimiseks. Tänu graafilisele liidesele on kasutajal võimalik näha satelliitide liikumist kaardil ja andmeid erinevates tabelites ning graafikutel. Joonisel 4 näeme GPredict graafilist kasutajaliidest, millel on selgesti näha satelliitide asukohad. Mugavuse poole pealt on kasutajal võimalik valida kui palju ja millist infot ta tahab ülelennu või satelliidi kohta näha, samuti on GPredicti vastavalt kasutaja tahtmisele võimalik modifitseerida. GPredicti eristab teistest satelliitide jälgimise programmidest see, et selle puhul on võimalik satelliidid grupeerida visualiseerimise moodulitesse ning igat moodulit on võimalik eraldi konfigureerida.[11]



Joonis 4. GPredict on satelliitide jälgimise programm, millega on võimalik jälgida ühte satelliiti ja arvutada mitme satelliidi kontaktide ajad. Satelliidi jälgimise vaade, kus kollasega on kaardil näidatud satelliidid ning sinisega on tähistatud maajaam. Lisaks on info satelliitide ja nendega toimuvate järgmiste kontaktide kohta. Ringid ümber satelliidi tähistavad raadioside ulatust ning punased jooned näitavad satelliitide orbiite.[11]

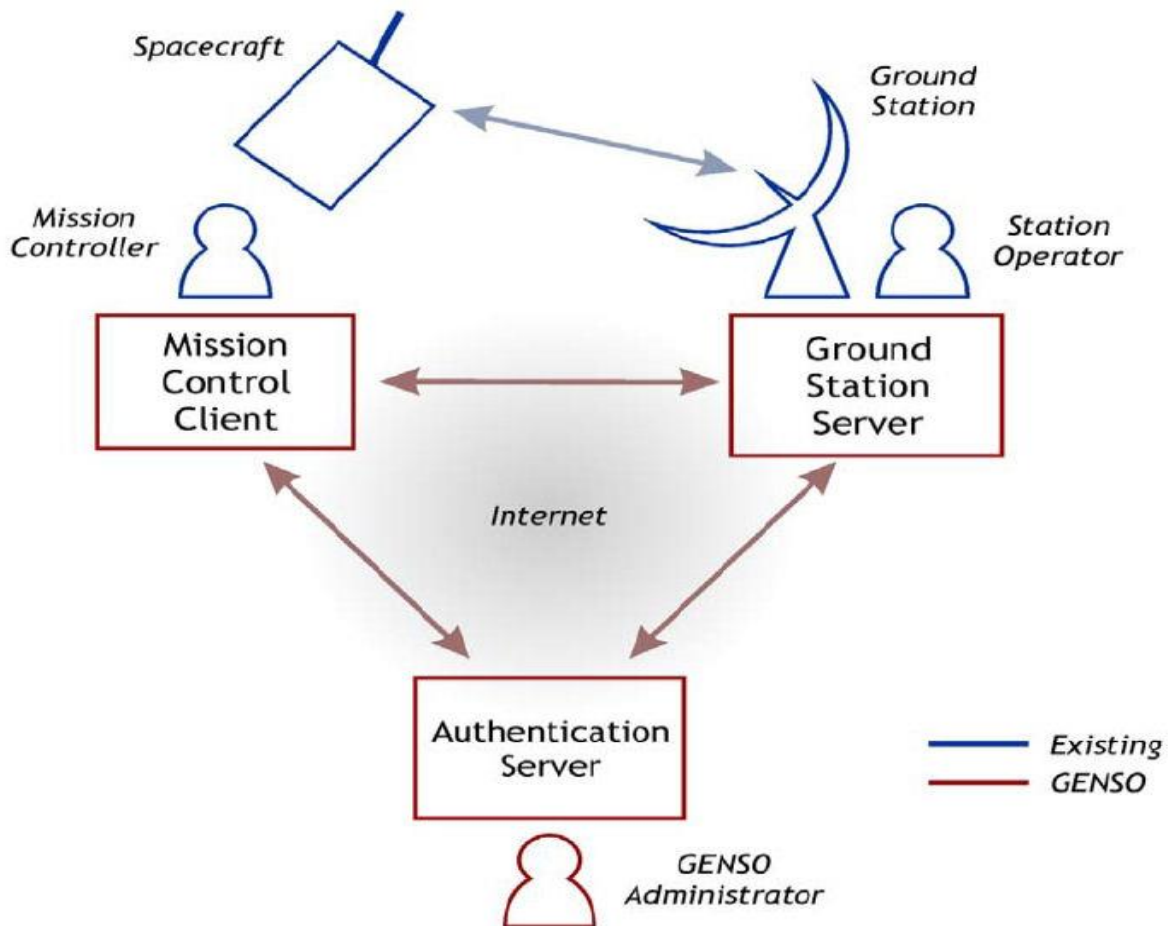
GENSO on satelliitide jälgimise tarkvara, mille peamiseks eesmärgiks on koondada suur hulk maajaamasid ühtesse võrgustikku ja mis koosneb mitmetest riist- kui ka tarkvaralistest komponentidest. Maajaamad omavahel ühendatud üle Interneti, seega saavad satelliidi operaatorid

kasutada maajaamasid erinevates maailma piirkondades. GENSO koosneb kolmest eraldi seisvast komponendist: AUS (ingl. k. Authentication Server), GSS (ingl. k. The groundstation server) ja MCC (ingl. k. The Mission Control Client). Komponentide omavahelised seosed on toodud joonisel 5.[12]

AUS ehk Autentimisserver on GENSO projekti keskne server, mille ülesandeks on võrgu autentimine ja krüpteerimine, satelliitide nimekirjade edasi jagamine, maajaamade monitoorimine ning võrgu statistika koostamine.[12,13]

GSS ehk maajaamaserver on tarkvara, mis tegeleb maajaama juhtimisega. Selle programmi ülesandeks on suunata maajaama antenne ja sättida paika raadiosagedused, et satelliidi ülenenu jooksul. GSS peab olema installeeritud vastavat maajaama juhtivasse masinasse. Antud tarkvara suhtleb AUSiga, mis annab GSSile satelliitide nimekirja.[12,13]

Kolmas osa GENSO süsteemist on MCC ehk missiooni juhtimisklient. Selle abil saab satelliidi operaator juhtida kindlat satelliiti. MCC eesmärgiks on edastada GSSile infot satelliidi kohta, mis sagedustel saab suhelda ja millal satelliit üle lendab. MCC saab infot olemasolevate maajaamade kohta läbi AUSi. Kui MCC on ühendatud AUSiga siis vastutab see ka selle eest, et uuendatud satelliidi andmed jõuaks AUSi ja sealt edasi GSSi.[12,13]

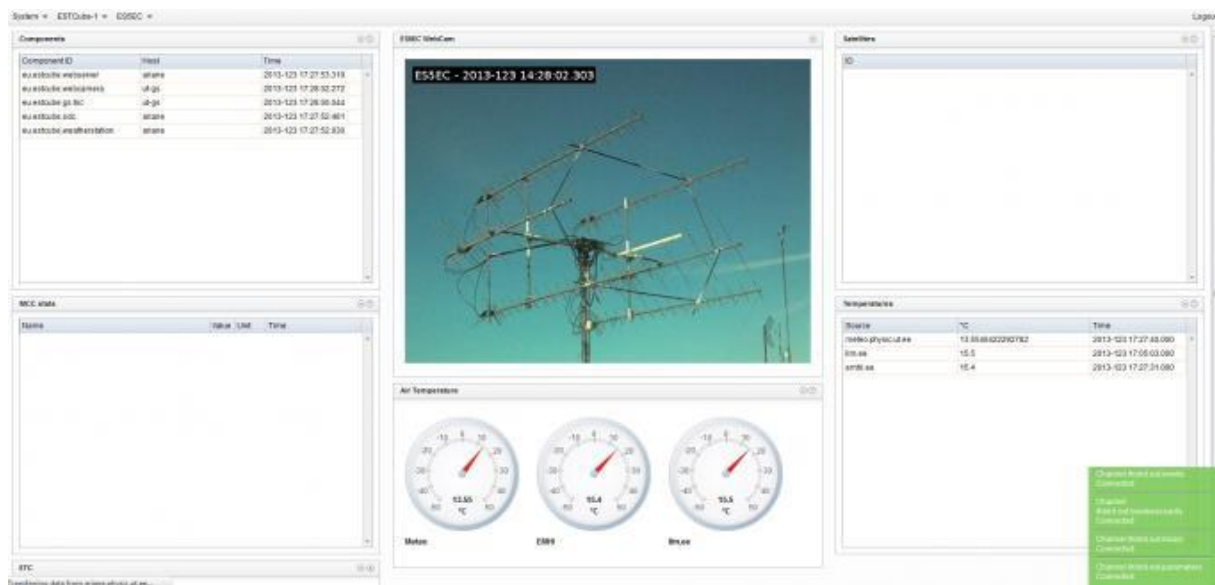


Joonis 5. GENSO on arenduses olev satelliite jälgimise tarkvara, mille kolm erinevat komponenti on ühendatud ühtseks süsteemiks üle Interneti [13]. GENSO võimaldaks erinevate maajaamadega jälgida korraga mitut satelliiti.

MCS on loodav missioonijuhtimistarkvara, mille peamiseks eesmärgiks on juhtida ESTCube-1 missiooni. Samamoodi nagu varem mainitud tarkvarade puhul on ka MCSga võimalik jälgida satelliidi teekonda ja saada infot satelliidi orbiidi kohta, kuid erinevuseks on see, et MCS on üks terviklik süsteem, millega on võimalik lisaks satelliidile juhtida ka maajaamasid ja edastada satelliitidele käsklusi. Lisaks on võimalik vaadata süsteemis olevaid andmeid reaalsajas kui ka näha logisid kuni missiooni alguseni välja. Eriliseks teeb MCSi see, et tegemist on vabavaralise veebirakendusega. Selle tõttu on võimalik kasutada MCSi ka teiste ESTCube-1 sarnaste projekti puhul. Olles veebpõhine on rakendusele võimalik ligi pääseda igast punktist kus on Internet, mille tõttu on võimalik luua ühtne maajaamade võrgustik. Oma veebipõhisuse tõttu on operaatorile võimalik MCSi kasutada erinevatel platvormidel ja eraldi installeerimata. Eesti missioonijuhtimistarkvara aluseks on tarkvararaamistik

Hummingbird. Joonisel 6 on kujutatud MCS süsteemi avaaken, millelt on näha maajaama antenni ja informatsiooni süsteemi erinevate komponentide kohta.

Hummingbird on vabavaraline tarkvararaamistik üksteisest eemal asuvate komponentide monitoorimiseks ja juhtimiseks. Hummingbird ühendab erinevad serveri komponendid ühtseks süsteemiks, mida on lihtsam hallata kui eraldi seisvaid üksuseid. Hummingbird on loodud juhtima kosmosega seotud komponentide juhtimiseks nagu on satelliidid, maajaamad, raketid jne. Kuid on kasutatav kõikide süsteemidega, kus on vaja koondada mitmed komponendid ühtseks süsteemiks.[14]



Joonis 6. MCS süsteem on Humminbirdil põhinev avatud lähtekoodiga missioonijuhtimistarkvara, mis võimaldab luua maajaama võrgustikke üle Interneti. Avaaken, millelt saab kasutaja näha informatsiooni kogu süsteemi kohta ja valida suhtluseks sobiva maajaama ning satelliidi. Käesoleva bakalaureusetöö eesmärgiks on luua kontakti arvutamise moodul MCS süsteemi jaoks.

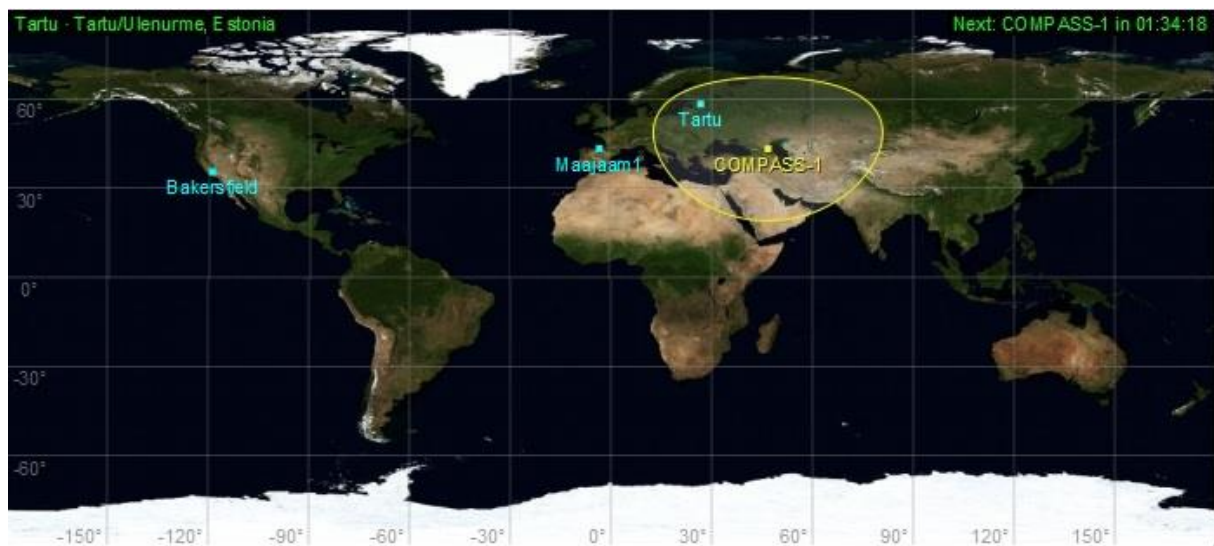
## 2.2. Erinevad ülelennu strateegiad

Lähtuvalt satelliidi operaatori vajadusest või soovist on võimalik jälgida üle lendavaid satelliite erinevalt. Kõige lihtsam variant on see, kus üks maajaam jälgib ühte kindlat satelliiti. Sellisel puhul seisneb kogu strateegia selles, et kui jõuab kätte ülelennu aeg on maajaam valmis satelliidiga sidet pidama ja ei lase teistest ülelendavatest satelliitidest end segada. Tuleb ainult jälgida, et soovitatav maajaam oleks võimeline kuulama vastavat satelliiti ja maajaam oleks vajalikul ajahetkel saadaval. Joonistel 7 kuni 10 on kaardil näha erinevad ülelennu strateegiad.



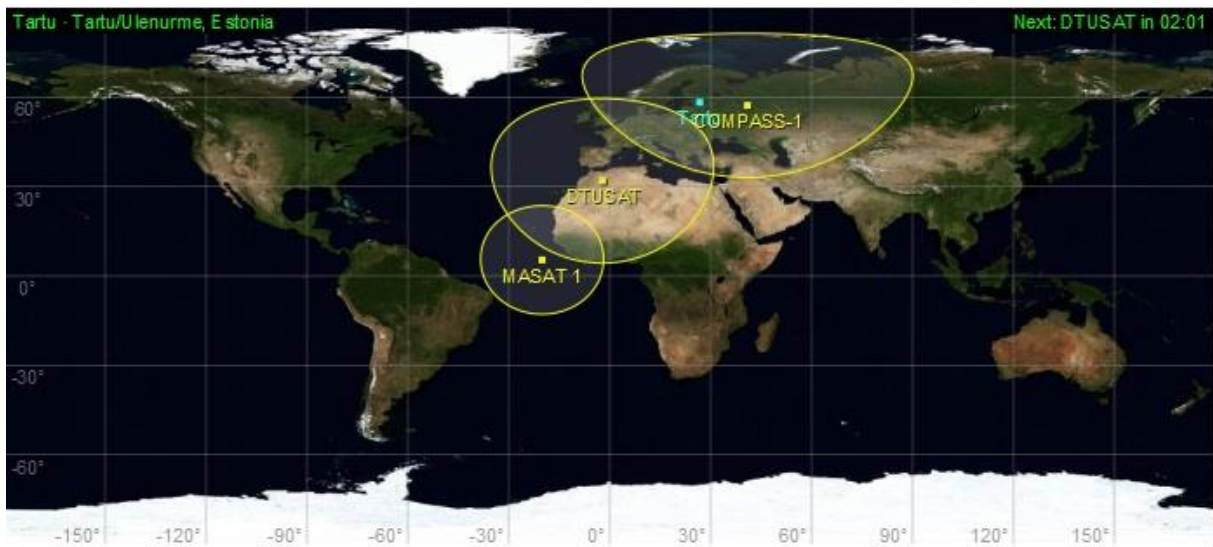
*Joonis 7. Ülelennu strateegia, kus tegemist on ühe kindla maajaama ja satelliidiga. Kaardil on maajaam kujutatud sinise täpiga ja satelliit kollase täpiga. Kollane ring ümber satelliidi tähistab raadioside ulatust.*

Üks satelliit ja mitu maajaama: mitu maajaama kuulavad ühte satelliiti, tuleb kindlaks teha kuidas täpselt see vajalik on, kas korraga kuulavad kõik ühte, järjestikku ilma vaheta või sobilik mingi kindla aja tagant. Tuleb arvestada ka sellega, et soovitud maajaam oleks vaba, ega oleks mõne teise satelliidi jaoks broneeritud. Sobitada omavahel üksteist mõistvad satelliidid ja maajaamad. Mingi kindla aja tagant arvutamisel tuleb silmas pidada, et oleksid maajaamad sobivate vahedega. Lisaks on võimalus ainult päevaseid ülelende kuulata, mille eeliseks on see, et satelliidil on rohkem energiat andmeedastuseks ja nähtavus on ka parem. Antud strateegia on ka kõige laialdasemalt kasutatav, sest üldjuhul tegeleb operaator korraga ühe satelliidiga.



Joonis 8. Ülelennu strateegia, kus operaatoril on võimalik kasutada mitut maajaama ühe satelliidi jälgimiseks. Kaardil on maajaam kujutatud sinise täpiga ja satelliiti kollase täpiga. Kollane ring ümber satelliidi tähistab raadioside ulatust.

Mitu satelliiti ja üks maajaam: Veel keerulisem, tuleb määratleda milline satelliit on kõige prioriteetsem, ehk siis millist kuulatakse eelisjärjekorras, et ei tekiks olukorda kus mitme satelliidi ülelennu puhul ei osata valida millist kuulata. Samuti tuleb arvestada kui väikse vahega suudetakse satelliite järjest jälgida. Sõltub maajaama võimekusest. Näiteks kui kiiresti on maajaam võimeline keerama antennidega täispööret tagasi olukorrast kust enam edasi ei saa pöörata. Täispööre tuleb keerata sellepärast, et saaks jätkata sealt kust pooleli jäi. Olenevalt maajaama võimekusest on võimalik kuulata korraga ka mitut satelliiti.



Joonis 9. Ülelennu strateegia, kus ühte maajaama kasutatakse mitme satelliidi kuulamiseks. Kaardil on maajaam kujutatud sinise täpiga ja satelliit kollase täpiga. Kollane ring ümber satelliidi tähistab raadioside ulatust.

Mitu satelliiti ja mitu maajaama: kõige pealt tuleks sobitada omavahel sobivad maajaamad ja satelliidid. See tähendab, et maajaamade ja satelliitide vastuvõtu ja saatmise sagedused oleksid sobivad. Sobitamised saab teha vastavalt vajadusele, nagu varasemate puhul kas tahetakse kvaliteetset ülelendu, mingit kindlat ülelendude arvu või hoopis pikka ülelendu. Lisaks tuleb arvesse võtta, et soovitud maajaamad poleks juba varem broneeritud. Kehtivad ka kõik varasemad nõuded.



Joonis 10. Ülelennu strateegia, kus operaator kasutab mitut maajaama mitme satelliidiga suhtlemiseks. Kaardil on maajaam kujutatud sinise täpiga ja satelliit kollase täpiga. Kollane ring ümber satelliidi tähistab raadioside ulatust.



### 2.3. Satelliitide ülelendude optimeerimine

Ümber Maa tiirleb väga palju satelliite ja Maa peal asub mitmeid maajaamu, ehk siis ressursse mille vahel on võimalik sidet pidada on palju ning pidevalt toimub mitmeid ülelende.

Ülelendude optimeerimist kasutatakse, et missiooni operatsioone planeerides saaksid sobivad ressursid optimaalselt kasutatud. Ülesanneteks on satelliidilt andmete saamine, andmete edastamine ja satelliidi juhtimine ning ressursideks on satelliidid ja maajaamad. Lähtuvalt ülesande iseloomust tuleb maajaamade kasutamine nii optimeerida, et soovitud ülesannet oleks võimalik täita. Näiteks kui tegemist on andmete allalaadimisega siis ei pruugi ühest ülelennust jätkuda, et kõik andmed Maale jõuaks. Sellisel juhul tuleb kasutada mitut maajaama, mis järjest kuulaks ühte ja sama satelliiti. Niimoodi maajaamasid kasutades muutub sideseansi aeg pikemaks ja on võimalik edastada suuremaid andmemahte. Samuti tuleb arvestada ka sellega, et planeeritavad ülesanded ei segaks üksteist ega oleks vales järjekorras. Mõned ülesanded tuleb läbi viia kindlas järjekorras, näiteks enne andmete alla laadimist tuleb sobivad andmed koguda.[15]

Optimeerimist kasutatakse, et satelliitide ülelennud oleks võimalikult optimaalselt ära kasutatud. Sõltumata ülelennu strateegiast, kus on tegemist rohkem kui ühe satelliidi või maajaamaga tuleb eelkõige arvestada omavahelist sobilikkust. Kui maajaam ja satelliit oma vahel sobivad, siis tuleks arvestada kasutaja soovi millist ülelendu on tarvis. Optimeeritavaid ülelende on neli erinevat:

- võimalikult pikk ülelend,
- võimalikult kvaliteetne ülelend,
- ülelennud kindla ajavahemiku tagant,
- ülelennud ainult päikese käes.

Need ülelendude tüübid on sõltumatud ülelennu strateegiast. Optimeerimisel tuleb arvestada, et erinevate kontaktide ajad ei kattuks. Ülelendude kattumise korral tuleb vaadata, kas on võimalik kasutada alternatiivseid lähedal asuvaid maajaamasid. Sõltuvalt soovitatavast ülelennu tüübist toimub ka erinev optimeerimine.

## **3. Metoodika**

### **3.1. Testimisplaan**

Loodud kontaktide arvutamise tarkvara tuleb kindlasti ka testida. Kuna ESTCube-1 pole selle bakalaureusetöö aluseks oleva tarkvara testimise ajal veel oma orbiidile lennanud siis ei saa füüsiliselt katsetada, kas antud tarkvara töötab meie satelliidiga õigesti. Selle tõttu kasutame testimiseks teisi satelliite nagu Compass-1 ja ISS. Testimise käigus võrreldakse omavahel kahe erineva programmi poolt saadud tulemusi. Esimene programm on antud bakalaureusetöö raames loodav tarkvaramoodul ja teine on ennast juba varasemalt tõestanud satelliitide jälgimise programm GPredict.

Testimise käigus tuleb katsetada, et antud bakalaureusetöö raames loodud tarkvara töötaks ilma vigadeta ka erijuhtudel näiteks kui tegemist on geostatsionaarse satelliidiga või maajaam asub poolusel. Testijuhud on testitud kasutades ühiktestide tehnoloogiat. Lisaks nendele testijuhtudele on loodud ka ühiktestid, mis kindlustavad programmi koodi õige käitumise ja annavad märku sellest, et kui kasutatavat teeki OREKIT on muudetud ja sellega seoses on midagi katki läinud. Ühiktestide kirjutamiseks kasutatakse Java JUnit raamistikku. Jälgitakse ka, et vähemalt 80% loodud koodist saaks ühiktestidega katsetatud, selle jaoks kasutatakse tööriista EclEmma, mis on Java koodi kattuvuse tööriist, ja Clover raporteid.

Kontakti arvutamise tarkvara arvutab lisaks kontakti aegadele ka sinna juurde kuuluvad parameetrid, mille abil saab hinnata toimuva ülelennu kvaliteeti. Samuti tuleb testidega katta ka erinevate parameetrite arvutamine ja võrrelda neid GPredictiga saadud tulemustega.

### **3.2. Üks satelliit ja mitu maajaama juhtum**

Optimeerimise juht, kus tegemist on ühe satelliidi ja mitme maajaamaga on kõige laialdasemalt kasutatav. Seda sellepärast, et üldjuhul soovib kasutaja saada võimalikult palju ja täpset infot ühe kindla satelliidi kohta, kasutades selleks mitut maajaama. See kuidas maajaamad satelliidi kuulavad sõltub juba kindlast ülesandest. Antud juhul on optimeerimise ülesandeks leida kõige sobivam maajaamade kasutamise võimalus vastavalt kasutaja vajadustele. Optimeerimise suurimaks väljakutseks on eristada erinevad maajaamade kasutamise olukorrad ning selle järel satelliit siduda vastava maajaamaga.[16]

Satelliidiga suhtlemisel on kaks peamist ülesannet millega tuleb tegeleda, esiteks on satelliidile andmete või käskude edastamine ja teiseks on satelliidilt andmete vastu saamine.

Lähtuvalt ülesande iseloomust tuleb hinnata umbkaudset ajakulu antud ülesande lahendamiseks. Optimeerimise puhul tuleb arvestada ajakulu, kui see on piisavalt lühike on võimalik kasutada ainult ühte maajaama, sest ülesanne saab täidetud ainsa ülelennu korral. Sellises olukorras, aga tuleb vaadata kas satelliit asub päikese käes või mitte ja kas on veel segavaid faktoreid, mis võivad raskendada andmete edastust satelliidile. Antud olukorras on optimeerimise ülesandeks välja valida parim võimalik maajaama ja satelliidi paar ning näidata seda ka kasutajale. Optimeerimine saab alguse kõigi võimalike kontaktide välja arvutamisega.[16]

Olukorras, kus on tarvis kasutada rohkem kui ühte maajaama on optimeerimine oluliselt keerukam. Optimeerimise tarkvara peaks suutma aru saada, kas maajaamad kuuluvad satelliiti järjest või korraga. Kui mitu järjest siis tuleb kindlaks teha kui suures ulatuses on ülelennud kattuvad ja võib-olla oleks tarvilikum kasutada teisi maajaamu. Samuti peab arvestama ka satelliidi energiavarudega vastavate ülesannete täitmiseks. Alati on paremad ülelennud, kus satelliit asetseb päikese käes.

Lihtsamad on olukorrad kus satelliidiga suhtleb samaaegselt ainult üks maajaam, näiteks olukord kus on vajalik kindel ülelendude arv ööpäevas või ainult päikese käes toimuvad ülelennud. Sellistel puhkudel tuleb välja selgitada kontaktid, mis vastavad antud nõudmistele ja valida välja sobivaim lahendus kasutajale.

Optimeerimise tarkvara oleks otseselt seotud kontakti arvutamise mooduliga, sest mooduli abiga arvutatakse kõik võimalikud kontaktid. Tarkvara sisendiks oleks nimekiri maajaamadest ja satelliidi metaandmed. Nende andmete põhjal sorteeritakse sobivad maajaama ja satelliidi paardid ning arvutatakse võimalikud kontaktid kindla ajavahemiku jooksul. Optimeerimise programm valib arvutatud kontaktidest välja sobivad lähtuvalt täidetava ülesande iseloomust.

### **3.3. Kontakti arvutamine**

Antud bakalaureusetöö praktiliseks osaks oli luua maajaama ja satelliidi vahelise kontakti arvutamise tarkvara, mis on ühtlasi osa loodavast missiooni juhtimise tarkvarast MCS. Kontakti arvutamise protsess saab alguse kasutajaliidesest, kust määratakse millise perioodi jooksul ning missuguste maajaamade ja satelliitide vahelisi kontakte on vaja arvutada. Kontakti aja ning sinna juurde kuuluvate parameetrite arvutamiseks kasutatakse Java teeki OREKIT, mis on vabavaraline teek kosmose dünaamika arvutusteks. Lisaks tuleb kontakti arvutamise protsess käivitada ka juhul kui TLE on uuenenud või maajaama metaandmed on muutunud, sest sel juhul võivad varem arvutatud kontakti ajad olla valed ja need tuleb uuesti

arvutada. Maajaama metaandmeteks on maajaama asukoht, satelliidi nägemise nurk, nimi, üles- ja allalink ning signaali kadu. Maajaama metaandmed on olulised arvutamaks kõiki vajalikke parameetreid. Maajaama asukoht on määratud pikkus ja laiuskraadidega ning kõrgusega merepinnast. Satelliidi nägemise nurk on elevatsiooni nurk, millest alates on maajaamal võimalik sidet satelliidiga sidet pidada. Elevatsioon tähistab nurka horisondi ja satelliidi vahel. Lisaks, et eristada omavahel maajaamasid on igal maajaamal olemas nimi. Samuti iseloomustavad maajaamu üles- ja allalinki raadiosagedused, millega on maajaam suuteline sidet pidama ja signaalikadu, mis näitab maajaama saatja signaali sumbuvus.

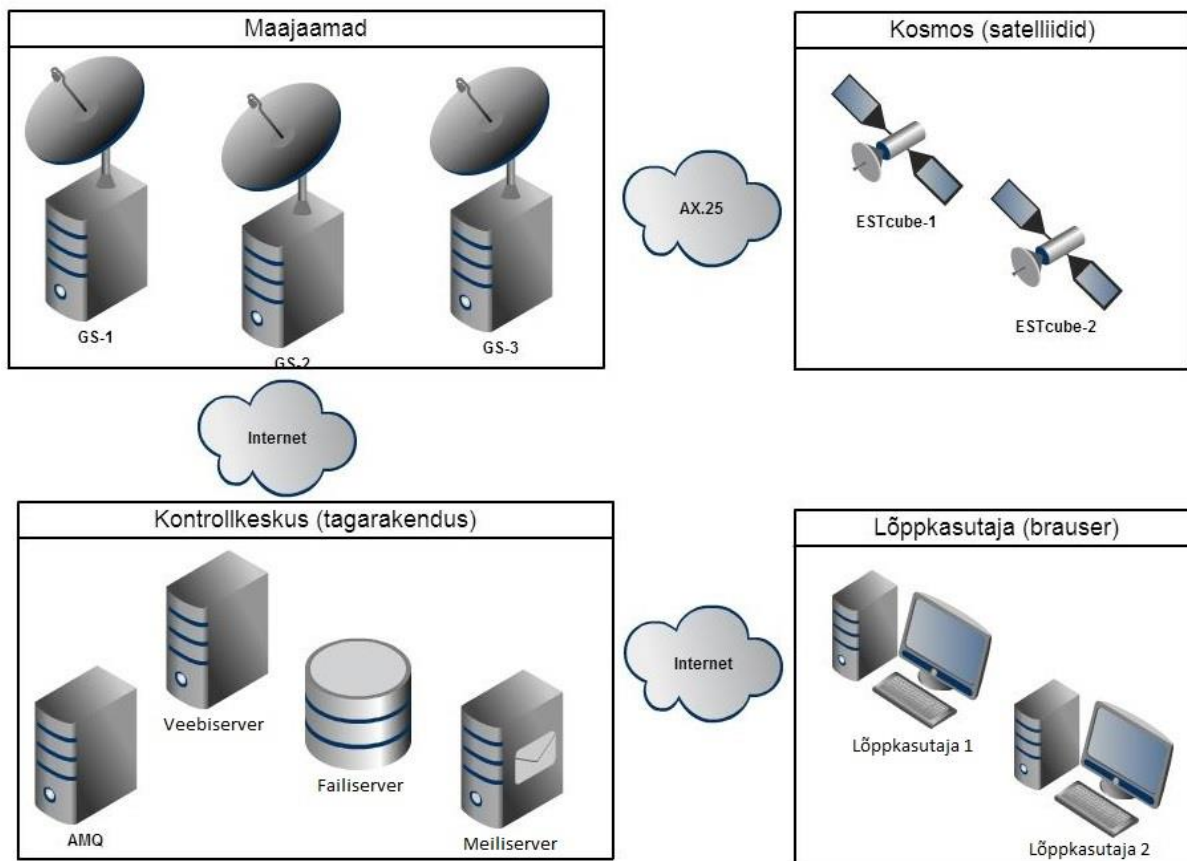
Kuna satelliidid tiirlevad ümber maa, siis arvutustes on vaja silmas pidada ka mitmeid maaga seotud konstante nagu näiteks Maa gravitatsiooniline konstant ja ekvatoriaalne raadius, lisaks on tarvilik teada ka valguskiirust. Antud rakenduses on kasutusel WGS84 mudelil põhinevad konstandid. Nende konstantide abil määratakse maajaama õige asukoht maapinnal ja pannakse paika algne orbiit.

OREKIT kasutab oma arvutuste tegemiseks NORAD SGP4/SDP4 mudeleid, nende abil luuakse vastava satelliidi TLE-ga koos eeldatav orbiit, millel saab satelliiti ajas edasi liigutada. Arvutuste sisendiks on TLE, maajaama ja satelliidi metaandmed ning ajavahemik mil soovitakse kontakte arvutada. Kõik arvutatud andmed salvestatakse kindlasse Java objekti, iga kontakt eraldi ning väljundina saab nimekirja vastavatest Java objektidest. Saadud tulemused saadetakse edasi komponendile, mis juhib maajaama antennipöörajaid ja raadiosagedusi. Samuti saavad kontakti ajad nähtavaks ka kasutajale läbi kasutajaliidese. Kontakti aja määramiseks kasutatakse loodud eeldatavat orbiiti ja nihutatakse satelliiti mööda seda edasi punktini, mil maajaam on suuteline vastavat satelliiti nägema. Selles punktis märgitakse aeg ning täpne satelliidi positsioon. Samal viisil arvutatakse ka kontakti aja lõpp, nihutatakse satelliiti punkti kust maajaam enam ei ole võimeline satelliiti nägema.

### **3.4. MCS süsteem**

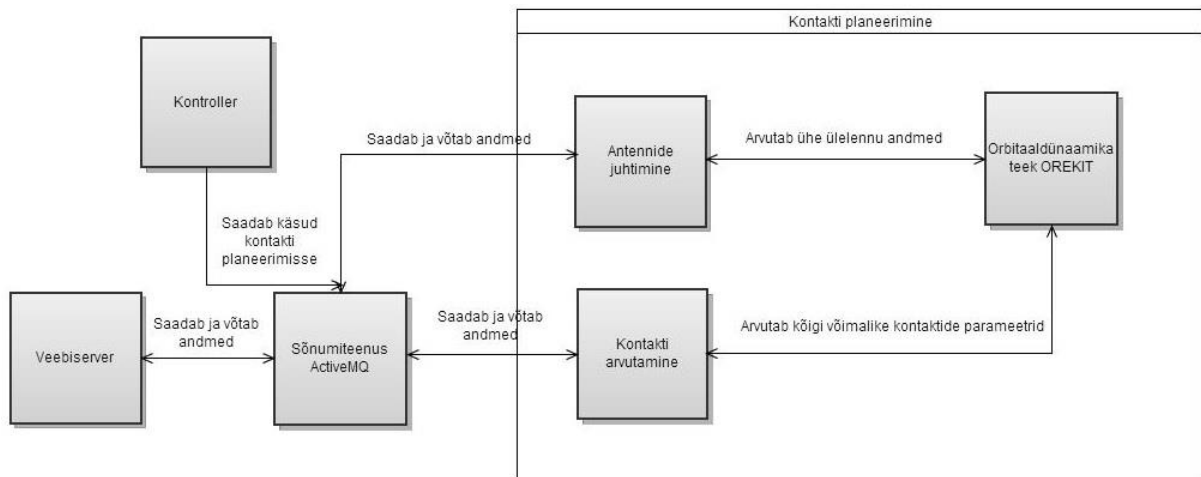
MCSi puhul on tegemist veebirakendusega, mis on mõeldud satelliitide jälgimiseks ja nendega suhtlemiseks. Kasutajaliidese pool on arendatud kasutades HTML5 ja Javascripti Dojo raamistikku. MCS „selgrooks“ on Hummingbird, mis on mõeldud eraldi asetsevate ressurside ehitamiseks, monitoorimiseks ja kontrollimiseks. Hummingbirdi eesmärgiks on siduda erinevad komponendid ühtseks süsteemiks[14]. MCS süsteem on loodud kasutades Java programmeerimiskeelt. Java keele valimisel on mitmeid põhjuseid:

- Java on mitmeplatvormiline keel, see tähendab, et Java töötab erinevatel operatsioonsüsteemidel nagu näiteks Windows, Linux, Mac.
- Lihtne ja õpetatakse Tartu Ülikoolis. Kuna MCS luuakse tudengite poolt siis on kasutatavate tehnoloogiate lihtne õpitavus väga oluline.
- Laialt levinud ja suur kasutajaskond. Java keelele on loodud väga palju erinevaid raamistikke ja teeke, mille kasutamine muudab programmeerimise oluliselt lihtsamaks.



*Joonis 11. MCS süsteemi arhitektuur, mis näitab kuidas erinevad MCS komponendid omavahel suhtlevad alustades satelliitidest kuni lõppkasutajani. MCS on jaotatud neljaks osaks: kosmos ehk satelliidid, maajaamad, kontrollkeskus, mis hõlmab endas veebiserverit, failiserverit ja sõnumiteenust ActiveMQ, ja lõppkasutaja. Loodav kontakti arvutamise tarkvaramoodul on osa tagarakendusest.*

MCS süsteemi erinevad komponendid on omavahel seotud sõnumiteenusega ActiveMQ. ActiveMQ on Java põhine sõnumiedastus platvorm, mille kaudu saadetakse kindlat kanalit pidi serverile sõnumeid ja samu kanaleid saab järgmine süsteemi osa kuulata.[17] Joonisel 12 on näha milliste komponentidega on kontakti arvutamise moodul seotud.



Joonis 12. Kontakti arvutamise moodul on seotud antennide juhtimise mooduli ja veebiserveriga läbi ActiveMQ sõnumiteenuse. Kontakti arvutamise moodul ja antennide juhtimise moodul koos OREKIT kosmosearvutuste teegiga moodustavad kontakti planeerimise komponendi. OREKIT on mõlema mooduli toetavaks osaks, kasutatakse vajalike arvutuste tegemiseks. Kontroller komponendi eesmärgiks on juhtida, millist kontakti planeerimise osa on parasjagu vaja kasutada

### 3.5. Satelliidi ülelendu iseloomustavad parameetrid

Lisaks kontakti aegade arvutamisele on antud tarkvaramoodulil teinegi väga oluline ülesanne, nimelt arvutada parameetrid satelliidi ülelennu iseloomustamiseks ja kvaliteedi hindamiseks. Ülelendu iseloomustavad parameetrid on vajalikud antennide juhtimise mooduli jaoks. Arvutatud näitajate abil saab satelliitide jaama valmis seada satelliidi ülelennuks, seadistada raadiod õigetele sagedustele ning keerata paika antennid. Maajaamale olulisteks parameetriteks on kontakti alguse ja lõpu asimuudid. Asimuut on nurk põhjasuuna ja satelliidi vahel. Lisaks on oluline satelliidi maksimaalne elevatsioon, mis tähendab nurka horisondi ja satelliidi vahel. Elevatsiooni ja asimuutidega määratakse antennipööraja juhtimise moodulis pööraja käitumine, et satelliiti saaks jälgida pidevalt ilma katkestusteta. Maajaama raadiosageduste täpseks sättimiseks arvutatakse Doppleri nihe, mis näitab kui palju muutub raadiosignaali sagedus seoses satelliidi liikumisega.

Satelliidi ülelennu kvaliteedi hindamiseks on mitmeid parameetreid. Olulisim neist on see kas satelliit asub ülelennu hetkel päikese käes või mitte. Kuna väikestel satelliitidel on piiratud elektrienergia, siis Päikese käes olles on nende poolt võimalik kasutatav energia hulk suurem ja sellega saab korda saata energiakulukamaid protsesse. Teada saamiseks, kas satelliit asub päikese käes või mitte arvutatakse välja nurk, kus satelliit jääb Päikese suhtes Maa varju.

Järgmine oluline parameeter satelliidi ülelennu iseloomustamiseks on raadiosignaali sumbu-  
vus. Raadiosignaali sumbuvus arvutatakse eraldi üles- ja allalangi sageduste kohta kontakti  
alguses ja lõpus ning leitakse minimaalne ja maksimaalne signaalisumbuvus. Signaali sumbu-  
vuse arvutamiseks kasutatakse valemit, mis on toodud valemil 1.[18]

$$L_0 = 32.4 + 20 \log(f_c) + 20 \log(d)$$

*Valem 1. Signaali sumbuvuse üldine valem.[18]*

Signaali sumbuvuse arvutamiseks on tarvis teada maajaama ja satelliidi vahelist kaugust. Li-  
saks arvutatakse signaali viivitus ja orbiidinumber, millest viimane näitab mitmendat tiiru  
teeb satelliit ümber Maa.

## 4. Tulemused

### 4.1. Kontakti arvutamise moodul

Kontakti arvutamise protsess saab alguse kasutajaliidesest kui defineeritakse vastav maajaam ja satelliit, mille vahelisi kontakte soovitakse arvutada. Protsessi sisenditeks on satelliidi ja maajaama metaandmed ning soovitatav ajavahemik, mille jaoks kontakte arvutatakse. Satelliidi metaandmeteks on satelliidi TLE, satelliidi nimi ja satelliidi signaalikadu. Maajaama metaandmed on maajaama asukoht, satelliidi nägemise nurk, üles- ja allalangi sagedused ning signaalikadu.

Kontakti arvutamise jaoks kasutatakse OREKIT poolt loodud *ElevationDetector* klassi, mille eesmärgiks on leida punkt, kus satelliit muutub maajaamale nähtavaks ja punkt, kus satelliit kaob maajaama vaateväljast. Kontakti alguse ja lõpu leidmiseks liigutab programm satelliiti mööda oma orbiiti punktini, kus satelliidi elevatsioon maajaama suhtes on võrdne ette antud nurgaga. Selles olukorras registreeritakse *ElevationDetector* abil toimunud sündmus, mis on ühtlasi ka kontakti alguse ajaks. Satelliiti liigutatakse edasi kuni elevatsioon on uuesti võrdne maajaama metaandmetest saadud satelliidi nägemise nurgaga ja see on kontakti lõpu aeg. Nii alguses kui lõpus salvestatakse satelliiti iseloomustav muutuja *SpacecraftState*, mis omab endas informatsiooni satelliidi kohta kindlal ajahetkel. Antud muutuja abil arvutatakse vajalikud ülelendu iseloomustavad parameetrid. Kontakti alguse ja lõpu aeg ning sinna juurde arvutatud parameetrid hoitakse eraldi Java objektis *ContactEventData*. Kasutaja poolt etteantud ajavahemikul võib toimuda mitu kontakti, iga kontakti kohta on oma *ContactEventData* objekt ja need kirjutatakse Java List objekti. Listi eesmärk on hoida endas kõiki etteantud ajavahemikul toimunud kontakte ühtse objektina, et oleks lihtsam saadud infot edasi saata.

Kontakte iseloomustavate parameetrite arvutamiseks kasutatakse vastavaid OREKITi meetodeid. Asimuudid arvutatakse kasutades OREKIT vastavat *getAzimuth()* meetodit, mis kasutades maajaama asukohta, satelliidi positsiooni antud ajahetkel ning antud aega arvutab vajaliku nurga radiaanides. Samamoodi kasutatakse elevatsiooni arvutamiseks vastavat OREKIT meetodit *getElevation()*. Kontakti maksimaalse elevatsiooni arvutamiseks võetakse kontakti alguse- ja lõpuaeg ning satelliidi positsioon alguses. Satelliiti hakatakse liigutama mööda orbiiti edasi 30 sekundiliste vahedega ja iga vahe tagant arvutatakse uuesti elevatsioon ning nendest tagastatakse suurim. Järgnevateks parameetriteks on Doppleri nihe. Doppleri nihe arvutatakse kontakti alguses ja lõpus ning eraldi üles- ja allalangi sagedustele. Arvutami-



seks kasutatakse sobivat OREKIT meetodit, mis arvutab satelliidi positsiooni ja kiiruse vektorite skalaarkorrutis, mille abil arvutatakse standard valemiga Doppleri nihe.

Signaali sumbuvuse arvutamiseks kasutatakse satelliidi kaugust maajaamast, signaali sagedust ning liidetakse maajaama ja satelliidi poolsed sumbuvused. Signaali sumbuvuse üldist valemit on muudetud, et see vastaks MCSi nõuetele. Signaali sumbuvus arvutatakse kontakti alguses ja lõpus eraldi üles- ja allalingi sagedustele. Samuti leitakse signaali sumbuvus seniidis ja horisondil. Signaali sumbuvuse arvutamiseks on vajalik teada maajaama ja satelliidi vahelist kaugust. Selle arvutamiseks kasutatakse OREKIT meetodit *getRange()*, mis teades maajaama asukohta ning satelliidi positsiooni kindlal ajahetkel, leiab nende vahelise kauguse. Antud moodul arvutab maksimaalse ja minimaalse kauguse ülelennu vältel ning leiab selle ka kontakti alguses ja lõpus. Lisaks arvutatakse ka signaali viivitus, jagades maajaama ja satelliidi vahelise kauguse valguse kiirusega. Viimane arvutatav parameeter on orbiidinumber, mis näitab mitmendat tiiru satelliit ümber Maa teeb. Orbiidinumber arvutatakse satelliidi TLE põhjal arvestades satelliidi orbiidil oleku aega ja liikumise kiirust. TLEs on defineeritud ligikaudselt mitu tiiru teeb ööpäevaga satelliit maale ning sellest lähtuvalt arvutatakse orbiidinumber.

Kuigi TLE muutub ajas ja vajab uuendamist, siis antud kontakti arvutamise moodulil sellega probleeme ei teki, sest TLE antakse moodulile ette koos satelliidi metaandmetega ja TLE korrektsus kontrollitakse juba eelnevalt komponendis, mis haldab kontaktide planeerimist. Kui TLE on aegunud ja sellega seoses on vaja uusi kontakti aegasid, siis kutsutakse kontakti arvutamise meetod välja uuendatud TLEga.

## 4.2. Testimine

Testimise käigus sai läbi proovitud erinevad testjuhud:

- maajaam asub poolusel,
- satelliit on geostatsionaarne,
  - satelliit asub maajaama ulatuses,
  - satelliit ei asu maajaama ulatuses,
- polaarorbiidil asetsev satelliit,
- mitte polaarorbiidil asetsev satelliit (ISS).

Lisaks on kood kaetud ühiktestidega, mille eesmärgiks on aru saada OREKIT poolt tehtud muudatustest, mis mõjutavad kasutatavaid funktsioone.

Esimene testitav olukord on see, kus maajaam asub poolusel. See testijuht on selleks, et katsetada orbiidinumbrite arvutamise korrektsust. Polaarorbiidil lendab satelliit igal orbiidil üle pooluse, seega peavad olema orbiidinumbrit järjestikku ilma igasuguse vaheta. See lähenemine tagab selle, et orbiidi numbrite arvutamise meetod toimib korrektselt. Ühe orbiidi läbimiseks kulub satelliidil ligikaudu 90 minutit. Tabelis 3 on näha, et ülelendude vahed on umbes 90 minutit ja igal ülelendul suureneb orbiidi number ühe võrra. Tabelis on toodud orbiidi numbrid, mis on saadud kasutades GPredict tarkvara ning kontakti arvutamise mooduli poolt arvutatud orbiidi numbrid.

*Tabel 3. Orbiidi numbri arvutamise testi tulemused. Testimise tulemused on saadud kasutades välja mõeldud maajaama koordinaatidega 90° põhjalaius ja 0° idapikkus, mille kõrgus merepinnast on 0 meetrit. Satelliidiks oli Compass-1, mille TLE oli võetud 15.04.2013. Kontaktide arvutamise ajavahemikuks oli 15.04.2013 10:20:00 kuni 16.04.2013 02:00:00 UTC (ingl. k Coordinated Universal Time) järgi.*

Kontakti alguse aeg	Kontakti lõpu aeg	Arvutatud orbiidi number	GPredict orbiidi number
2013-04-15T10:24:27	2013-04-15T10:36:49	26864	26864
2013-04-15T12:01:19	2013-04-15T12:13:40	26865	26865
2013-04-15T13:38:10	2013-04-15T13:50:32	26866	26866
2013-04-15T15:15:02	2013-04-15T15:27:23	26867	26867
2013-04-15T16:51:53	2013-04-15T17:04:15	26868	26868
2013-04-15T18:28:44	2013-04-15T18:41:07	26869	26869
2013-04-15T20:05:36	2013-04-15T20:17:58	26870	26870

Järgmine testijuht on see kui tegemist on geostatsionaarse satelliidiga. Esimesel juhul kui satelliit asub maajaama nägemisulatuses tuleb arvutada vastava satelliidi parameetrid, et operaatoril oleks võimalik vastava satelliidiga suhelda. Kuna kontakti alguse ja lõpuaga pole siis on nendeks ette antud ajavahemiku algus ja lõpp. Antud testijuht on samaväärne olukorraga, kus kasutaja poolt sisestatud ajavahemik on lühem kui juba käimas oleva kontakti kestvus. Testimise tulemused on toodud tabelis 4.

Tabel 4. Tabelis on toodud geostatsionaarse satelliidi XTAR\_EUR kontakti parameetrid. Maajaam asub Tartus koordinaatidel 58,3° N ja 26,73° E. Testimise ajavahemikuks oli 27.04.2013 6:16:00 kuni 27.04.2013 13:35:00 UTC järgi. Test on läbi viidud 27.04.2013, mil võetud andmed GPredictist.

Parameeter	GPredict tulemus	Kontakti arvutamise mooduli tulemus	Lubatud viga
Alguse aeg	2013-04-27T06:16:00	2013-04-27T06:16:00	-
Lõpu aeg	2013-04-27T13:35:00	2013-04-27T13:35:00	-
Elevatsioon (°)	23,80	23,75	0,5
Signaalikadu (dB)	164,26	164,36	0,5
Maajaama ja satelliidi vaheline kaugus (km)	39182	39179	20
Signaali viivitus (msek)	130,70	130,68	0,5
Oribiidi number	3005	3005	0

Teisel juhul kui satelliit on maajaama nägemisulatusest väljas tuleb kontrollida, et tarkvaramoodul ei saaks ühtegi viga ega tagastaks kasutajale mingit infot. Testimiseks kasutati LES-9 geostatsionaarset satelliiti ja maajaamaks oli Tartu maajaam koordinaatidega 58,3° N ja 26,73° E ja kõrguseks merepinnast on 59 meetrit. Testimise ajavahemikuks oli 27.04.2013 6:16:00 kuni 27.04.2013 13:35:00 UTC järgi. Testimise tulemusena ei toimunud ühtegi kontakti, sest satelliit asub maajaama vaateväljast väljas.

Kolmas testijuht on eeldatavalt kõige kasutatavam olukord antud tarkvaramooduli jaoks. Sellel juhul arvutatakse välja kasutaja poolt valitud ajavahemikul kõik toimuvad kontaktid ja kontakte iseloomustavad parameetrid. Antud testijuhtu katsetades arvutatakse välja vajalikud parameetrid ja võrreldakse neid GPredictist saadud tulemustega. Kontakti arvutamise mooduli poolt saadud tulemused jäid võrreldes GPredictiga lubatud vigade raamesse. Testimise tulemused on toodud tabelis 5.

Tabel 5. Tabelis on toodud satelliidi Compass-1 kontakti parameetrid. Maaajaam asub Tartus koordinaatidel 58,3° N ja 26,73° E. Testimise ajavahemikuks oli 12.04.2013 11:16:00 kuni 12.04.2013 11:35:00 UTC järgi. Test on läbi viidud 12.04.2013, samal ajal võetud ka andmed GPredictist.

Parameeter	GPredict tulemus	Kontakti arvutamise mooduli tulemus	Lubatud viga
Alguse aeg	2013-04-12T11:22:33	2013-04-12T11:22:37	30 sek
Lõpu aeg	2013-04-12T11:30:23	2013-04-12T11:30:43	30 sek
Alguse asimuut (°)	6,18	6,10	0,5
Lõpu asimuut (°)	286,10	286,15	0,5
Maksimaalne elevatsioon (°)	6,33	6,30	0,5
Doppleri nihe alguses (Hz)	1491	1487	10
Doppleri nihe lõpus (Hz)	-1466	-1460	10
Signaalikadu horisondil (dB)	141,61	141,66	0,5
Signaalikadu seniidis (dB)	139,47	139,54	0,5
Kaugus horisondil (km)	2875	2870	20
Kaugus seniidis (km)	2257	2251	20
Signaali viivitus horisondil (msek)	9,59	9,57	0,5
Signaali viivitus seniidis (msek)	7,53	7,50	0,5
Päikese käes (JAH/EI)	JAH	JAH	PUUDUB
Orbiidinumber	26820	26820	0

Neljanda testijuhu eesmärk on sama mis eelmiselgi, kuid erinevus seisneb selles, et kasutatakse mitte polaarorbiidil asetsevat satelliiti ISS. Antud testijuht on oluline, sest see näitab, et loodud tarkvaramoodul töötab korrektselt erinevate orbiitidega ja sellest lähtuvalt on kasutatav erinevate satelliidi tüüpide puhul. Testimise tulemused on toodud tabelis 6.

Tabel 6. Tabelis on toodud satelliidi ISS kontakti parameetrid. Maajaam asub Tartus koordinaatidel 58,3° N ja 26,73° E. Testimise ajavahemikuks oli 25.04.2013 11:25:00 kuni 25.04.2013 11:37:00 UTC järgi. Test on läbi viidud 25.04.2013, samal ajal võetud ka andmed GPredictist.

Parameeter	GPredict tulemus	Kontakti arvutamise mooduli tulemus	Lubatud viga
Alguse aeg	2013-04-25T11:27:27	2013-04-25T11:27:29	30 sek
Lõpu aeg	2013-04-25T11:35:13	2013-04-25T11:35:09	30 sek
Alguse asimuut (°)	188,90	188,58	0,5
Lõpu asimuut (°)	96,66	96,67	0,5
Maksimaalne elevatsioon (°)	7,41	7,32	0,5
Doppleri nihe alguses (Hz)	1648	1642	10
Doppleri nihe lõpus (Hz)	-1653	-1647	10
Signaalikadu horisondil (dB)	139,75	139,85	0,5
Signaalikadu seniidis (dB)	136,81	136,94	0,5
Kaugus horisondil (km)	2331	2331	20
Kaugus seniidis (km)	1662	1668	20
Signaali viivitus horisondil (msek)	7,77	7,77	0,5
Signaali viivitus seniidis (msek)	5,54	5,56	0,5
Päikese käes (JAH/EI)	JAH	JAH	PUUDUB
Orbiidinumbr	82651	82651	0

Testimine oli edukas, kõik testid töötasid korrektselt ja vigu ei esinenud. Ühiktestidega saavutatud koodikattuvus on väga hea, üle 90%. Arvesse on võetud *ContactCalculation* klassi, mis on põhiline kontakti arvutamise mooduli osa. Järgnevatel joonistel on kujutatud koodikattuvus.

ContactCalculationTest (4.05.2013 12:00:32)

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
eu.estcube.contact	82,6 %	3 372	711	4 083
src/main/java	93,4 %	1 498	106	1 604
eu.estcube.contact	93,4 %	1 498	106	1 604
ContactEventData.java	100,0 %	224	0	224
ContactCalculation.java	92,3 %	1 274	106	1 380
src/test/java	75,6 %	1 874	605	2 479
eu.estcube.contact	75,6 %	1 874	605	2 479
ContactCalculationTest.java	75,6 %	1 874	605	2 479
ContactCalculationTest	75,6 %	1 874	605	2 479

Joonis 14. Väljavõte Eclipse konsoolist testi kattuvust mõõtes.

**Clover Coverage Report - ESTCube Build 0.0.1-SNAPSHOT (Aggregated)**  
 Coverage timestamp: Sat May 11 2013 04:09:59 EEST

App Test Results Clouds

Overview **Package** File

FRAMES NO FRAMES SHOW HELP

**Statistics for package eu.estcube.contact:**  
 Stmts: 240 LOC: 853 Total cmp: 92 Stmts/Method: 3  
 Branches: 22 NCLOC: 567 Cmp density: 0.38 Methods/Class: 40  
 Methods: 80 Files: 2 Avg method cmp: 1.15  
 Classes: 2

Package	Files	% Filtered	Average Method Complexity	Uncovered Elements	TOTAL Coverage
eu.estcube.contact	2	0%	1.15	24	93%

Class	Methods	% Filtered	Average Method Complexity	Uncovered Elements	TOTAL Coverage
ContactCalculation	17	0%	1.71	24	88.9%
ContactEventData	63	0%	1	0	100%

Report generated by Clover Code Coverage v3.1.7  
 Sat May 11 2013 04:10:07 EEST. Clover: Community License registered to Estonian Student Satellite.

Joonis 15. Clover testiraport kontakti arvutamise mooduli testikattuvuse jaoks.

## 5. Kokkuvõte

Käesoleva bakalaureusetöö praktiliseks osaks oli kavandada ja luua maajaama ja satelliidi vahelise kontakti aegade ning kontakti iseloomustavate parameetrite arvutamise tarkvaramoodul. Samuti tuli loodud rakendust ka testida. Lisaks pakuti teoreetilised lahendused ühe satelliidi ja mitme maajaama juhtumi optimeerimisele.

Käesoleva bakalaureusetöö raames sai täidetud järgmised eesmärgid:

- loodud tarkvaramoodul, mis arvutab maajaama ja satelliidi vahelisi kontakti aegasid,
- loodud tarkvaramooduli osa, mis arvutab satelliidi ülelennu kvaliteeti iseloomustavad parameetrid,
- testitud rakendust kasutades ühikteste ja reaalse maajaama ja satelliidiga olukorras,
- saades testikattuvuseks ühiktestidega üle 90%,
- pakuti välja teoreetiline lahendus ühe satelliidi ja mitme maajaama juhtumi optimeerimisele.

Töö raames sai loodud kontakti arvutamise tarkvaramoodul Eesti missiooni juhtimiskeskuse tarkvara MCS jaoks, sest seni puudus MCSil võimalus satelliitide ülelende jälgida. Satelliitide jälgimise ja sidepidamise programme on teisigi, näiteks PREDICT, GPredict ja GENSO. MCS erineb neist selle poolest, et tegemist on veebipõhise missiooni juhtimise süsteemiga, see teeb MCS kasutamise palju lihtsamaks ja on ligipääsetav kõikjalt, kus on internetiühendus.

Kontakti arvutamise mooduli ülesandeks on arvutada maajaama ja satelliidi vahelised kontaktide ajad kindlal ajavahemikul ning arvutada ülelendu iseloomustavad parameetrid. Antud moodul kasutab arvutuste tegemiseks orbitaaldünaamika teeki OREKIT. Kontakti aegade arvutamiseks kasutatakse OREKITi klassi *EventDetector*, mille eesmärgiks on tuvastada kontakti alguse ja lõpu aeg. Ülelendu iseloomustavate parameetrite arvutamisel kasutatakse vastavaid OREKITi meetodeid. Kontakti arvutamise mooduli poolt saadud tulemused salvestatakse kindlasse Java objekti *ContactEventData*. Ühtsesse kindlasse objekti salvestatuna on lihtsam andmeid edastada. Kontakti arvutamise moodul saadab andmed edasi antennide pööramise moodulisse, mis saadud informatsiooni abil saab paika sättida antennid ja raadiosagedused, et olla valmis üle lendava satelliidiga sidet pidada. Antud mooduli puhul on oluline, et satelliidi operaatoritel oleks võimalik ette planeerida satelliidiga suhtlemist, eelseadistada vajalikud käsud ja sättida paika maajaamad. Testimise käigus sai katsetatud

programmi käitumist erinevates olukordades, lisaks sai kood kaetud ka ühiktestidega. Testimise erinevateks testijuhtudeks olid erinevat tüüpi satelliidid ja erinevatel asukohtadel olevad maajaamad. Kõik testimise tulemused olid positiivsed. Testikattuvus ühiktestidega oli üle 90%.

Samas leiti ka teoreetiline lahendus ühe satelliidi ja mitme maajaama ülelennu strateegia optimeerimisele. Selline olukord tekib siis, kui ühte satelliiti soovitakse jälgida mitme maajaamaga. Mitme maajaama kasutamine ühe satelliidi jälgimiseks annab võimaluse planeerida pikemaid ning kvaliteetsemaid ülelende või rohkem kontakte ööpäeva jooksul. Optimeerimise puhul tuleb arvestada täidetava ülesande iseloomu ja maajaamade ning satelliidi omavahelist sobivust. Töö edasiarenduseks oleks optimeerimise tarkvaraline lahendamine ja sellega seoses MCS süsteemi edasiarendus.



# **Using space-dynamics library OREKIT for optimal satellite contacts planning**

**Mart Vellak**

**Bachelor's thesis (6 EAP)**

## **Summary**

Since 7<sup>th</sup> of May, Estonia has its own satellite ESTCube-1 in space. ESTCube-1 is a small 10x10x10 cm CubeSat created by Estonian students. Aim of the ESTCube-1 mission is to test the electric solar wind sail. For a successful mission, the satellite operators have to get information from satellite. To exchange data with satellite there has to be radio contact between the ground station and the satellite. All this communication and ground station control is being done by Estonian mission control system MCS.

MCS is based on open-source Hummingbird software framework, which is still in development and which was missing a component for planning satellite contacts. The component, which is able to calculate contact times between satellites and ground stations. Although there are other satellite tracking software like PREDICT, GPredict and GENSO, MCS differs from all of these. PREDICT is command-line based satellite tracking software, which allows tracking multiple satellites with one ground station. GPredict is similar to PREDICT, but has graphical user interface. GENSO has possibility to use multiple ground stations to track satellites, using ground station network. The biggest advantage of MCS is that it is web-based mission control system, which gives opportunity to use it in every place with Internet connection. MCS is system that connects multiple ground stations in one big network and so gives opportunity to contact with satellite more often.

Aim of this thesis was to design and program software for calculating contact times between ground station and satellite. Also this software has to calculate parameters for characterizing contact between satellite and ground station and created software had to be tested. Another point in this thesis was to find solution for one satellite and several ground stations optimizing problem.

Contact calculation module purpose is to calculate satellite and ground station contact times over specific time period and calculate parameters for describing contact. Contact calculation module uses free low-level space dynamics Java library OREKIT. OREKIT is used to calculate contact times, using its built-in class *EventDetector*, that detects contact start and end events. Also OREKIT is used for calculating parameters for describing contact. All calculated parameters and contact start and end time is saved in specific Java object *ContactEventData*. Given module is important, because satellite operators have opportunity to plan communication with satellite, pre-set commands for satellite. Data calculated by given module sends it to antenna controlling system so it can set ground station antennas and radio frequencies for coming satellite. Testing covered different cases like non-polar orbit satellites or geostationary satellites and ground stations in different locations – in North-Pole and in Tartu. Also program code was covered with unit tests. Unit tests coverage was more than 90%. All test passed and results were positive.

There was also theoretical solution for one satellite and multiple ground stations strategy optimization. This situation occurs when operator needs to use several ground stations for tracking one satellite. Using multiple ground stations gives opportunity to plan longer or higher quality contacts. With optimization software has to understand format of tasks and find matching satellite and ground station pairs. For further development would be designing and programming optimization software.

## Viited

1. Munakata, R. (2009). CubeSat Design Specification Rev. 12: The CubeSat Program, Cal Poly SLO.
2. M. Pajusalu, R. Rantsus, M. Pelakauskas, A. Leitu, E. Ilbis, J. Kalde, H. Lillmaa, R. Reinumägi, K. Voormansik, K. Zālīte, V. Allik, M. Noorma, S. Lätt, Design of the Electrical Power System for the ESTCube-1 satellite. *Latvian Journal of Physics and Technical Sciences*. 2012. 49(3), 16–24.
3. Mihkel Pajusalu, Erik Ilbis, Jaanus Kalde, Henri Lillmaa, Risto Reinumägi, Ramon Rantsus, Martynas Pelakauskas, Ahto Leitu, Viljo Allik, Mart Noorma, Silver Lätt, Jouni Envall, Electrical Power System for ESTCube-1: a Fault-tolerant COTS Solution. In: 63rd International Astronautical Congress. 2012.
4. Janhunen, P., P.K. Toivanen, J. Polkko, S. Merikallio, P. Salminen, E. Haeggström, H. Seppänen, R. Kurppa, J. Ukkonen, S. Kiprich, G. Thornell, H. Kratz, L. Richter, O. Krömer, R. Rosta, M. Noorma, J. Envall, S. Lätt, G. Mengali, A.A. Quarta, H. Koivisto, O. Tarvainen, T. Kalvas, J. Kauppinen, A. Nuottajärvi and A. Obraztsov, Electric solar wind sail: Towards test missions (Invited article), *Rev. Sci. Instrum.*, 81, 111301, 2010.
5. William Tyrell Thomson. *Introduction to space dynamics*. 1986  
[http://books.google.ee/books?hl=en&lr=&id=vRNdPve1XuWC&oi=fnd&pg=PA1&dq=space+dynamics&ots=FfihFZoyMz&sig=6gTiVIZQw46YgG1\\_nuqQf\\_xupzE&redir\\_esc=y](http://books.google.ee/books?hl=en&lr=&id=vRNdPve1XuWC&oi=fnd&pg=PA1&dq=space+dynamics&ots=FfihFZoyMz&sig=6gTiVIZQw46YgG1_nuqQf_xupzE&redir_esc=y) (Viimati külastatud 06.05.2013)
6. TLE formaat <http://www.celestrak.com/columns/v04n03> (Viimati külastatud 06.05.2013)
7. Compass-1 TLE <http://celestrak.com/NORAD/elements/cubesat.txt> (Viimati külastatud 06.05.2013)
8. Ennustusmudelid <http://znark.com/sat/models.html> (Viimati külastatud 06.05.2013)
9. OREKIT <https://www.orekit.org> (Viimati külastatud 04.04.2013)
10. Predict <http://www.qsl.net/kd2bd/predict.html> (Viimati külastatud 06.05.2013)
11. GPredict <http://gpredict.oz9aec.net> (Viimati külastatud 05.05.2013)
12. G. Shirville, B. Klofas, GENSO: A Global Ground Station Network  
[http://www.klofas.com/papers/AMSAT\\_2007.pdf](http://www.klofas.com/papers/AMSAT_2007.pdf) (Viimati külastatud 06.05.2013)
13. GENSO [http://www.esa.int/Education/How\\_GENSO\\_works](http://www.esa.int/Education/How_GENSO_works) (Viimati külastatud 06.05.2013)

14. Hummingbird <http://www.hbird.de> (Viimati külastatud 06.05.2013)
15. J. C. Pemberton, F. Galiber. A Constraint-Based Approach to Satellite Scheduling [http://www.cs.sfu.ca/CourseCentral/827/havens/papers/topic%2312\(SatelliteScheduling\)/dimacs98.pdf](http://www.cs.sfu.ca/CourseCentral/827/havens/papers/topic%2312(SatelliteScheduling)/dimacs98.pdf) (Viimati külastatud 06.05.2013)
16. S. Spangelo, J. Cutler. Optimization of Single-Satellite Operational Schedules Towards Enhanced Communication Capacity [http://exploration.engin.umich.edu/blog/wp-content/uploads/2011/10/GNC\\_2012\\_Sat\\_Ops\\_Paper\\_New.pdf](http://exploration.engin.umich.edu/blog/wp-content/uploads/2011/10/GNC_2012_Sat_Ops_Paper_New.pdf) (Viimati külastatud 06.05.2013)
17. ActiveMQ <http://activemq.apache.org/> (Viimati külastatud 06.05.2013)
18. Free-space path loss <http://www.ece.uvic.ca/~peterd/35001/ass1a/node1.html> (Viimati külastatud 08.05.2013)

## **Lisa 1. Kontakti arvutamise mooduli kood**

Kontakti arvutamise mooduli kood on lisatud CD-1.

## **Lisa 2. Kontakti arvutamise mooduli testiplaan**

### **Unit/component Test Plan**

### **Contact calculation**

**(MCS)**

### **Software Quality Assurance Plan**

## Document History and Distribution


### 1. Revision History

Revision #	Revision Date	Description of Change	Author
1.0	10.04.2013	Created document	Mart Vellak
1.1	11.04.2013	Updated document	Mart Vellak
1.2	23.04.2013	Update document	Mart Vellak

### 2. Distribution

Recipient Name	Recipient Organization	Distribution Method

# Table of Contents

- 1. Introduction
  - 1.1 Objectives
  - 1.2 Scope
  - 1.3 Reference Material
- 2. Test items
- 3. Features to be tested
- 4. Features not to be tested
- 5. Approach
- 6. Pass/Fail criteria
- 7. Testing process
  - 7.1 Test deliverables
  - 7.2 Resources
  - 7.3 Schedule
- 8. Environmental Requirements
  - 8.1 Hardware
  - 8.2 Software
  - 8.3 Security
  - 8.4 Tools
  - 8.5 Publications
  - 8.6 Risks and Assumptions
- 9. Change Management Procedures
- 10. Plan Approvals 



# 1. Introduction

## 1.1. Objectives

The objective is to perform testing for contact calculation system used by mission control system (MCS) in order to verify that calculated contact times are accurate and parameters for contact are correct. Contact calculation system will be used to give data about occurring contact for different MCS parts.

## 1.2. Scope

Contact calculation unit/component test plan defines only basic functional testing approach. Test scope includes functional testing for different satellite types and for different contact parameters calculations.

## 1.3. Reference Material

- <http://gpredict.oz9aec.net/>

## 2. Test items

- Contact calculation software

## 3. Features to be tested

The following features will be tested:

- Software ability to calculate contact times
- Software ability to calculate parameters for contacts

## 4. Features not to be tested

- Any other features

## 5. Approach

Unit/component testing will be divided in two parts. All parts are tested code-based and using unit tests. Unit tests will cover some sample test cases.

1. Software ability to calculate contact times – software will calculate contact times and these will be compared to GPredict values. Software will get required input values: TLE, ground station coordinates and time period when contact should occur. Calculate values have to match GPredict values
2. Software ability to calculate parameters for contacts – software will calculate contact parameters like start and end azimuth, max elevation, signal loss, etc. Calculated parameters will be compared to GPredict values and they have to match.

Case 1: Geostationary satellite calculation doesn't fail

Case 2: Cube satellite contact times

Case 3: Cube satellite contact parameters

Case 4: Ground station is located in North-Pole

Case 5: Using satellite with non-polar orbit

Unit tests will be run:

1) Software code is updated

2) On request

## 6. Pass/Fail criteria

For first part testing pass criteria:

- No major differences with GPredict

For second part testing pass criteria:

- No major differences with GPredict

Fail criteria:

- Software fails – gets error
- Major differences with GPredict
- Illogical results

Approval criteria:

- 100% of tests in test suite are executed

## 7. Testing process

All test cases are executed fully manually. After test cases have been executed, test results are analyzed. All errors will be reported.

Test Case ID:	<b>ContactTimes001</b>		
Test case name:	<b>Geostationary satellite</b>		
Created:	Mart Vellak	Last edited:	
Date:	11.04.2013	Date:	

Description:	This test will test geostationary satellite contact calculation doesn't fail
Preconditions:	Test file ContactCalculationsTest.java is opened

Priority:	1 (always run)
Steps:	<ol style="list-style-type: none"> <li>1. Run method testGeoStatsSat() in test file - uses precoded test-data taken from GPredict</li> <li>2. Run method testGeoStatsSatInRange() in test file – uses precoded testdata taken from GPredict</li> </ol>
Expected results:	1. Calculation doesn't fail;
Exceptions:	-
Special and non-functional requirements:	<ul style="list-style-type: none"> <li>• (out of scope)</li> </ul>
Comments	-

Test Case ID:	<b>ContactTimes002</b>		
Test case name:	<b>Cube satellite</b>		
Created:	Mart Vellak	Last edited:	
Date:	11.04.2013	Date:	

Description:	This test will test if calculated contact times are identical to GPredict values
Preconditions:	Test file ContactCalculationsTest.java is opened
Priority:	1 (always run)
Steps:	<ol style="list-style-type: none"> <li>1. Run method testCubeSat() in test file - uses precoded test-data taken from GPredict</li> </ol>
Expected results:	1. Calculation doesn't fail;
Exceptions:	-
Special and non-functional requirements:	<ul style="list-style-type: none"> <li>• (out of scope)</li> </ul>
Comments	-

Test Case ID:	<b>ContactTimes003</b>		
Test case name:	<b>Contact parameters</b>		
Created:	Mart Vellak	Last edited:	
Date:	11.04.2013	Date:	

Description:	This test will test if calculated contact parameters are identical to GPredict values
Preconditions:	Test file ContactCalculationsTest.java is opened
Priority:	1 (always run)
Steps:	1. Run method testContactParams() in test file - uses precoded test-data taken from GPredict;
Expected results:	1. Calculation doesn't fail;
Exceptions:	-
Special and non-functional requirements:	<ul style="list-style-type: none"> <li>(out of scope)</li> </ul>
Comments	-

Test Case ID:	<b>ContactTimes004</b>		
Test case name:	<b>GS in pole</b>		
Created:	Mart Vellak	Last edited:	
Date:	11.04.2013	Date:	

Description:	This will test if orbit numbers are correct when ground station is in north-pole
Preconditions:	Test file ContactCalculationsTest.java is opened
Priority:	1 (always run)
Steps:	1. Run method testOrbitNumbers() in test file - uses precoded test-data taken from GPredict;

Expected results:	<ol style="list-style-type: none"> <li>1. Calculation doesn't fail;</li> <li>2. Result is same as GPredict;</li> </ol>
Exceptions:	-
Special and non-functional requirements:	<ul style="list-style-type: none"> <li>• (out of scope)</li> </ul>
Comments	-

Test Case ID:	<b>ContactTimes005</b>		
Test case name:	<b>Non-polar orbit</b>		
Created:	Mart Vellak	Last edited:	
Date:	11.04.2013	Date:	

Description:	This test will test if calculated contact times are identical to GPredict values, if we are using satellite with non-polar orbit
Preconditions:	Test file ContactCalculationsTest.java is opened
Priority:	1 (always run)
Steps:	<ol style="list-style-type: none"> <li>1. Run method testISS() in test file - uses precoded test-data taken from GPredict</li> </ol>
Expected results:	<ol style="list-style-type: none"> <li>1. Calculation doesn't fail;</li> </ol>
Exceptions:	-
Special and non-functional requirements:	<ul style="list-style-type: none"> <li>• (out of scope)</li> </ul>
Comments	-

## 7.1 Test deliverables

- Test report
- Test results

## **7.2 Resources**

- Human: tester needed for testing
- Tools: environment for testing

## **7.3 Schedule**

System test suite shall be executed every time system and/or software gets updated or when need is triggered.

## **8. Environmental Requirements**

System test plan has following requirements:

### **8.1 Hardware**

- Ground control system.

### **8.2 Software**

- Latest version of software;
- Ground control system software to play antenna moving schemas.

### **8.3 Risks and Assumptions**

- Issues to set up software and test environment;
- Software is delayed or is in bad condition;
- No human resources to run tests;
- Hardware not available;
- Other priority issues.

## **9. Change Management Procedures**

This test plan shall be updated every time major changes are made in scope by its owner, test manager, tester or project manager. A review is required in case major change has been made to the scope and this document.

## **10. Plan Approvals**

*(This plan has not yet been approved)*

## **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

**Mina Mart Vellak**

(sünnikuupäev: 04.10.1989)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **Orbitaaldünaamika teegi OREKIT kasutamine satelliitide kontaktide optimaalseks planeerimiseks**, mille juhendajad on **Urmas Kvell, Lauri Kimmel ja Tõnis Eenmäe**.
  - 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **13.05.2013**