



IEGULDĪJUMS TAVĀ NĀKOTNĒ



EIROPAS SAVIENĪBA

Eiropas Sociālais fonds

Prioritāte: 1.1. Augstākā izglītība un zinātne

Pasākums: 1.1.1. Zinātnes un pētniecības potenciāla attīstība

Aktivitāte: 1.1.1.2. Cilvēkresursu piesaiste zinātnei

Projekta nosaukums: "Viedo sensoru un tīklotu iegulto sistēmu pētījumu un attīstības centrs" (VieSenTIS)

Līguma noslēgšanas datums: 27.11.2009.g.

Projekta sākuma datums: 01.01.2010.g.

Projekta beigu datums: 31.12.2012.g.

Vienošanās Nr.2009/0219/1DP/1.1.1.2.0/09/APIA/VIAA/020

Eiropas Savienības fonda finansējuma saņēmējs: Elektronikas un datorzinātņu institūts (EDI)

ZINĀTNISKĀ PĒTĪJUMA PROGRESU APLIECINOŠĀ DOKUMENTĀCIJA

Pārskata numurs Nr.10. par periodu no 01.04.2012.g līdz 30.06.2012.g.

<i>Projekta zinātniskais vadītājs:</i>	Leo Seļāvo, Ph.D. datorzinātnēs.
<i>Pētījuma projekta izpildītāju saraksts:</i>	Modris Greitāns, Dr.sc.comp.
	Uldis Grunde, topošais zinātnieks, Mg.Fizikā.
	Atis Elsts, doktorants.
	Rolands Šāvelis, doktorants.
	Oļegs Ņikišins, doktorants.
	Gatis Šūpols, Mg. Elektronikā.
	Kaspars Ozols, doktorants.
	Andris Gordjušins, Mg. Elektronikā.
	Zanda Seržāne, Mg. Elektronikā.
	Ivars Driķis, Dr.phys.
	Rinalds Ruskuls, Mg. Elektronikā.
	Reinholds Zviedris, doktorants.
	Georgijs Kanonirs, Mg. Datorzinātnēs.
	Arturs Seļivanovs, Mg. Elektronikā.

Saturs

Ievads	3
VieSenTIS	3
1 BioSen grupas atskaite	4
1.1 Ievads	4
1.2 Saistītie risinājumi	5
1.3 Risinājums	5
1.3.1 MSLBP operatora principi	5
1.3.2 Parametru svarošana	5
1.4 Eksperimenti un-vai testi	6
1.5 Rezultāti	6
1.6 Kopsavilkums un secinājumi	8
1.7 Nākotnes perspektīvas	8
Literatūra	8
2 CarMote grupas atskaite	9
2.1 Ievads	9
2.2 Saistītie risinājumi	9
2.3 Risinājums	10
2.3.1 Navigācijas sistēma	10
2.4 Eksperimenti un-vai testi	11
2.5 Kopsavilkums un secinājumi	14
2.6 Nākotnes perspektīvas	15
Literatūra	15
3 EdiMote grupa	16
3.1 Ievads	16
3.2 Risinājums	17
3.2.1 Perifērijas	17
3.2.2 Testa moduļi	19
3.2.3 Moduļu saskarne	20
3.3 Arhitektūras implementācija	21
3.3.1 Vadības modulis	21
3.3.2 USB Host funkcionalitātes nodrošināšana	22
3.3.3 EdiMotes pielsēgšana Ethernet tīklam	22
3.4 Kopsavilkums	23
3.5 Nākotnes perspektīvas	24
Literatūra	24

4	MansOS grupas atskaite	25
4.1	Ievads	25
4.2	MansOS arhitektūras novērtējums	26
4.3	MansOS pirmkoda novērtējums	28
4.4	MansOS binārā koda izmēra novērtējums	28
4.5	Pavedienu novērtējums	32
4.6	Kopsavilkums un nākotnes perspektīvas	34
	Literatūra	35
5	SAntArray grupas atskaite	36
5.1	Ievads	36
5.2	Rezultāti	37
5.2.1	Divu elementu ESPAR antena	37
5.2.2	Trīs elementu ESPAR antena	39
5.2.3	Septiņu elementu ESPAR antena	40
5.2.4	Antena ar spiestās plates līniju	43
5.3	Kopsavilkums un secinājumi	44
5.4	Nākotnes perspektīvas	44
	Literatūra	44
6	SenSigA grupas atskaite	46
6.1	Ievads	46
6.2	Ar ASDM pārveidotu signālu atjaunošana	46
6.2.1	Atjaunošana visā signāla garumā	47
6.2.2	Atjaunošana reālā laikā	48
6.3	Rezultāti un secinājumi	49
	Literatūra	50

Ievads

Anotācija

”Viedo sensoru un tīklotu iegulto sistēmu pētījumu un attīstības centrs” (VieSenTIS) ir ESF atbalstīts projekts zinātnisko grupu veidošanai. Šis projekts nodarbojas ar bezvadu sensoru un tīklotu iegulto sistēmu pētniecību. Projekta ietvaros ir sekojošas apakšgrupas: BioSen – sensoru sistēmu pielietojumi biometrijā, CarMote – viedās transporta sensoru sistēmas, EdiMote – jaunas arhitektūras bezvadu sensoru aparatūrai, MansOS – operētājsistēma un programmatūras rīki bezvadu sensoru un iegulto sistēmu atbalstam, SAntArray – antenu masīvu pielietojumi tīklotas iegultas sistēmās, un SenSigA – sensoru signālu apstrādes metodes. Šajā dokumentā izklāstītas zinātniskās pētniecības aktivitātes un rezultāti augšminētajās grupās. Pielikumā ir iekļautas šajā periodā iesniegtās un apstiprinātas zinātniskās publikācijas. Projektu atbalsta Eiropas Savienības fondi, līguma Nr. 2009/0219/1DP/1.1.1.2.0/09/APIA/VIAA/020.

VieSenTIS

VieSenTIS grupas galvenais pētniecības virziens ir tīklotu iegulto sistēmu un viedu sensoru risinājumi un pielietojumi cilvēku dzīves kvalitātes uzlabošanai, industrijā un citās nozarēs. Miniaturas iegultās sistēmas ir ar ierobežotiem resursiem, piemēram, enerģijas avots bieži vien ir neliela baterija. Tāpēc ir aktuāli pētīt kā šos resursus pēc iespējas efektīvi izmantot, kā arī radīt rīkus jaunas aparatūras un programmatūras izstrādei. Bez tam, bezvadu sensoriem rodami pielietojumi arvien jaunas dzīves sfērās, kas prasa gan specifisku signāla apstrādi gan pielietojuma metožu pētniecību.

VieSenTIS projektā ir sekojošas apakšgrupas saskaņā ar pētnieku individuālo pieredzi un interesēm:

- BioSen – sensoru sistēmu pielietojumi biometrijā;
- CarMote – viedās transporta sensoru sistēmas;
- EdiMote – jaunas arhitektūras bezvadu sensoru aparatūrai;
- MansOS – operētājsistēma un programmatūra bezvadu sensoru un iegulto sistēmu atbalstam;
- SAntArray – antenu masīvu pielietojumi tīklotas iegultas sistēmās;
- SenSigA – sensoru signālu apstrādes metodes.

Turpmākajās nodaļās ir apkopoti galvenie sasniegumi minētajos virzienos.

Nodaļa 1

BioSen grupas atskaite

Anotācija

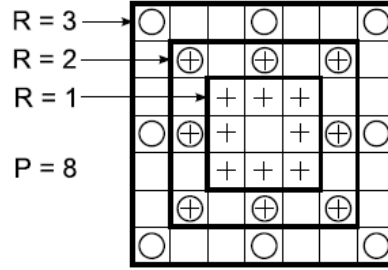
BioSen grupas ietvaros tika izveidots jauns sejas atpazīšanas algoritms, kas balstās uz tekstūras operatora Multi-scale Local Binary Patterns (MSLBP). MSLBP operators pirmo reizi tiek apskatīts [3]. Efektīva MSLBP kombinācija ar ieejas attēla filtrāciju un svaru piemeklēšanu objektu aprakstošiem parametriem ļauj paaugstināt sejas atpazīšanas sistēmas precizitāti. Attēlu filtrācija un MSLBP ļauj uzlabot aprakstošo parametru stabilitāti pie dažādiem seju mērogiem. Papildinātais svaru piemeklēšanas algoritms efektīvi pastiprina atpazīšanas procesam svarīgas komponentes un novājina mazsvarīgas. Sejas atpazīšanas algoritma parametri tika notestēti uz **color FERET** [1] datubāzes.

1.1 Ievads

Tipiska sejas atpazīšanas sistēma sastāv no diviem galvenajiem moduļiem: pirmais ir sejas aprakstošo **parametru iegūšanas** bloks un otrais ir **klasifikācijas modulis**. Atskaitē apskatīti abi minētie aspekti. Parametru iegūšanas bloks balstās uz MSLBP un ieejas attēla filtrācijas, bet klasifikācijas moduļa pamatā ir Weighted Nearest Neighbour klasifikators (WNNC).

Tekstūras operatoru pielietojums sejas atpazīšanas jomā ir diezgan izplatīts. Klasiskais piemērs ir uz Local Binary Patterns (LBP) balstīta sejas atpazīšana [2]. Tomēr fakts, ka materiāla tekstūras vizuālais izskats ir atkarīgs no objekta optiskas palielināšanas, apgrūtina LBP pielietojamību sejas atpazīšanas jomā. Šis aspekts tiek minēts [3] (MSLBP) un [4], kur ir prezentēts Multi-scale Block LBP (MB-LBP). Gan MSLBP, gan MB-LBP ļauj paaugstināt sistēmas noturību pret ieejas attēla / sejas mērogu. Tika apvienotas abas paradigmas un rezultātā tika iegūta augsta atpazīšanas precizitāte.

Iepriekšējā atskaitē tika aprakstīts algoritms, kas ir paredzēts svaru piemeklēšanai objektu aprakstošiem parametriem. Algoritms tika implementēts pēc **mini-batch** principa, kas katrā iterācijā nejauši izvēlas apmācības datus. Nejaušības pieeja tiek aizstāta ar speciālu apmācības datu izvēlēšanas procedūru. Faktiski apmācības procesā tiek izmantoti piemēri, uz kuriem sejas atpazīšanas sistēma var kļūdīties ar vislielāko varbūtību.



Att. 1.1: MSLBP operatora aprēķināšanas princips. ($P = 8$), $R = \{1, 2, 3\}$.

1.2 Saistītie risinājumi

Piedāvātās metodikas ir Local Binary Patterns operatora paplašinājums [2]. Tomēr kombinācija no MSLBP un ieejas attēla filtrācijas ļauj izveidot stabilāku deskriptoru un ievērojami paaugstināt sejas atpazīšanas sistēmas precizitāti.

1.3 Risinājums

1.3.1 MSLBP operatora principi

MSLBP operators ir ļoti līdzīgs LBP transformācijai, bet MSLBP gadījumā ieejas attēla lokālajā reģionā tiek izvēlēti vairāki LBP operatora rādiusi R (Attēls 1.1) un rezultāti tiek kombinēti summējot attiecīgās histogrammas. Katrs lokālais MSLBP reģions tiek aprakstīts ar sekojošiem parametriem: ($P, R = \{R_1, R_2, \dots, R_{n_R}\}$), kur n_R ir rādiusu numuri, kas tiek izmantoti MSLBP operatorā. Pēc transformācijas katrs pikselis ieejas attēlā tiek aprakstīts ar n_R vērtībām.

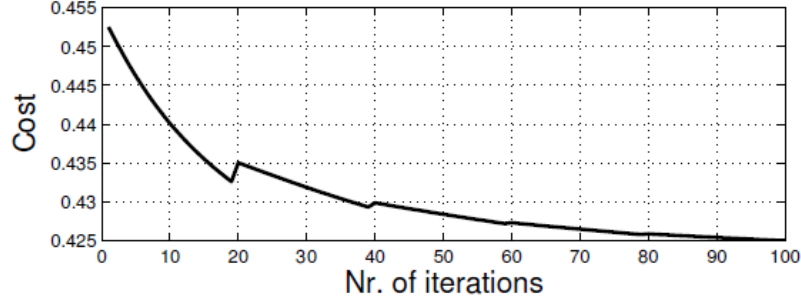
Katrai R vērtībai var izrēķināt savu LBP histogrammu: $H^{(1)}, H^{(2)}, \dots, H^{(n_R)}$. MSLBP histogramma tiek iegūta summējot LBP histogrammas:

$$H^{MSLBP} = \sum_{i=1}^{n_R} H^{(i)}. \quad (1.1)$$

Iegūtais H^{MSLBP} vektors efektīvi apraksta gan lokālas, gan globālas sejas īpašības un ir izmantojams atpazīšanas procesā. Lai panāktu vēl labāku MSLBP deskriptora stabilitāti pirms MSLBP aprēķināšanas ieejas attēls tiek nofiltrēts ar **mean** filtru, kas ļauj samazināt lokālās tekstūras ietekmi uz rezultējošo parametru vektoru.

1.3.2 Parametru svarošana

Iepriekšējā atskaitē un publikācijā [5] detalizēti tiek aprakstīts parametru svorošanas algoritms, kas tiek pielietots divos līmeņos: bloku svorošana un parametru svorošana. Šis algoritms tiek ievērojami uzlabots pielietojot apmācības datu izveles procedūru. Procedūras princips ir ļoti vienkāršs: izvēlēties datus, kas ar vislielāko varbūtību var izraisīt atpazīšanas kļūdu. Kad



Att. 1.2: Piemērs Vērtības Funkcijas atkarībai no iterāciju daudzuma.

šie dati ir izvēlēti, tiek veikta iteratīva svaru piemeklēšana ar mērķi palielināt attālumu starp tuvākajām histogrammām, kas pieder dažādiem cilvēkiem. Šie dati tiek pārreķināti ik pēc δ_{iter} iterācijām, kas ļauj veikt optimizāciju ar aktuāliem datiem. Implementējot minēto principu, izdevās ievērojami paaugstināt algoritma stabilitāti un atpazīšanas precizitāti. Optimizācijas algoritma Vērtības Funkcija ar $\delta_{iter} = 20$, tiek parādīta 1.2 attēlā.

1.4 Eksperimenti un-vai testi

Algoritmi tika testēti izmantojot color FERET datubāzi. Vispirms tika piemeklēti MSLBP operatora parametri, kas ļauj dabūt visaugstāko atpazīšanas precizitāti uz **fa** un **fb** color FERET datubāzes apakškopām. MSLBP operators tiek aprakstīts ar $(P, R = \{R_1, R_2, \dots, R_{n_R}\})$, kur $P = 8$, $n_R = 3$ ir fiksētas vērtības, bet R vērtības ir vienādas ar:

$$R = \left\{ \frac{L-1}{2} - 2, \frac{L-1}{2} - 1, \frac{L-1}{2} \right\},$$

kur L - ir MSLBP operatora izmērs, kas tiek piemeklēts.

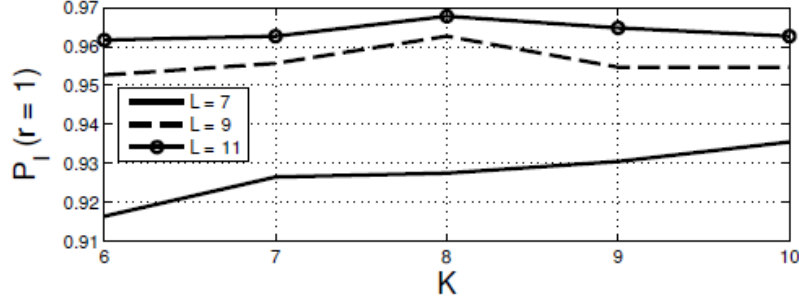
Vēl viens parametrs ir K - faktors, kas nosaka cik reizes MSLBP sejas attēls tiek sadalīts pa horizontāli un vertikāli. Reģionu daudzums ir vienāds ar K^2 . Tiek novērtētas sekojošās vērtības $L = \{7, 9, 11\}$, $K = \{6, 7, 8, 9, 10\}$, attiecīgais atpazīšanas precizitātes grafiks tiek parādīts 1.3 attēlā.

Visaugstākā precizitāte tiek sasniegta pie $L = 11, K = 8$ un ir vienāda ar 96.8%.

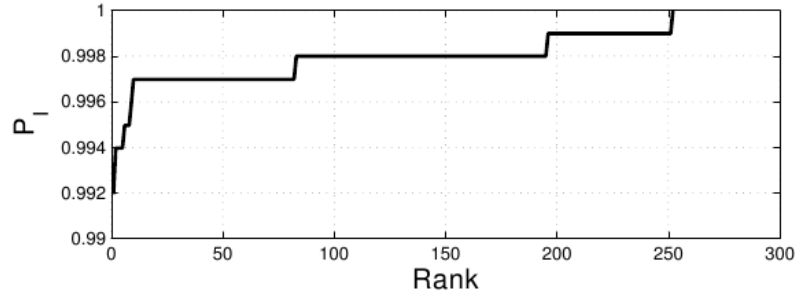
1.5 Rezultāti

Pievienojot ieejas attēla filtrēšanas posmu (mean filter - MF), precizitāte pie $L = 11, K = 8$ uz color FERET datubāzes paaugstinājās līdz 97.8%. Nākošais aspekts, kas tika apskatīts ir empīriskā svaru piemeklēšana katram parametram MSLBP histogrammā (Feature weighting - FW). Šis posms balstās uz sekojošas formulas:

$$w_j = 1 - \frac{(M-1) \sum_{i=1}^M (x_{i,j}^{(1)} - x_{i,j}^{(2)})^2}{M \sum_{i=1}^{M-1} (x_{i,j}^{(1)} - x_{i+1,j}^{(1)})^2}$$



Att. 1.3: Atpazīšanas precizitātes atkarība no parametriem K un L .



Att. 1.4: Cumulative Match Characteristics pēc MSLBP + MF + FW + BW, testējot uz color FERET datubāzes, apakškopas **fa** un **fb**.

$$w_j = \begin{cases} w_j, & \text{if } w_j \geq 0, \\ 0, & \text{if } w_j < 0, \end{cases} \quad (1.2)$$

kur augšējais indekss apzīmējumos $x_{i,j}^{(1)}$ un $x_{i,j}^{(2)}$ reprezentē apmācības piemēra numuru, kas pieder vienam cilvēkam ar numuru datubāzē, i un j ir parametra numurs. Ar svorošanu pēc formulas 1.2 atpazīšanas precizitāte ir 98.1%.

Svaru piemeklēšana katram blokam, pielietojot iteratīvo metodiku, bija nākošais apstrādes etaps (Block weighting - BW), kas uzlaboja atpazīšanas precizitāti līdz 99.2%.

Algoritma etapu apkopojums un to ietekme uz atpazīšanas precizitāti, testējot uz color FERET datubāzes:

- MSLBP ($L = 11$, $K = 8$, $P = 8$, $n_R = 3$): 96.8%,
- MSLBP + mean filter (MF): 97.8%,
- MSLBP + MF + Feature Weighting (FW): 98.1%,
- MSLBP + MF + FW + Block Weighting (BW): **99.2%**.

Rezultējošā statistika tiek attēlota Cumulative Match Characteristics (CMC) formātā, attēls 1.4.

1.6 Kopsavilkums un secinājumi

Piedāvātās sejas atpazīšanas algoritms balstās uz inovatīvās kombinācijas. Labākais atpazīšanas rezultāts ir sasniegts pielietojot sekošas metodes: MSLBP + MF + FW + BW. Atpazīšanas precizitāte šajā gadījumā sasniedza **99.2%**, kas ierindoja pasaules labāko rezultātu augšgalā.

1.7 Nākotnes perspektīvas

Nākotnē tiek plānots apvienot iepriekšējās atskaitēs aprakstītas metodikas ar mērķi izveidot automātisko sejas atpazīšanas algoritmu un novērtēt tā precizitāti uz FERET datubāzes. Automātiskais sejas atpazīšanas algoritms sastāv no trim posmiem: sejas detektēšana, sejas lokalizācija / acu detektēšana un sejas atpazīšana. Sejas detektēšana un lokalizācija balstīsies uz ANN un LBP kombinācijas.

Literatūra

- [1] <http://face.nist.gov/colorferet/>.
- [2] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. *Computer Vision, ECCV 2004 Proceedings, Lecture Notes in Computer Science 3021, Springer*, pages 469–481, 2004.
- [3] C. H. Chan. *Multi-scale Local Binary Pattern Histogram for Face Recognition*. PhD thesis, Centre for Vision, Speech and Signal Processing School of Electronics and Physical Sciences University of Surrey, 2008.
- [4] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li. Learning multi-scale block local binary patterns for face recognition. In *ICB*, pages 828–837, 2007.
- [5] O. Nikisins and M. Greitans. A mini-batch discriminative feature weighting algorithm for lbp - based face recognition. *Proceedings of International Conference on Imaging Systems and Techniques (IST 2012)*, July 2012.

Nodaļa 2

CarMote grupas atskaite

Anotācija

Līdz ar dažādu transporta veidu attīstību, aizvien lielāka nozīme ir viedajām transporta sistēmām, kuras tiek veidotas, izmantojot aparatūras un programmatūras risinājumus. Šie risinājumi ietver gan transporta plūsmu un apkārtējo vidi raksturojošu datu iegūvi, gan arī to apstrādi un koplietošanu. CarMote grupa nodarbojas ar multimodālu datu ieguves un apstrādes problēmu risināšanu transporta nozarē, izmantojot gan dažādu konvenciālu mobilo datortehniku, gan arī speciāli izstrādātas iegultas iekārtas. Šajā darbā ir aprakstītas aktivitātes, kas ir saistītas ar navigācijas sistēmas izstrādi.

2.1 Ievads

CarMote grupas mērķis ir kooperatīva transportlīdzekļa izstrāde, kas spēj efektīvi plānot kustību un pārvietoties līdzīga veida transportlīdzekļu plūsmā. Šī mērķa sasniegšanai tika noteikti vairāki uzdevumi elektronikas, programinženierijas, digitālās signālu apstrādes, bezvadu komunikācijas u.c. jomās. Par kooperatīva transportlīdzekļa izstrādes prasību pamatu tika paņemti GCDC noteikumi [1], kas deva vairākas priekšrocības:

- ekonomēja laiku prasību uzstādīšanai;
- tika izmantots lokāls standarts, kuru izstrādāja un novērtēja profesionālu zinātnieku/izstrādātāju grupa;
- varēja veikt sekmīgu izstrādāta risinājuma testēšanu, salīdzinot to ar citu komandu risinājumiem no dažādiem pasaules institūcijām un valstīm;

2.2 Saistītie risinājumi

Kooperatīvo transportlīdzekļu un autonomo transportlīdzekļu izstrāde ir ļoti populārā tēma pasaules zinātnē [2]. Katru gadu parādās ar vien vairāk transportlīdzekļu, līdz ar to efektīva transportlīdzekļu plūsmas plānošana un drošības paaugstināšana ir aizvien svarīgāka. Tēmas attīstību atbalsta dažādu sacensību rīkošana [3, 4].

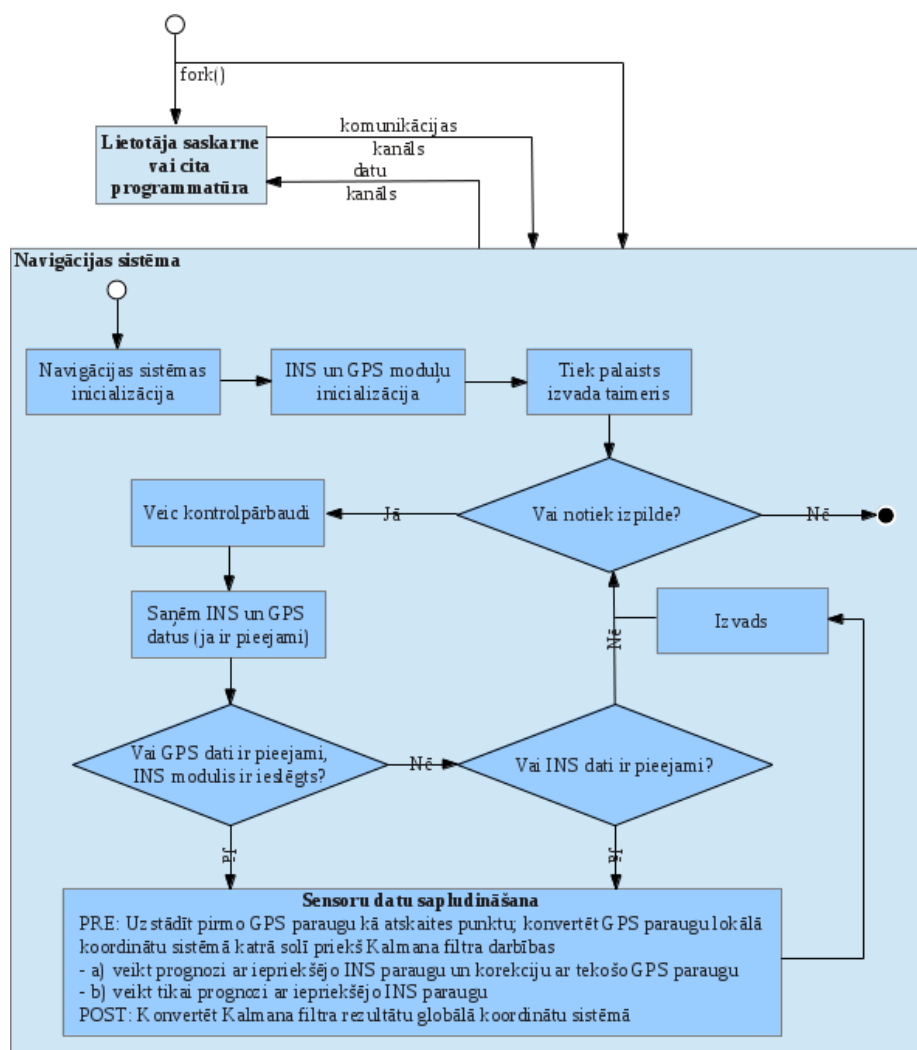
2.3 Risinājums

2.3.1 Navigācijas sistēma

Turpinot navigācijas sistēmas izstrādi tika veikti vairāki pasākumi, kas ir saistīti ar GPS (Globālās Pozicionēšanas Sistēmas), INS (Inerciālās Navigācijas Sistēmas) un digitālā kompasa datu analīzi un navigācijas sistēmas programmatūras pilnveidošanu/atklūdošanu.

Navigācijas sistēmas programmatūras arhitektūra

Apskatīsim navigācijas sistēmas programmatūras arhitektūru (sk. att. 2.1). Palaižot programmatūru tika izveidoti divi paraleli procesi. Pirmā procesā



Att. 2.1: Navigācijas sistēmas programmatūras arhitektūra

darbojas programmatūra (lietotāju saskarne vai cita programmatūra), kas izmanto navigācijas sistēmu, un otrā procesā darbojas pati navigācijas sistēma. Starpprocesu komunikācijas mērķiem tika izveidoti divi kanāli, t.i. komunikācijas kanāls, lai sūtīt komandas navigācijas sistēmai, un datu kanāls, lai saņemt datus no navigācijas sistēmas. Pēc navigācijas sistēmas palaišanas notiek tās inicializācija, INS un GPS moduļu inicializācija, kā arī tiek palaists izvada taimeris. Šis taimeris ir nepieciešams, lai varētu vienmērīgi laikā sūtīt navigācijas sistēmas rezultātu (transportlīdzekļa pozīciju un ātrumu) caur datu kanālu. Kāmr netiek saņemta sistēmas aptures komanda (tiek noteikts kontrolpārbaudes laikā), darbojas navigācijas sistēmas pamatcikls, kur notiek transportlīdzekļa stāvokļa novērtēšana un saistītas darbības.

Transportlīdzekļa stāvokļa novērtēšana notiek 10 reizes sekundē, tomēr šeit varētu būt divi gadījumi. Pirmā gadījumā ir pieejami gan INS, gan GPS dati. Šajā gadījumā tiek veikta sistēmas stāvokļa novērtēšana jeb sensoru datu sapludināšana ar Kalmana filtra palīdzību. Sapludināšanas laikā INS dati tiek izmantoti, lai veikt stāvokļa prognozi, un GPS dati tiek izmantoti, lai veikt stāvokļa korekciju. Savukārt, otrā gadījumā ir pieejami tikai INS dati. Šajā gadījumā tiek veikta tikai sistēmas stāvokļa prognoze. Lai varētu veiksmīgi veikt transportlīdzekļa stāvokļa novērtēšanu abos gadījumos, pozīcija stāvoklī tiek attēlota lokālo koordinātu sistēmā. INS prognoze uzreiz dod rezultātu lokālo koordinātu sistēmā, tomēr GPS paraugus ir jākonvertē no globālās koordinātu sistēmas uz lokālo koordinātu sistēmu. Konvertācijas uzdevuma pildīšanai pirmais uzņemtais GPS paraugs tiek izvēlēts par atskaites punktu un turpmāk katrs nākamais GPS paraugs tiek transformēts, lai iegūt pozīciju attiecībā pret izvēlēto atskaites punktu.

Programmatūras pilnveidošana un atklādošana

Lai efektīvi veikt turpmāko navigācijas sistēmas programmatūras pilnveidošanu un atklādošanu tika nolemts izveidot kontrolējamo vidi, kur varētu atkārtoti veikt eksperimentus ar iepriekš uzņemtiem GPS un INS sensoru datiem. Tāda vide tika izveidota izmantojot octave programmatūru. Sensoru datu sapludināšanai tika izveidota funkcija, kas darbina UKF (Unscented Kalman Filter) filtru. Pirms UKS filtra funkcijas izsaukšanas ir nepieciešama sakotnējo parametru uzstādīšana, lai pēc tam varētu cikliski izsaukt sensoru datu sapludināšanas funkciju un katrā iterācijā novērtēt sistēmas stāvokli.

2.4 Eksperimenti un-vai testi

Eksperimentu veikšanai izveidotā kontrolējamā vidē tika savākti GPS un INS sensoru dati, kā arī izvēlēti 2 testēšanas posmi (sk. att. 2.2). Lai varētu veikt sapludināšanu ar Kalmana filtra palīdzību, GPS paraugus ir jākonvertē no globālās uz lokālo koordinātu sistēmu, ko var izdarīt sekojoši:

- 1. Nodēfinēt atskaites punktu:
Atskaites punkts = pirmais testēšanas posma GPS punkts
- 2. Noteikt attālumus un virzienus, izmantojot atskaites punktu un testēšanas posma GPS punktus:

$$a = \sin\left(\frac{dLat}{2}\right) \cdot \sin\left(\frac{dLat}{2}\right) + \sin\left(\frac{dLng}{2}\right) \cdot \sin\left(\frac{dLng}{2}\right) \cdot \cos(lat1) \cdot \cos(lat2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$distance = R \cdot c \cdot 1000$$

$$y = \sin(dLng) \cdot \cos(lat2)$$

$$x = \cos(lat1) \cdot \sin(lat2) - \sin(lat1) \cdot \cos(lat2) \cdot \cos(dLng)$$

$$bearing = \text{atan2}(y, x)$$

(lat1, lng1) - atskaites punkta koordinātes

(lat2, lng2) - cita punkta koordinātes

dLat, dLng - punktu koordinātu starpība

R = 6371 km - Zemes radiuss

distance - attālums

bearing - virziens

- 3. Noteikt punktu koordinātes lokālā koordinātu sistēmā, izmantojot aprēķinātus attālumus un virzienus:

$$x = distance \cdot \cos(bearing)$$

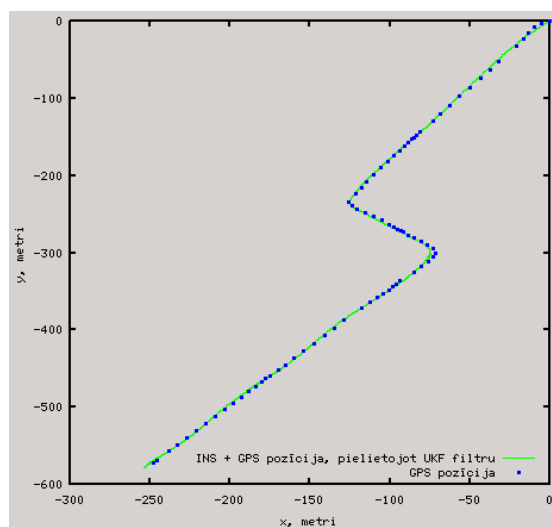
$$y = distance \cdot \sin(bearing)$$

Apskatīsim eksperimentu pirmajā testēšanas posmā (sk. att. 2.3), kur ar ziliem rimbulem tiek attēloti GPS paraugi un ar zaļo līniju - UKF filtra rezultējoša trajektorija. Transportlīdzekļa stavoklī tiek iekļautas pozīcija (x un y ass), ātrums (kustības virzienā), kustības virziens, akselerometra un žiroskopa nobīdes. Kustības virziens prognozes laikā tiek aprēķināts izmantojot žirokopa mērījumus, bet korekcijas laikā - izmantojot magnetometra mērījumus. Šajā gadījumā orientācija tika reprezentēta stavoklī ar Eilera leņķiem.

Apskatīsim eksperimentu otrā testēšanas posmā (sk. att. 2.4), kur pa kreisi ar ziliem rimbulem tiek attēloti GPS paraugi, bet ar zaļo līniju - UKF filtra rezultējoša trajektorija, un pa labi ar zilo līniju tiek attēlots magnetometra

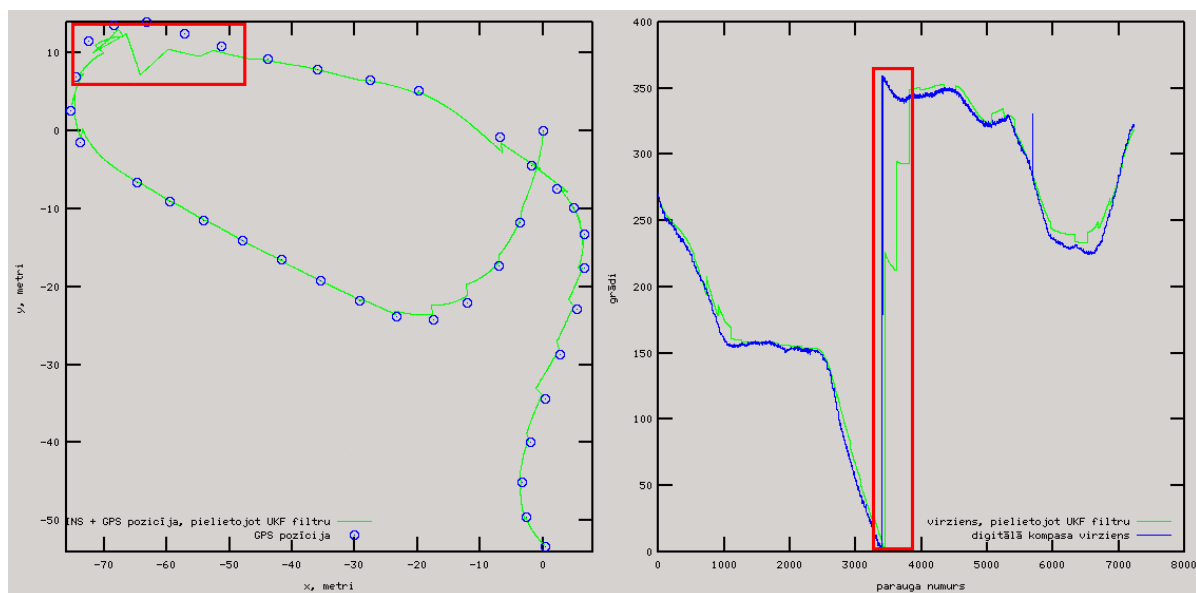


Att. 2.2: Testēšanas posmi

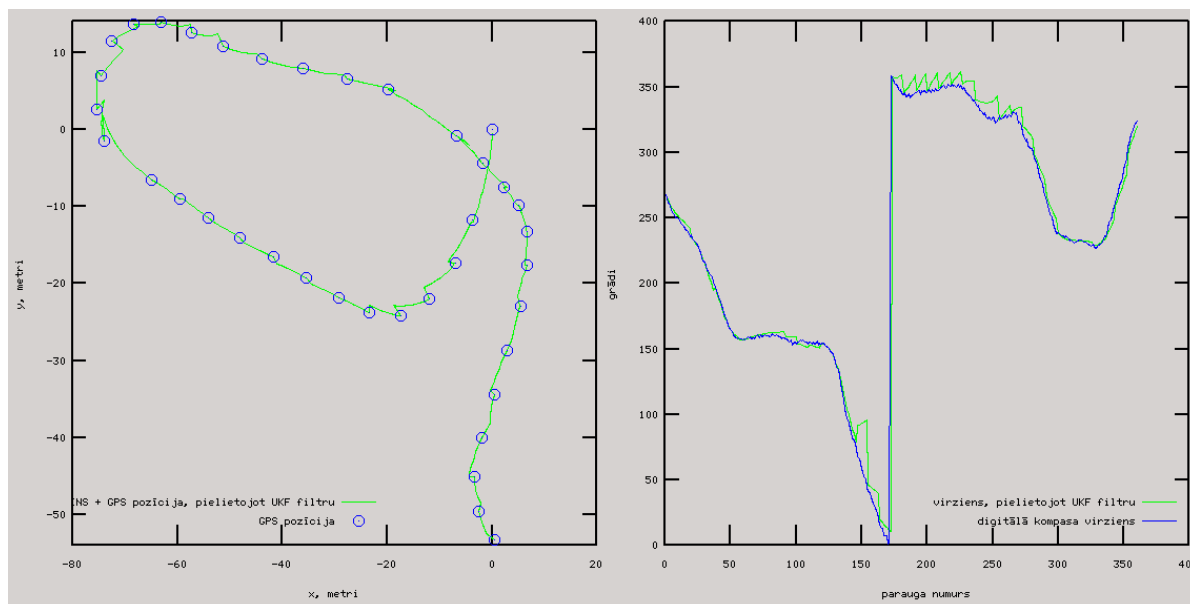


Att. 2.3: Pirmais testēšanas posms

kurss, bet ar zaļo līniju - UKF filtra rezultējošs kurss. Rēķināšanas ziņā šis eksperiments ir identisks pirmajam, tomēr šeit var redzēt posmu ar ļoti stipri izteiktu kļūdu. Šī kļūda rodas Eilera leņķu orientācijas reprezentācijas veida dēļ.



Att. 2.4: Otrais testēšanas posms - pozīcija (labajā pusē), kurss (kreisajā pusē)



Att. 2.5: Otrais testēšanas posms - pozīcija (labajā pusē), kurss (kreisajā pusē)

Apskatīsim vēl vienu eksperimentu otrā testēšanas posmā (sk. att. 2.5), kur pa kreisi ar ziliem rimbulem tiek attēloti GPS paraugi, bet ar zaļo līniju - UKF filtra rezultējoša trajektorija, un pa labi ar zilo līniju tiek attēlots magnetometra kurss, bet ar zaļo līniju - UKF filtra rezultējošs kurss. Rēķināšanas ziņā šis eksperiments ir identisks otram, tomēr šeit nav posmu ar tik pat stipri izteiktam kļūdām. Lai to panākt, orientācijas reprezentācijai stavoklī šeit tika izvēlēts kvaternionu veids.

2.5 Kopsavilkums un secinājumi

Navigācijas sistēmas pilnveidošanas laikā tika paveikts:

- Navigācijas sistēmas programatūras arhitektūras plānošana un implementācija;
- UKF (Unscented Kalman Filter) filtra implementācija octave vidē;
- Programmatūras izveide UKF filtra testēšanai ar INS un GPS simulācijas datiem octave vidē;
- Programmatūras pielāgošana darbībai ar reāliem INS un GPS datiem;
- Eksperimentu veikšana INS un GPS datu savākšanai;
- UKF filtra darbības rezultāta novērtēšana, ņemot vērā žiroskopa un akselerometra sensoru datu nobīdes;

- UKF filtra darbības rezultāta salīdzināšana Eilera leņķu un kvaternionu orientācijas reprezentācijas formas izmantošanas gadījumos;
- Lietotāja web-saskarnes pilnveidošana, sensoru datu lejuplādēšanas iespējas ieviešana.

Secinājumi:

- Eilera leņķi ir viegli uztverams un interpretējams orientācijas reprezentācijas veids, tomēr tie nav efektīvi un nav stabili lietošanā;
- Kvaternioni nav viegli iztverams, bet ir viegli interpretējams orientācijas reprezentācijas veids un tie ir ļoti efektīvi un stabili pielietošana;
- Sensoru datu sapludināšanas laikā ir vieglāk izmantot Eilera leņķus, jo tie ir pilnīgi neatkarīgi, savukārt kvaternionu pielietošana ir sarežģīta, jo kvaterniona komponentes ir cieši saistītas;
- Orientācija kā sistēmas stāvokļa sastāvdaļa tiek novērtēta ar UKF filtra palīdzību tikai 1 reizi sekundē, kas pašlaik izskatās nepietiekami;
- Programmas izveide octave vidē bija labs gājiens, kas varētu stipri palīdzēt navigācijas sistēmas koda pilnveidošanā un atklādošanā.

2.6 Nākotnes perspektīvas

Nākotnē tiek plānota navigācijas sistēmas programmatūras pilnveidošana un attīstīšana, kas dotu iespēju iegūt gatavu navigācijas risinājumu uz Sparkfun IMU moduļa un Holux M-1200E GPS uztvērēja bāzes. Lai sasniegtu labāku rezultātu ir nepieciešami:

- rūpīgāka sensoru kalibrēšana;
- orientācijas atjaunošanas frekvences palielināšana;
- eksperimentu veikšana.

Literatūra

- [1] GCDC organization. *GCDC 2011 Rules & Technology Document Final version*, April 2011.
- [2] The New York Times: Google Car. <http://www.nytimes.com/2010/10/10/science/10google.html>.
- [3] DARPA Grand Challenge. <http://www.darpa-grandchallenge.com/>.
- [4] Grand Cooperative Driving Challenge. <http://www.gcdc.net/>.

Nodaļa 3

EdiMote grupa

Anotācija

EdiMotes grupa nodarbojas ar universālas platformas izstrādi, kas būtiski atvieglo iegulto sistēmu aparatūras un programmatūras izstrādes procesu. Šāds rīks ir modulārs un nav piesaistīts pie konkrētiem iegulto sistēmu moduļiem (MCU, radio, sensoru, atmiņu, utt.) kā tas ir vairumā izstrādes rīku [4] [1] [2]. Tas nodrošina iegulto sistēmu izstrādātājam iespēju savstarpēji savienot iekārtas moduļus ar minimālu vai bez iespiedplašu projektēšanas nepieciešamības, izstrādes gaitā, kā arī veikt atsevišķu moduļu un kopējās sistēmas testēšanu. Tiek veidota arī analogo un ciparu signālu atklūdošanas un analīzes iespēja, kā arī atsevišķu moduļu enerģijas mērīšanas funkcija. Platforma tiek vadīta no datora, kā arī analizējamo datu apstrāde, attēlošana un glabāšana notiek datorā.

EdiMote [5] palīdzētu atrisināt tipiskās iegulto sistēmu projektēšanas problēmas, ar ko saskaras gan inženieri, gan pētnieki. Šīs problēmas var iedalīt ar aparatūru saistītās problēmās un ar programmatūru saistītās problēmās:

- aparatūras problēmas:
 - moduļu savstarpējā savienošana;
 - signālu komutēšana;
 - enerģijas patēriņa mērīšana;
- programmatūras problēmas:
 - dažādu mikrokontrolieru programmēšanas atbalsts;
 - ciparu un analogo signālu atklūdošana;
 - programmu veicamo operāciju ietekme uz enerģijas patēriņu.

3.1 Ievads

Iegulto sistēmu izstrāde ir laikietilpīgs un sarežģīts process. To iespējams atvieglot, izmantojot dažādus palīgrikus, kas palīdzētu savienot atsevišķos iegultās sistēmas moduļus, nodrošināt to savstarpējo signālu savietojamību,

analogo un ciparu signālu atklūdošanu, kā arī novērtēt atsevišķu moduļu un sistēmas kopumā patērējamo enerģiju konkrētu procesu veikšanas brīdī. Izstrādājot iegultās sistēmas, kas izstrādes sākumā tiek veidotas kā prototipi, jāveic moduļu vai atsevišķu integrālo mikroshēmu savienošana, kam nepieciešama laikietilpīga spiesto plašu izstrāde [3], nepieciešamas specifiskas ierīces procesoru programmēšanai un dažādi mērinstrumenti signālu novērošanai un atklūdošanai. EdiMote [5] ir universāls iegulto sistēmu izstrādes rīks, kas paredzēts visu iepriekšminēto uzdevumu atvieglošanai.

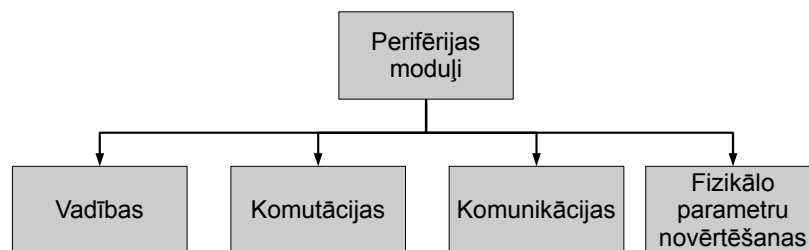
Iepriekšējā laika posmā tika izstrādāts jaunas arhitektūras koncepts. Tās pamatā ir dažāda veida moduļi - perifērijas, testa, kuri savā starpā ir savienojami ar konektoru palīdzību. Tāda arhitektūra ir mērogojama.

3.2 Risinājums

Risinājums balstās uz dažāda veida moduļiem, kurus var saspraust savā starpā, jebkurās kombinācijās. Lai šāda pieeja būtu realizējama, ir nepieciešami vismaz divi moduļu veidi - perifēriālie un testējamie moduļi. Perifēriālie moduļi kalpo prototipējamās iekārtas infrastruktūras veidošanai, piemēram, testējamās iekārtas enerģijas patēriņa noteikšanai, signālu atklūdošanai utt. Šajā pieejā atšķirībā no citām pieejām, piemēram, izmantojot izstrādes rīkus, infrastruktūras konfigurācijas sarežģītību lietotājs pats var brīvi definēt. Šāda arhitektūra ir mērogojama. Prototipējamās iekārtas sarežģītība ir aktīvā no izmantoto moduļu daudzuma.

3.2.1 Perifērijas

Perifērijas moduļi - funkcionālie mezgli, kas nodrošina veidojamā prototipa infrastruktūru. Šos moduļus iedala pēc funkcionalitātes 3.1



Att. 3.1: Perifērijas moduļu iedalījums

Prototipējamās iekārtas fizikālo parametru novērtēšana

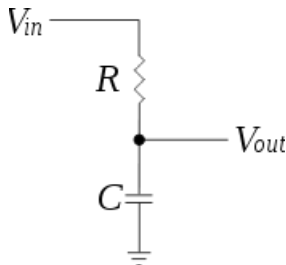
Veidojot elektroniskas iekārtas, piemēram, iegultu sistēmu, bezvadu sensoru mezglu, tām ir jānomēra dažādi fizikālie parametri. Prototipēšanas sākuma stadijā lieti noder signālu atklūdošana - analogo un ciparu. Izveidojot šādu

perifērijas moduli tiek izslēgta nepieciešamība pēc ārējiem mērinstrumentiem - osciloskopiem vai multimetriem. Šāds atklūdošanas modulis nodrošina to, ka lietotājs var atrast vai nodalīt kļūdas, piemēram, saprast vai vaina ir izstrādātajā programmatūrā, vai arī signāli ir nepareizi savienoti.

Nemot vērā to, ka iegultas sistēmas, it īpaši BSM, tiek darbinātas no baterijām, tad īpaša vērība tiek pievērsta tam, cik daudz šāda sistēma patērē enerģijas. Vairums gadījumos šo parametru ir iespējams noteikt, kad ir uzbūvēts iekārtas prototips. Šajā pieejā ir iespējams noteikt izveidotās sistēmas enerģijas patēriņu, gan kopējai testējamai iekārtai, gan katram testa moduļim atsevišķi kā arī noteikt atsevišķu programmas procesu patēriņu.

Prototipējamās iekārtas signālu komutēšana

Šajā perifērijas moduļu kategorijā ietilpst moduļi, kas nodrošina analogo un ciparu signālu komutāciju. Šīs funkcionalitātes realizēšanai var izmantot analogos - multiplexorus, slēdžus, matricas. Maksimālā caurlaižamā frekvence tiek definēta izejot no pretestības lieluma, kad signāli ir saslēgti savā starpā (turpmāk tekstā R_{on}), kā arī no pieslēgtā kondensatora kapacitātes vērtības.



Att. 3.2: Zemo frekvenču filtrs

Izveidojas zemo frekvenču filtrs (Battervorta pirmās kārtas filtrs). Izvedot pārvades rakstulīkni un ieliekot attiecīgos mainīgos - R_{on} , C ir iespējams izrēķināt, cik lielas frekvences signālus ir iespējams komutēt ar attiecīgajām integrālajām mikroshēmām IMS. Iepriekš aprakstītais risinājums der gan analogiem, gan ciparu signāliem. Tālāk tiks apskatīti varianti, kas ir pielietojami ciparu signālu komutēšanai. To nodrošina - paralēlie programmējamie loģiskie masīvi - CPLD, FPGA.

Komunikāciju moduļi

Komunikāciju moduļi nepieciešami, lai nodrošinātu prototipējamās iekārtas konfigurēšanu (signālu komutāciju) kā arī, lai nosūtītu datus uz datorus no prototipējamās iekārtas. Pagaidām ir plānots izveidot divus komunikācijas moduļus - USB, Ethernet. USB var pielietot, ja prototips ir tikai viens. Šīs komunikācijas nodrošināšanai plānots izmantot FTDI IMS - FT232, FT2232 atkarībā no pārsūtāmo datu intensitātes. FT232 pārveido USB <-> UART, savukārt FT2232 nodrošina gan virknes, gan paralēlās saskarnes.

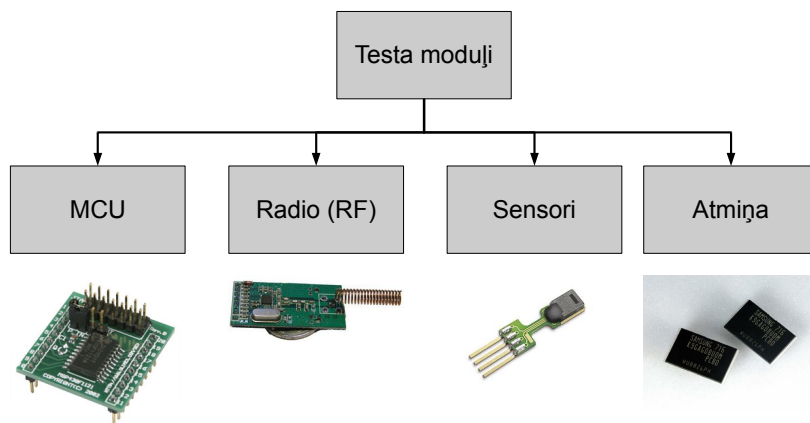
Ja ir nepieciešamība testēt vairākus prototipus, tad ērtāk ir izmantot Ethernet pieslēgvietu. Tā ir iespējams pieslēgties katram prototipam atsevišķi un iegūt sev vēlamos datus. Ir atrodami vairāki tehniskie risinājumi, kas Ethernet pārveido virknes saskarnē - UART, SPI.

Vadības moduļi

Vadības moduļiem ir paredzēts kontrolēt pārējos perifērijas moduļus. Tas ir, veikt šo moduļu konfigurēšanu, ievākt datus, kā arī eikt iegūto datu priekšapstrādi.

3.2.2 Testa moduļi

Apskatot Bezvadu Sensoru Mezglu (BSM), tad tur var izdalīt sekojošas daļas - (mikrokontrolieris) MCU, (raidītājs un/vai uztvērējs) RF, Sensori, Atmiņa (pastāvīga un/vai operatīvā). Šajā kategorijā ietilpst moduļi, kurus ir nepieciešams testēt. Tie ir sadalīti pēc veicamās funkcionalitātes, kas ir atrodama uz iegultām, bezvadu sensoru sistēmām. Tās ir - MCU modulis, Radio (RF), Atmiņas modulis, barošanas daļa, saskarne ar datoru (UART), sensori.



Att. 3.3: BSM iedalījums pa funkcionāliem moduļiem

Test MCU

Test MCU ir katras BSM galvenā sastāvdaļa, kas vada citus BSM funkcionālos blokus. Lai nodrošinātu ērtāku šādas sistēmas lietošanu ir jānodrošina vairāku mikrokonrolieru izstrādātāju, piemēram, AVR, MSP, PIC, produktus. Tas nodrošinās, ka šāda sistēma nav piesaistīta konkrētam MCU ražotājam. Tādā veidā tiks pārklāts lielāks lietotāju skaits, kas šo arhitektūru spēs izmantot. Attīstot šo domu ir jāparedz, ka ir vairāki MCU, kas atšķiras pēc veikspējas, enerģijas patēriņa. Tā ir iespējams pārklāt plašāku diapazonu un nodrošināt lietotājam plašākas, arhitektūras veidošanas, iespējas.

Sensori

Šo moduļu galvenais uzdevums ir pārvērst dabas fizikālos parametrus elektriskajā signālā - analogā vai ciparu, kuru pēc tam varētu apstrādāt, attēlot.

Atmiņas moduļi

Atmiņas moduļi paredzēti ciparu datu, oriģinālu vai apstrādātu, pagaidu vai īslaicīgai uzglabāšanai. Kuru pēc izdevības var nosūtīt uz savācējiekārtu.

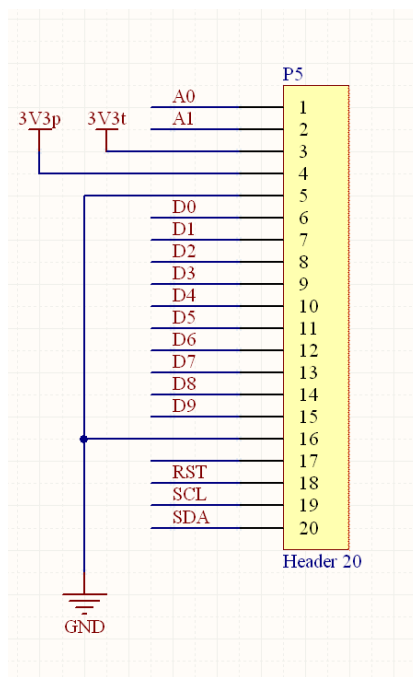
Iegūtos datus - oriģinālus vai apstrādātus, kuru pēc izdevības var nosūtīt uz savācējiekārtu.

RF

Tiek izmantots, lai iegūtos mērījumus nosūtītu uz savācējiekārtu, veidotu komunikāciju ar citiem BSM moduļiem radioviļņu pārraides ceļā.

3.2.3 Moduļu saskarne

Sekmīgai moduļu savstarpējai savienošanai ir nepieciešams izveidot standartizētu kopni, kur signālu izvietošana būtu pēc viena pamata. Pirmais, kas tika apskatīts bija barošanas spriegumi un zemes signāli. Tika izvēlēti 3V3 divi barošanas spriegumi - perifērijas un testa moduļiem. Tas darīts ar nolūku, lai būtu iespējams noteikt enerģijas patēriņu, kopējai prototipa iekārtai kā arī katram testa modulim atsevišķi. Globālai atiestatīšanai katram modulim ir kopējs RST signāls. Lai konfigurētu perifērijas iekārtas ir nepieciešams izveidot kopni, kas ļāutu to darīt. Papildus tam būtu iespējams nosūtīt datus kā arī iegūt identifikācijas ziņojumu no katra moduļa. Šim nolūkam tika izvēlēta I2C kopne, kuras realizācijai izmanto 2 signāla vadus - SDA, SCL. Pārējie signāli tiek sadalīti pēc nozīmes - ciparu signāli un analogie signāli.



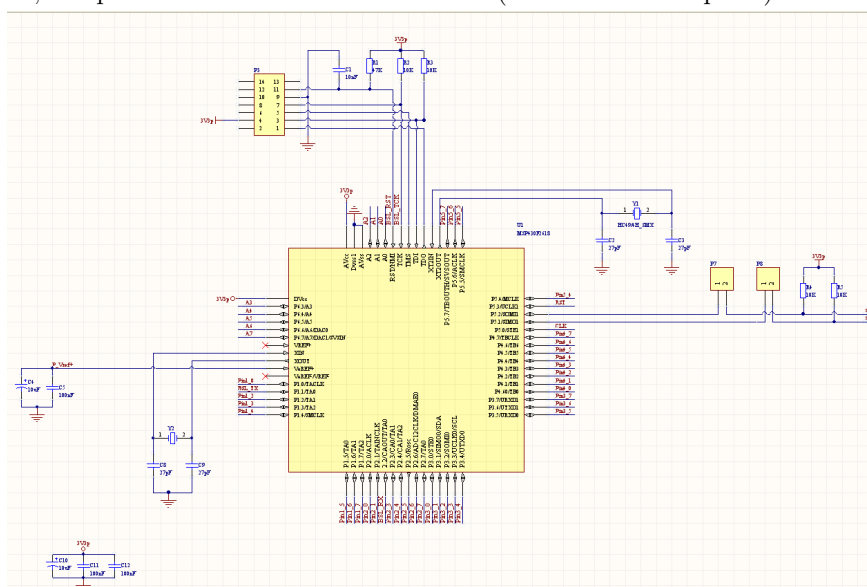
Att. 3.4: Moduļu signālu iedalījums

3.3 Arhitektūras implementācija

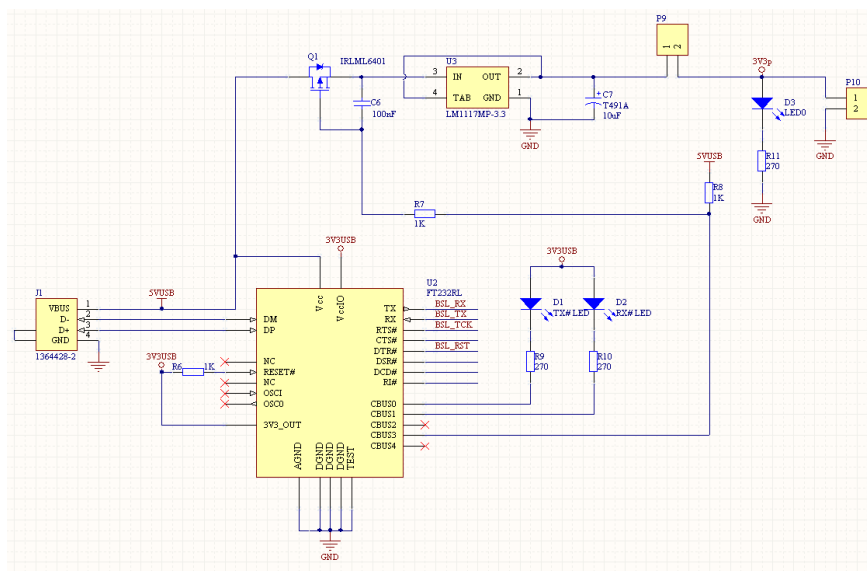
Šajā laika posmā tika uzprojektēti un izveidoti vairāki perifērijas moduļi - vadības, USB Host, ADC un DAC. Vadības un USB Host moduļi tika salodēti un ar tiem tika veikti praktiskie testi.

3.3.1 Vadības modulis

Vadības moduļim tika izvēlēts MSP430F2618 mikrokontrolieris, to var ieprogrammēt caur JTAG vai arī caur BSL. BSL tika realizēts izmantojot FT232 IMS, kas pārveido USB uz virknes saskarni (Virtuālais COM ports).



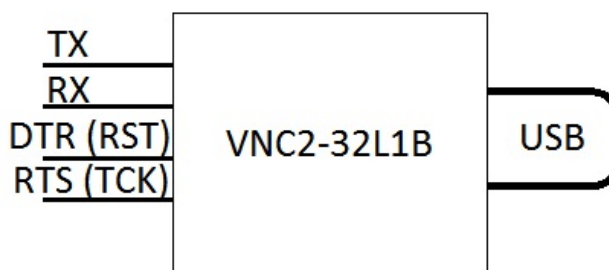
Att. 3.5: Vadības mikrokontroliera principiālā elektriskā shēma



Att. 3.6: Vadības moduļa saskarne ar datoru

3.3.2 USB Host funkcionalitātes nodrošināšana

Iespējami vairāki iekārtu savienojumu veidi jeb saskarnes moduļu savstarpējai savienošanai. Bieži iekārtas tiek savienotas, izmantojot USB saskarni, ko EdiMote tiešā veidā neatbalsta. Moduļu jeb iekārtu ar USB saskarni pievienošanai nepieciešamas speciālas mikroshēmas, kas nodrošinātu USB Host funkcionalitāti – EdiMote darbotos „master” režīmā, savukārt pievienotais modulis – „slave” režīmā. Šādā veida būtu iespējams arī EdiMotes savienot savstarpēji, t.i., uzskatīt EdiMoti par virtuālu sensoru mezglu. Šādas funkcionalitātes nodrošināšanai piemērota FTDI Vinculum II mikroshēma, kas paredzēta dažādu USB iekārtu savienošanai kā arī USB saskarnes iekārtu savietošanai ar SPI, UART, I2C saskarnēm. Šī mikroshēma nodrošina arī kontrolējamus loģiskos izvadus (GPIO). Lai nodrošinātu izvēlēto funkcionalitāti, nepieciešams veidot mikroshēmai specifisku programmatūru VinIDE izstrādes vidē. Tika izveidota testa modulis, kas sastāv no Vinculum II VNC2-32L1B mikroshēmas, četrām USB pieslēgvietām, programmatūras pieslēgvietas, barošanas sprieguma pieslēgvietas un visiem mikrokontroliera izvadiem. Šis modulis izmantots Vinculum II mikroshēmas programmas izstrādei un testēšanai. USB Host modulis paredzēts USB iekārtu, pieslēgšanai, kuras izmanto FTDI FT232 mikroshēmas USB-virknes saskarnes nodrošināšanai. EdiMotes savienojums ar testa moduli veidots izmantojot UART saskarni. Šādas pieejas priekšrocības ir savietojamība ar citu ražotāju piedāvātajiem bezvadu sensoru mezgliem ar USB saskarni, kas dod iespēju pārprogrammēt šos sensoru mezglus, kā arī veikt komunikāciju ar tiem datu iegūšanai un parametru uzstādīšanai. Tā kā Vinculum II mikroshēma neatbalsta tiešu virknes saskarnes DTR un RTS izvadu kontroli, kas ir neatņemama sastāvdaļa izmantojot BSL (bootstrap loader) programmēšanas pieeju, tās programma tika pārveidota tā, lai šie signāli mikroshēmas ieejā tiktu dublēti komandu veidā USB saskarnē. Tas panākts šos signālus pieslēdzot pie diviem mikroshēmas GPIO izvadiem.



Att. 3.7: USB Host – virknes saskarnes sistēma

3.3.3 EdiMotes pielsēgšana Ethernet tīklam

EdiMotes vadība, pārprogrammēšana un datu ieguve iepriekš tika veikta izmantojot USB saskarni, kas ierobežo lietotāju vadīt to attālināti. Šī iemesla dēļ tika meklētas iespējas pieslēgt EdiMoti Ethernet tīklam, kā papildus vadības variantu. Tika izvēlēts Lantronix Xport modulis, kas nodrošina virknes saskarnes savienošanu ar Ethernet TCP, UDP un Telnet protokoliem.

Lai nodrošinātu BSL programmēšanas funkcionalitāti un kontrolētu DTR un RTS signālus, šī moduļa kontrolei jāizmanto Telnet protokols ar RFC2217 paplašinājumu.



Att. 3.8: Lantronix Xport modulis

Lantronix Xport modulim nepieciešams nodrošināt 3.3V spriegumu. Tas panākts, izmantojot LT1117-3.3 sprieguma regulatoru, kas pieslēgts 5V barošanas spriegumam un nodrošina 800mA izejas strāvu. Kad modulis pieslēgts barošanas spriegumam, ieslēgts, nodrošinot „reset” izvadam 3.3V spriegumu, un pieslēgts Ethernet tīklam, tam jāuzstāda IP adrese. To paveic, izmantojot programmu Lantronix Device Installer un meklējot iekārtu pēc unikālas MAC adreses. Kad iekārta atrasta tīklā, tai piešķir vēlamo IP adresi. Moduļa parametrus uzstāda programmā CPR Manager, kas veic datu tuneļa izveidi starp vēlamo Ethernet protokolu un virtuālu COM portu. Šajā logā uzstāda arī virknes saskarnes parametrus – ātrumu, paritāti, „stop” bitu skaitu. Pēc parametru uzstādīšanas iegūta caur Ethernet tīklu attālināti vadāma virtuāla virknes saskarne.

3.4 Kopsavilkums

Šajā laika posmā tika paveikti šādi uzdevumi

- apskatītas jaunas arhitektūras dinamiski konfigurējamiem rīkiem;
- sākts darbs pie jaunas arhitektūras izstrādes
- Izveidotas vairākas principiālās elektriskās shēmas un iespiedplates -
 - vadības modulis
 - USB Host modulis
 - ADC DAC modulis
- Praktiskie testi ar USB Host moduli - MSP MCU un Arduino ieprogrammēšana
- Testi kontrolēt EdiMoti attālināti caur Ethernet kā arī nodrošināt iekārtu pārprogrammēšanu caur Ethernet tīklu.

3.5 Nākotnes perspektīvas

Turpināt darbu saistībā ar jaunas arhitektūras izveidi. Līdz galam notestēt esošos moduļus kā arī veidot jaunus moduļus. Izstrādāt programmatūru vadības modulim. Novērtēt, cik dažādu mikrokontrolieru ir iespējams ieprogrammēt attālināti izmantojot Ethernet.

Literatūra

- [1] P. Dutta and D. Culler. Demo: Epic: An Open Mote Platform for Application-Driven Design. In *In Proceedings of the Seventh International Conference on Information Processing in Sensor Networks (IPSN'08) Track on Sensor Platforms, Tools, and Design Methods*. IPSN, 2008.
- [2] Micaz. <http://www.cmt-gmbh.de/MICAZ.pdf>.
- [3] Spiestās elektriskās plates. http://en.wikipedia.org/wiki/Printed_circuit_board.
- [4] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceeding IPSN '05 Proceedings of the 4th international symposium on Information processing in sensor networks*. IPSN, 2005.
- [5] R. Ruskuls and S. L. EdiMote: A Flexible Sensor Node Prototyping and Profiling Tool. In *Real-World Wireless Sensor Networks 4th International Workshop, REALWSN 2010 Colombo, Sri Lanka, December 2010 Proceedings*, pages 194–197. ACM, 2010.

Nodaļa 4

MansOS grupas atskaite

Anotācija

MansOS grupas mērķis ir izstrādāt operētājsistēmu bezvadu sensoru tīkliem (BST), kā arī strādāt pie šīs operētājsistēmas praktiskajiem pielietojumiem. Lai gan jau eksistē vairākas BST specifiskas operētājsistēmas, tām ir tādi trūkumi kā krauja apmācības likne vai slikta pārnesamība. MansOS izvairās no šīm problēmām, sekojot savas arhitektūras vadlīnijām – strādāt mērķauditorijai ar UNIX un C programmēšanas pieredzi, būt lietotājdraudzīgiem, un piedāvāt vieglu pārnesamību uz jaunām aparatūras platformām.

Šajā ceturksnī MansOS pamatā noritēja darbs pie MansOS eksperimentālas novērtēšanas un salīdzināšanas ar citām BST operētājsistēmām, kuras darbojas uz Tmote Sky [5] jeb TelosB platformas: tādām kā TinyOS [4], Contiki [2] un Mantis [1].

4.1 Ievads

BST operētājsistēmu portējamību varētu atvieglot, ja to arhitektūra būtu optimāli izveidota. MansOS parāda, ka lai gan liela daļa no BST OS koda ir līdzīgi strukturēts vairākos BST OS, atsevišķus elementus iespējams uzlabot. Mūsu novērojums parāda, ka lai gan MansOS platformatkarīgā un platformneatkarīgā koda apjomi ir aptuveni vienādi, konkrētas lietotnes koda lielākā daļa ir platformneatkarīga.

Liels binārā koda apjoms noved pie tā, ka lietotņu izstrādes cikls kļūst lēns. Mēs parādam, ka lietojot MansOS, vienādas funkcionalitātes programmu uzprogrammēšana aizņem ievērojami mazāk laika nekā uz citā BST operētājsistēmām. Tāpat arī MansOS sava kompaktā koda dēļ novērš resursu trūkuma problēmas, kuras ir sevišķi jūtamas uz Contiki.

Savukārt liels pirmkoda apjoms noved pie grūtas lasāmības un tādējādi apgrūtina BST OS un BST paradigmas vispār apgūšanu. MansOS demonstrē koda apjomus, kas ir zemāki par citām BST OS, tai skaitā ievērojami zemāki par TinyOS.

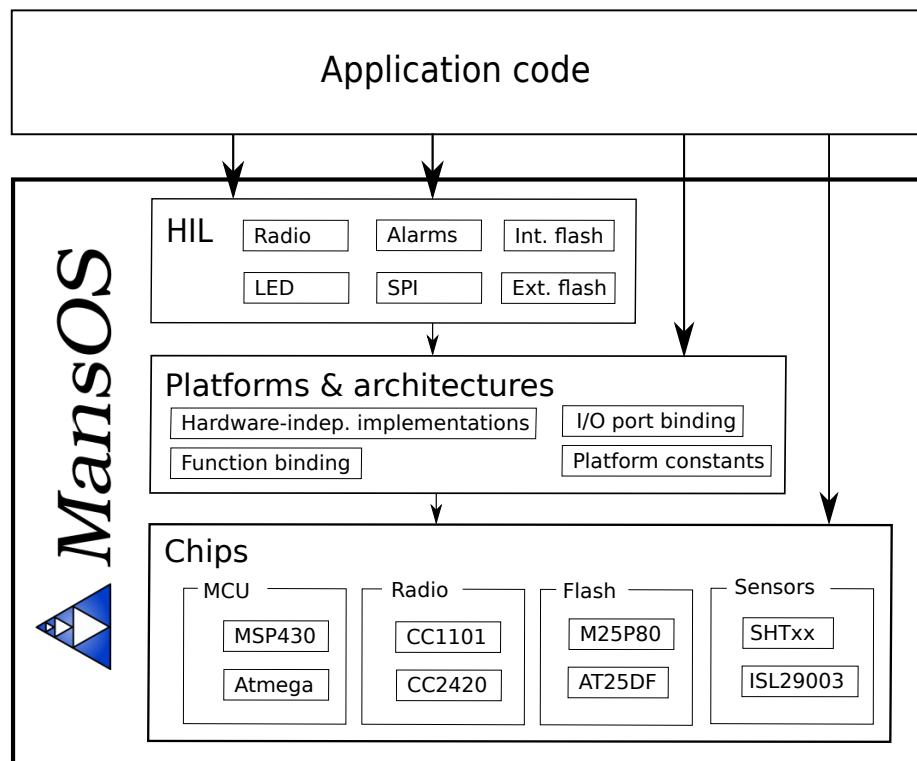
Viegla un efektīva pavedienu (threads) realizācija ir svarīga BST īpašība. Lai gan sākotnēji pavedienu atbalstīšana tika uzskatīta par ne pārāk kritiski BST OS, laika gaitā šim uzskatam ir tieksme mainīties, jo *event-based* lietotņu kods ir

grūtāk pārskatāms, bet resursu pārtēriņš pavadīdumu gadījumā nav nepārvarama problēma.

4.2 MansOS arhitektūras novērtējums

Tā kā viens no mūsu mērķiem ir portēšanas grūtību minimizācija, MansOS piedāvā modulāru arhitektūru. Čipiem specifisks kods ir nodalīts no platformām specifiska un platformneatkarīga. Dziņu kods ir platformneatkarīgs, kur vien tas ir iespējams, tāpēc vien un tas pats MansOS dzinis bieži vien var tikt lietots vairākās platformās.

Aparatūras abstrakcijas modelis MansOS ir balstīts uz novērojumu no [3]: energoefektivitātes prasību dēļ BST nav pietiekoši dot lietotājam tikai vienu, pilnībā platformneatkarīgu interfeisu. Ir jādod iespēja lietot ierīcei specifiskas aparatūras īpatnības, lai sasniegtu lielāku efektivitāti un fleksibilitāti.



Att. 4.1: MansOS komponenti un abstrakcijas slāņi

MansOS lietotājam ir pieeja visiem četriem abstrakcijas slāņiem:

- ierīcēm specifisks kods (direktorijā `chips`) – dziņi atsevišķām ierīcēm un MCU;
- arhitektūrām specifisks kods (direktorijā `arch`) – kods konkrētai arhitektūrai, kā MSP430 vai AVR;

Tabula 4.1: Pirmkoda sadalījums pa MansOS komponentiem, rindiņu skaits (neskaitot komentārus un tukšās rindīņas)

	All platforms		TelosB	
Čipiem specifisks kods	16415	34.4%	6476	20.4%
Arhitektūrām specifisks kods	5455	11.4%	1413	4.4%
Platformām specifisks kods	2435	5.1%	533	1.7%
HIL kods	8104	17.0%	8104	25.5%
Kodola kods	3540	7.4%	3540	11.1%
Tīklošanas kods	4071	8.5%	4071	12.8%
Bibliotēku kods	5211	10.9%	5211	16.4%
Menedžmenta protokola kods	2429	5.1%	2429	7.6%
Ierīcneatkarīgs kods kopā	23355	49%	23355	73.5%
Total code	47660	100%	31777	100%

- platformām specifisks kods (direktorijā `platforms`) – kods konkrētai platformai (tādai kā Arduino, TelosB vai Zolertia Z1);
- platformneatkarīgs kods, t.sk. aparatūras interfeisa līmenis (HIL), direktorija `hil`.

HIL kods sniedz vienotu aparatūras interfeisu kodolam un lietotāja programmām. Funkciju sasaiste, pinu un portu definīcijas, platformas un arhitektūras konstantes ir definētas `arch` un `platform` līmeņos. Konkrēts piemērs: radio dziņa interfeiss ir definēts HIL līmenī. Kompilācijas laikā, šis interfeiss tiek piesaistīts pie specifiskas implementācijas, kura tiek izvēlēta `platform` līmenī, kurš satur “salīmēšanas” kodu. Tādējādi Telosb platformai tiek izvēlēts CC2420 dzinis utt.

Šeit aprakstītais modelis ir līdzīgs tam, ko var atrast Contiki: `platforms` direktorija MansOS aptuveni atbilst `platforms` direktorijai Contiki, `arch` direktorija `cpu` direktorijai, `chips` direktorija `core/dev` direktorijai, un pārējais MansOS sistēmas kods (`kernel`, `hil`, un `lib`) pārējam `core` direktorijas saturam Contiki. Galvenā atšķirība starp šīm sistēmām ir labāka platformu un čipu specifiskā koda organizācija MansOS; piemēram, periodiskais taimera pārtraukumu apstrādes kods (sistēmas “sirdspuksts”) ir unificēts un kopējs visām platformām. Cita atšķirība ir funkciju sasaistes laiks. MansOS tā tiek veikta kompilācijas laikā, kas ļauj ietaupīt resursus.

Līdzīgas paraleles var tikt vilktas starp MansOS un TinyOS, lai gan pēdējai trūkst eksplīcītas arhitektūrām specifiskā koda nodalīšanas. `platforms` MansOS atbilst `platforms` TinyOS, `chips` atbilst `chips`, `hil` atbilst `interfaces`, `kernel` atbilst `system`, `lib` atbilst `lib`. Ievērojama atšķirība ir tā, ka TinyOS nav iespējams tieši piekļūt aparatūras komponentiem ar OS līdzekļiem no lietotāja programmām. Var minēt, ka šāds ierobežojums noved pie labākas koda organizācijas, bet mēs uzskatām, ka tas ir pārāk limitējošs lietotājam.

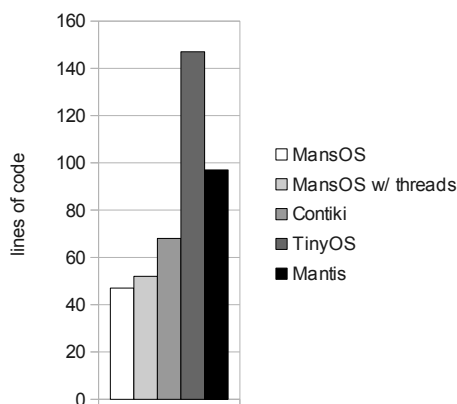
4.3 MansOS pirmkoda novērtējums

Kā parāda analīze (Tab. 4.1), aptuveni puse no visa MansOS kods ir neatkarīga no aparātūras. Tā kā aparātūras atkarīgā koda apjoms stipri variē atkarībā no tā, cik daudzas platformas tiek atbalstītas, precīzāku novērtējumu var iegūt, ja nofiksējam specifisku platformu. Kad TelosB tiek šādi izvēlēta, tikai ceturtdaļa no koda izrādās specifiska platformai vai čipi. Lielākā daļa no šī atkarīgā koda ir vienkāršs C; asamblera kods tiek lietots tika dažās, specifiskās vietās, kā pavelienu konteksta pārslēgšanā.

Novērtēšanas vajadzībām tika implementētas četras programmas: divas versijas MansOS (ar un bez paveliņiem), kā arī Contiki, TinyOS un Mantis:

- *loop* – vienkāršākā programma, kas izpilda OS inicializācijas kodu un pēc tam ieiet bezgalīgā ciklā;
- *radio tx* – periodiski sūta 100 baitu pakotnes pa radio;
- *radio rx* – pastāvīgi klausās radio pakotnes;
- *combined* – periodiski nolasa sensorus, ieslēdz/izslēdz LED, un nosūta nolasītos datus pa radio.

Slejā 1 dots pirmkods *extended combined* lietotnes implementācijā MansOS bez paveliņiem. Salīdzinot ar *combined*, implementācija ir paplašināta ar zibatmiņas izmantošanu.



Att. 4.2: Pirmkoda izmēru salīdzinājums *combined* lietotnei

Vispirms mēs salīdzinām pirmkoda izmēru *combined* lietotnei visās piecās implementācijās (Att. 4.2). Izmērs ir noteikts, izslēdzot komentārus un tukšās rindīņas. Salīdzinot ar citiem BST OS, MansOS ļauj rakstīt lietotnes ar to pašu funkcionalitāti, bet īsāku kodu. Tas ir svarīgs lietojamības ieguvums, tāpēc ka īsāku kodu ir vieglāk lasīt un apstrādāt. Tajā pašā laikā, lieli koda izmēri TinyOS signalizē potenciālu lietojamības problēmu ar šo OS.

4.4 MansOS binārā koda izmēra novērtējums

Iespējams, ka svarīgi rezultāti tiek iegūti, ja novērtējam binārā koda izmērus (Att. 4.3a). Pirmkods tika kompilēts TelosB platformai, lietojot MSP430

Listing 1 MansOS lietotnes piemērs

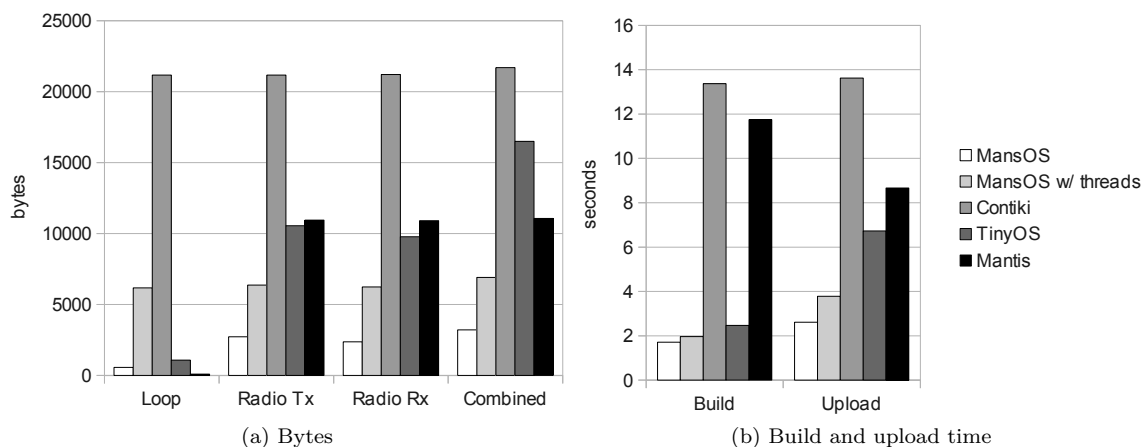
```
#include <stdmansos.h>
#include <hil/extflash.h>
// define sampling period in miliseconds
#define SAMPLING_PERIOD 5000
// declare our packet structure
struct Packet_s {
    uint16_t voltage;
    uint16_t temperature;
};
typedef struct Packet_s Packet_t;
// declare a software timer
Alarm_t timer;
// declare flash address variable
uint32_t extFlashAddress;

// Timer callback function. The main work is done here.
void onTimer(void *param) {
    Packet_t packet;
    // turn on LED
    ledOn();
    // read MCU core voltage
    packet.voltage = adcRead(ADC_INTERNAL_VOLTAGE);
    // read internal temperature
    packet.temperature = adcRead(ADC_INTERNAL_TEMPERATURE);
    // send the packet to radio
    radioSend(&packet, sizeof(packet));
    // write the packet to flash
    extFlashWrite(extFlashAddress, &packet, sizeof(packet));
    extFlashAddress += sizeof(packet);
    // reschedule our alarm timer
    alarmSchedule(&timer, SAMPLING_PERIOD);
    // turn off LED
    ledOff();
}

// Application initialization
void appMain(void) {
    // wake up external flash chip
    extFlashWake();
    // prepare space for new records to be written
    extFlashBulkErase();
    // initialize and schedule our alarm timer
    alarmInit(&timer, onTimer, NULL);
    alarmSchedule(&timer, SAMPLING_PERIOD);
}
```

GCC 4.5.3 kompilatoru. MansOS tika lietots -O optimizācijas līmenis; arī citām OS tika lietoti to optimizācijas līmeņi pēc noklusējuma.

MansOS uzrāda labākos rezultātus trijos no četriem gadījumiem. Vienīgais izņēmums ir *loop* programma: Mantis tā lieto tikai 102 baitus, salīdzinot ar 566 baitiem MansOS (bez pavedieniem).



Att. 4.3: Lietotņu binārā kods izmēru salīdzinājums, kombinētajai lietotnei

Trīs no četrām analizētajām BST OS mēģina reducēt bināros koda izmērus. MansOS: lietojot konfigurācijas mehānismu, Mantis: būvējot atsevišķus komponentus kā bibliotēkas un sasaistot tos kopā, TinyOS: topoloģiski sašķirotot visas funkcijas un izmetot nelietotās no galējā binārā faila. Tikai Contiki nepievērs uzmanību šai problēmai un uzrāda sliktākos rezultātus no visām OS.

Lielāku binārā koda izmēru TinyOS daļēji izraisa šīs OS aparatūras abstrakcijas slāņa limitācijas: tieša pieeja radio čipa dziņa kodam netiek pieļauta, un lietotājs ir spiests izmantot Active Message interfeisu.

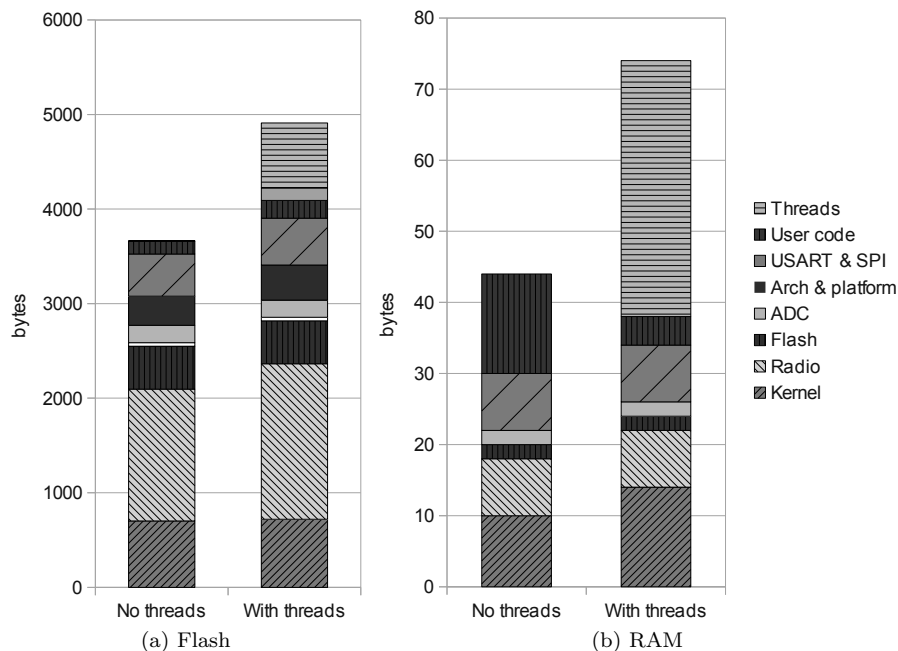
Kas attiecas uz Mantis, viņu pieeja ir efektīva, bet cieš no lietojamības problēmām. Lai uzbūvētu Mantis pēdējo versiju ar pašreizējo GNU C kompilatoru, bija nepieciešamas vairākas izmaiņas kodā. Tai skaitā, izrādījās vajadzīgs nodefinēt `putchar()` kā tukšu funkciju lietotāja kodā un aizkomentēt vairākus `mos_led_display()` lietojuma gadījumus kodola kodā (pretējā gadījumā bibliotēkām izveidojās cikliskas atkarības). Tādējādi mēs varam secināt, ka atsevišķi būvētu komponentu skaita palielināšana ir ir slihta BST lietojamībai, tā kā starp-komponentu atkarību skaits pieaug pārāk ātri.

Īsāks binārais kods nozīmē jūtamas priekšrocības BST lietotājam, Pirmkārt, pārprogrammēšanas enerģijas prasības ir tieši proporcionāla pārprogrammējamā koda izmēram (ja tiek lietota pilnā pārprogrammēšana). Lai gan visas OS atļauj kaut kāda veida daļējo pārprogrammēšanu, pilnā tomēr joprojām var būt nepieciešama atsevišķos gadījumos, kad sistēmas pamatkomponenti ir tikuši mainīti. Otrkārt, īsāks kods nozīmē īsāku izstrādes laiku, tā kā programmas uzlikšana uz motes kļūst ātrāka (Att. 4.3b). Vēl jo vairāk, uzbūvēt MansOS programmas ir ātrāk nekā to līdzinieces citos BST OS, tāpēc ka MansOS konfigurācijas mehānisms izslēdz lielu daļu no pirmkoda failiem jau pirms-kompilēšanas stadijā. Jaatzīst, ka arī TinyOS pieeja šajā ziņā ir novērojami efektīva – mūsu hipotēze par tā iemeslu ir tāda, ka visu *nesC* prekompilēšana uz vienu C failu noved pie ātrākas koda apstrādes.

MansOS tās uz notikumu apstrādi balstītajā (*event-based*) versijā izmanto ievērojami mazāk koda atmiņas vietu nekā versijā ar pavedieniem. Atšķirības pamatā ir pašas pavedienu implementācijas sarežģītība (Att. 4.4). Lai gan tā

Tabula 4.2: Resursu lietojums pa komponentiem, *paplašinātā kombinētā* lietotne, baiti

	RAM	Bez pavadieniem	Ar pavadieniem
Lietotāja kods		14	4
Kodols		10	14
Radio		8	8
USART un SPI		8	8
Zibatmiņa		2	2
ADC		2	2
Platform- un arhitektūrspecifiskais		0	0
Pavedieni		0	36
Kopā		44	74
	Flash	Bez pavadieniem	Ar pavadieniem
Radio		1394	1644
Kodols		702	720
Zibatmiņa		454	454
USART un SPI		446	496
Platform- un arhitektūrspecifiskais		308	370
ADC		182	182
Lietotāja kods		138	188
LED		38	38
Bibliotēkas		2	132
Pavedieni		0	686
Kopā		3270	4966



Att. 4.4: Resursu lietojums pa komponentiem, *paplašinātā kombinētā* lietotne

patērē vairāk resursu, versija ar pavedieniem noved pie vienkāršāka un īsāka lietotāja koda un mazāka RAM patēriņa tajā, jo lietotāja kodā ir nepieciešams saglabāt mazāk stāvokļa informācijas.

RAM lietojums ir dots neieskaitot atmiņu, kas alocēta stekiem (256 baiti uz katru pavedienu pēc noklusēšanas). Lai gan šādā veidā tiek lietots salīdzinoši liels atmiņas apjoms, tas reti kad radīs problēmas reālās pasaules lietotnēm, jo tieši koda atmiņa, nevis RAM, ir lielākais deficīts uz Tmote Sky. To demonstrē (Att. 4.4) piemēra lietotnes resursu patēriņš, tāpēc ka tā lieto proporcionāli vairāk no kopējās koda atmiņas (4966 baiti no 48 KB) nekā no kopējās RAM (74 + 256 baiti no 10 KB).

4.5 Pavedienu novērtējums

Pavedienu implementācija tiek salīdzināta ar Mantis, tāpēc ka no visām četrām apskatītajām OS tikai MansOS un Mantis iekļauj preemptīva daudzpavedienu režīma atbalstu pēc noklusēšanas.

Divas nozīmīgas īpašības MansOS pavedienu implementācijā kļūst acīmredzamas (Tab. 4.3 un 4.4)

- zemākas prasības pēc resursiem. Plānotājs, kurš tiek lietots MansOS ir vienkāršāks, piemēram, tas nenodala atsevišķas rindas pavedieniem *ready* un *sleeping* statusā. Šis kompromiss ir attaisnojams, jo BST lietotnēs parasti netiek lietots liels pavedienu skaits. Tāpēc plānotājam katrā konteksta pārslēgšanas reizē ir iespējams apskatīt visus pavedienus. Vēl jo vairāk, BST lietotnē var sagaidīt, ka pavedieni būs kooperatīvi, tāpēc plānotāja godīgums nav kritiska prasība. Visbeidzot, pavedienu

Tabula 4.3: Koda atmiņas lietojums. *PB*: plānošana pēc prioritātes, *RR*: *round-robin* plānošana.

	MansOS	Mantis
Bez pavedieniem	800	102
Viens lietotāja pavediens	1910	6368
Divi pavedieni, PB	1944	6394
Divi pavedieni, RR	1966	n/a
Trīs pavedieni, PB	1966	6408
Trīs pavedieni, RR	1988	n/a

Tabula 4.4: RAM lietojums. *PB*: plānošana pēc prioritātes, *RR*: *round-robin* plānošana.

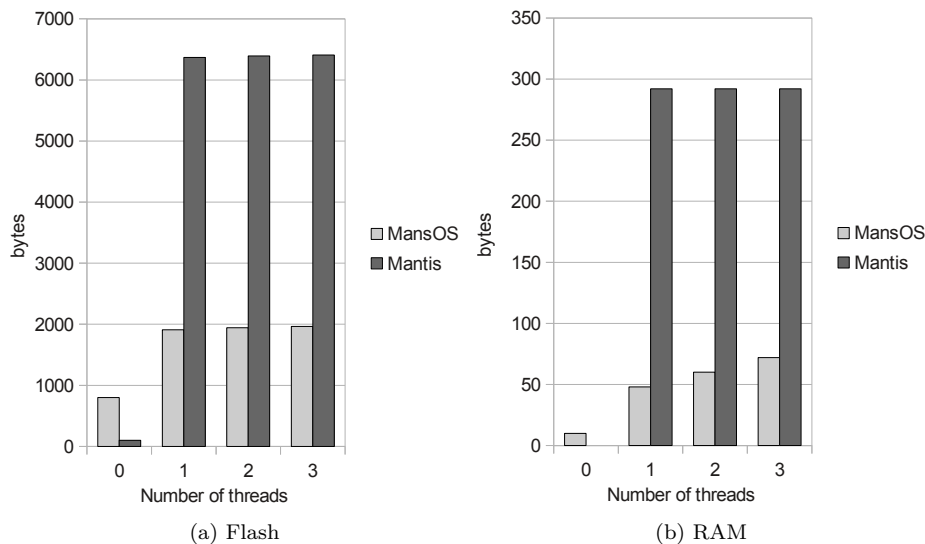
	MansOS		Mantis	
	Nesk. stekus	Iesk. stekus	Nesk. stekus	Iesk. stekus
Bez pavedieniem	10	10	0	0
Viens lietotāja pavediens	48	304	292	548
Divi pavedieni, PB	60	572	292	804
Divi pavedieni, RR	72	584	n/a	n/a
Trīs pavedieni, PB	72	840	292	1060
Trīs pavedieni, RR	88	856	n/a	n/a

hierarhija MansOS (lietotāja pavedieni nevar pārtraukt kodola pavedienu) ļauj mazināt koda slēgšanas apjomu.

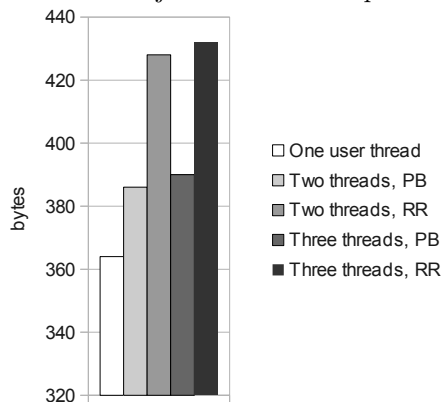
- labāka adaptācija. Mantis resursu prasības ir konstantas, MansOS prasības mainās atkarībā no pavedienu skaita.

Tehniskām detaļām – ir svarīgi piebilst, ka Mantis demo lietotnes lieto 128 baitu stekus pēc noklusēšanas. Mūsu izvēli lietot 256 baitu stekus noteica tas, ka bibliotēkas funkcijas lieto lielu steka apjomu. Piemēram, `PRINTF` makro eventuāli izsauc *libc* funkcijas. Izsaukts bez argumentiem, tas jau lieto 62 steka baitus. Tā argumenti var viegli lietot desmit vai vairāk papildus baitus. Tādējādi viens no nākotnes plāniem iekļauj sevī `printf()` funkcijas implementēšanu pašā MansOS kodā, līdzīgi kā tas ir darīts Mantis. No otras puses, Mantis alocē stekus dinamiskajā atmiņā, tāpēc tā lieto pat vairāk baitu, nekā parādīts tabulā (Tab. 4.3). Visbeidzot, jāpaskaidro, ka izmaiņas Mantis koda atmiņas patēriņā pie vairākiem pavedieniem ir tikai un vienīgi lietotāja koda izmaiņu dēļ.

Pavedienu implementācijas “sirds” MansOS ir `schedule()` funkcija, kas izvēlās, kuru pavedienu palaist tālāk. Binārā koda izmērs šai funkcijai ir atkarīgs no lietoto pavedienu skaita (Att. 4.6). *Round-robin* plānošana maksā dārgāk, daļēji tāpēc, ka tajā tiek lietotas 32-bitu *last-time-run* vērtības, salīdzinot ar 16 bitu prioritātes vērtībām, un daļēji tāpēc, ka *last-time-run* tiek atjaunots katru reizi, kad pavediens tiek palaists, kamēr prioritātes netiek mainītas visā izpildes laikā. Jebkurā gadījumā, resursu prasības var viegli apmierināt tipiska BST mote.



Att. 4.5: Resursu lietojums atkarībā no pavedienu skaita



Att. 4.6: Pavedienu resursu lietojuma salīdzinājums: koda atmiņa, kuru lieto `schedule()` funkcija.

4.6 Kopsavilkums un nākotnes perspektīvas

Kā nozīmīgu rezultātu atskaites periodā var minēt publikācijas “*Design and Implementation of MansOS: a Wireless Sensor Network Operating System*” iesūtīšanu LU zinātniskajiem rakstiem (publikācija pieņemta). Tā ir līdz šim apjomīgākā publikācija par MansOS. Publikācijas nolūkiem tika veikta MansOS eksperimentāla novērtēšana, kuras rezultāti iekļauti arī šajā atskaitē.

Vēl MansOS grupā šajā ceturksnī tika veikta:

- mūsu rezultātu prezentēšana starptautiskai auditorijai darbseminārā *Third International Workshop on Software Engineering for Sensor Network Applications* (SESENA'12);
- datu apstrādes pievienošana valodai SEAL;
- Zolertia Z1 platformas pievienošana;

- Windows instalācijas izveide ātrai un ērtai MansOS lietošanas uzsākšanai;

Literatūra

- [1] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han. MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. *Mobile Networks and Applications*, 10(4):563–579, 2005.
- [2] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. *Local Computer Networks, Annual IEEE Conference on*, 0:455–462, 2004.
- [3] V. Handziski, J. Polastre, J. Hauer, C. Sharp, A. Wolisz, and D. Culler. Flexible Hardware Abstraction for Wireless Sensor Networks. In *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN 2005)*, 2005.
- [4] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. Tinyos: An operating system for sensor networks. In *Ambient Intelligence*. Springer Verlag, 2004.
- [5] Moteiv Corporation. Tmote Sky: Low Power Wireless Sensor Module. <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>.

Nodaļa 5

SAntArray grupas atskaite

Anotācija

SAntArray grupas mērķis ir izveidot vienkāršu antenu ar elektroniski vadāmu virziendarbību. Šajā atskaitē aprakstīta **ESPAR** ģeometrisko parametru optimizācija izmantojot tas virziendarbības mērījumus.

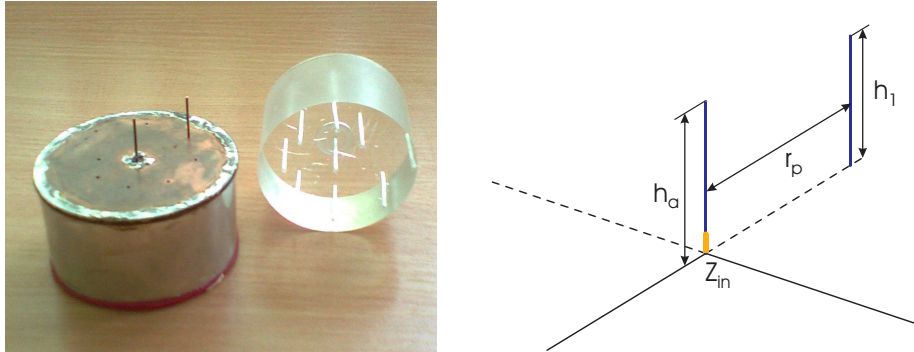
5.1 Ievads

Lai padarītu efektīvākas komunikācijai starp sensoru mezgliem, **SAntArray** grupa strādā pie Elektroniski Vadāma Parazītisko (pasīvo) Elementu Masīva (**ESPAR**) tipa virziendarbības antenas [3, 2] parametru optimizācijas. Šāda tipa antenas sastāv no viena aktīvā elementa antenas centrā, kuru ierosina no raidītāja pienakošais signāls, kā arī no vairākiem pasīvajiem elementiem, kuri noslogoti ar elektroniski vadāmas reaktīvās slodzes pretestībām.

Šajā atskaitē aprakstīti **ESPAR** antenas virziendarbības mērījumu sērijas rezultāti, kuras mērķis bija eksperimentāli noskaidrot šāda tipa antenas efektivitāti variējot tikai ar tās ģeometriskajiem izmēriem. Tas nozīmē, ka šoreiz elektroniski vadāmas reaktīvās slodzes pretestības netika lietotas – visās mērījumu sērijās, izņemot pēdējo, antenas pasīvie elementi bija elektriski savienoti ar antenas pamatni. Variēts tika ar pasīvo elementu garumiem, tādējādi veidojot ko līdzīgu Yagi antenām. Protams, šādas antenas virziendarbību pēc tās izgatavošanas var izmainīt tikai to mehāniski pagriežot, bet tas šoreiz ir pieļaujams, jo antenas tikai veidotas maksimāli vienkārši – izslēdzot analogos vadības elementus.

Antenas virziendarbības mērījumi un tās ģeometrisko parametru optimizācija tika veikta atbilstoši sekjošam plānam:

- Vispirms veic mērījums ar pašu vienkāršāko **ESPAR** antenu, kas sastāv no viena aktīvā un viena pasīvā elementa. Veic mērījumus dažādiem attālumiem starp šiem elementiem, katru reizi nosakot pasīvā elementa garuma ietekmi uz antenas virziendarbību. Mērījumu mērķis ir noteikt attālumu starp elementiem, kur pasīvā elementa garuma izmaiņas visvairāk mainītu antenas virziendarbību;



Att. 5.1: Divu elementu ESPAR testantena un tās dielektriskā cepure, kā arī antenas skice ar apzīmējumiem.

- Izgatavo un optimizē trīs elementu Yagi antenas ESPAR analogu: vienu aktīvo un divus pasīvos elementus novieto uz vienas līnijas. Attālumus starp elementiem ir iepriekšējā mērījumu solī noteiktais. Mērījumu mērķis ir atrast pasīvo elementu garumus, kuriem atbilst vislabākā antenas virziendarbība;
- Veic mērījumus ar septiņu elementu ESPAR antenu “papildinot” optimālo trīs elementu antenu ar blakus elementiem. Mērījumu mērķis ir noskaidro blakuselementu ietekmi uz antenas virziendarbību un atrast to optimālos garumus.

Papildus tika veikti mērījumi divielementu ESPAR antenai, kuras pasīvais elements pieslēgts spiestās plates līnijai. Šādas līnijas ieejas pretestība ir atkarīga no tās garuma. Tāpēc, līnijas garumu pakāpeniski samazinot, mainās antenas virziendarbība. Šāds eksperiments ļauj noteikt ESPAR antenas vadības maksimālo efektivitāti.

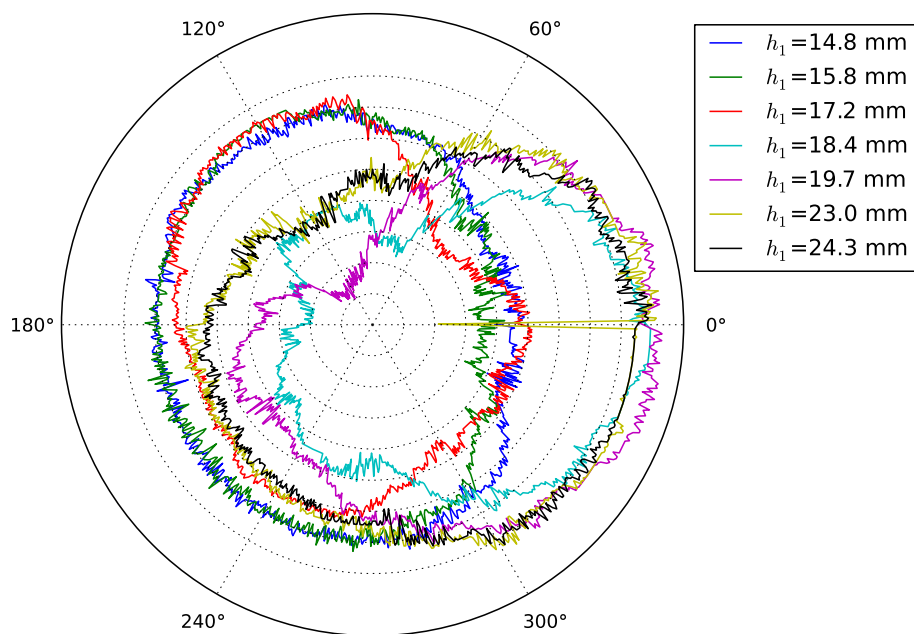
Visi eksperimenti tika veikti ar institūta rīcībā esošo antenu virziendarbības mērījumu standu [2].

5.2 Rezultāti

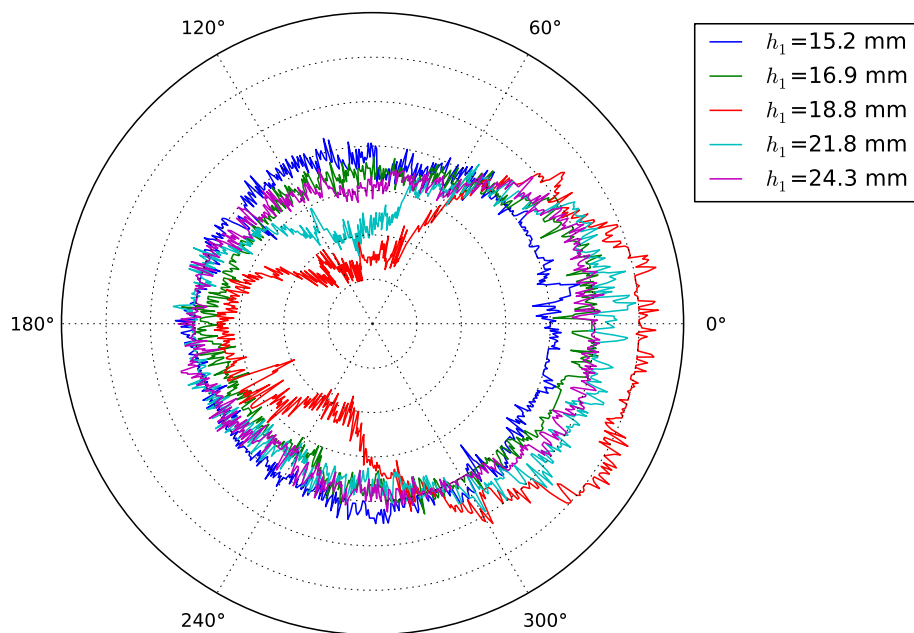
5.2.1 Divu elementu ESPAR antena

Vispirms veic mērījums ar pašu vienkāršāko ESPAR antenu, kas sastāv no viena aktīvā un viena pasīvā elementa, sk. 5.1 zīmējumu. Aktīvā elementa garums tika izvēlēts atbilstoši monopola antenas rezonansei $h_a = 18.3 \text{ mm}$. Tas atbilst apmēram $\lambda/4$ antenas dielektriskā materiāla cepurē. Attālums starp elementiem r_p mērījumu laikā tika mainīts no 14 mm līdz 28 mm ar soli 2 mm . Savukārt pasīvā elementa garums h_1 tika mainīts robežās no 14 mm līdz 25 mm . Daļa no virziendarbības mērījumu rezultātiem doti zīmējumos 5.2 un 5.3.

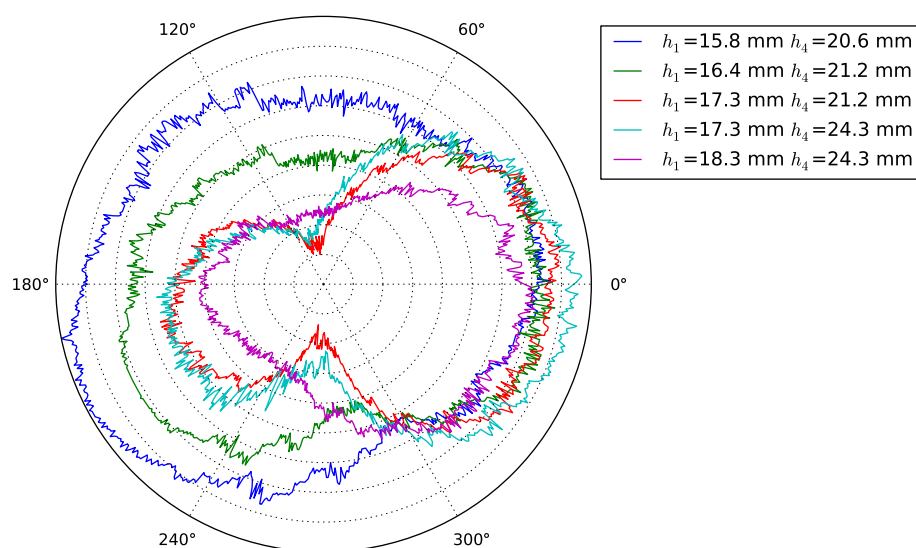
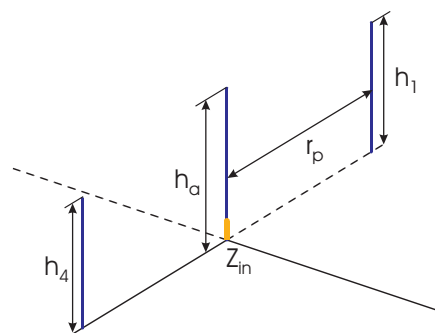
Zīmējumā 5.2 izlases veidā dotas virziendarbības diagrammas divu elementu ESPAR antenai, ja attālums starp elementiem $r_p = 14 \text{ mm}$. Vispirms varam secināt, ka palielinot pasīvā elementa garumu virs $h_1 > 23 \text{ mm}$ vai samazinot zem $h_1 < 15.8 \text{ mm}$ virziendarbības diagrammas vairs būtiski



Att. 5.2: Divu elementu ESPAR antenas virziendarbības diagramma atkarībā no h_1 . Visos mērījumos $h_a = 18.3 \text{ mm}$ un $r_p = 14 \text{ mm}$.



Att. 5.3: Divu elementu ESPAR antenas virziendarbības diagramma atkarībā no h_1 . Visos mērījumos $h_a = 18.3 \text{ mm}$ un $r_p = 24 \text{ mm}$.



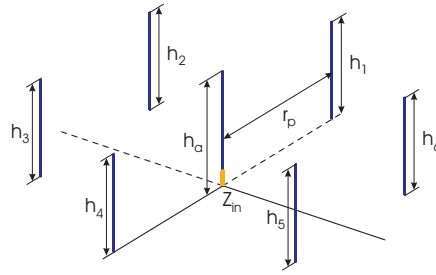
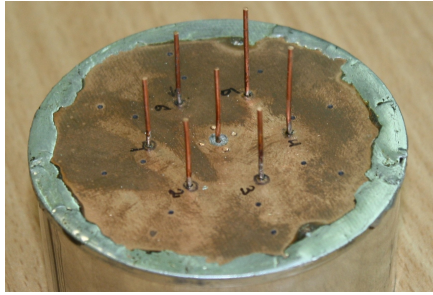
Att. 5.4: Trīs elementu ESPAR testantena, tās skice, kā arī virziendarbības mērījumu rezultāti. Visos mērījums $h_a = 18.3 \text{ mm}$ un $r_p = 14 \text{ mm}$.

nemainās. Savukārt h_1 mainoties robežas no 17.2 mm līdz no 18.4 mm antenas virziendarbība mainās no novirzīšanas uz atstarošanas. Interesanti, ka lielāku r_p gadījumā virziendarbības diagramma mainās tikai no vienmērigas uz novirzošu un tālāk atpakaļ uz vienmēriģu, kā tas redzams 5.3 zīmējumā. Šā iemesla dēļ visos tālākajos eksperimentos

$$r_p = 14 \text{ mm} \quad (5.1)$$

5.2.2 Trīs elementu ESPAR antena

Tālākiem mērījumiem tika izgatavota 3 elementu ESPAR antena tās elementus izvietojot vienā līnijā, sk. 5.4 zīmējumu. Pēc būtības, tā ir trīs elementu Yagi antena [4], kura izveidota uz elektrovadošas plāksnes. Atbilstoši iepriekšējās mērījumu sērijas rezultātiem, attālums starp aktīvo un pasīvajiem elementiem $r_p = 14 \text{ mm}$.



Att. 5.5: Septiņu elementu ESPAR testantena, kā arī tās skice ar apzīmējumiem.

Mērījumu laikā pasīvo elementu garumi h_1 un h_4 tika mainīti robežās no 14 mm līdz 25 mm . Par optimāliem tika atzītas sekojošas vērtības

$$h_1 = 17.3\text{ mm} \quad h_4 = 22.3\text{ mm} \quad (5.2)$$

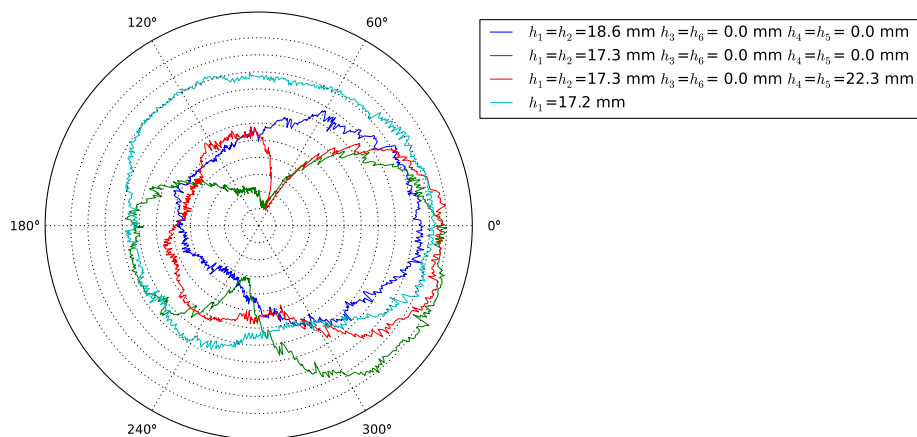
Lai ilustrētu parametru h_1 un h_4 ietekmi uz antenas virziendarbību, 5.4 zīmējumā doti dažu mērījumu rezultāti. Sāksim ar mazākajām h_1 un h_4 vērtībām. Pirmajam parametru pārim antenas virziendarbības īpašības ir ļoti vājas jo abi pasīvie elementi ir nedaudz par īsu. Tos nedaudz pagarinot, antenas virziendarbība uzlabojas, taču starpība starp pastiprinājumu virzienā uz priekšu un atpakaļ ir nepietiekoša. Palielinot h_1 līdz 17.3 mm antena sāk strādāt izteikti vienā virzienā, kas arī bija nepieciešams. Savukārt, palielinot h_1 līdz 24.3 mm , virziendarbība daudz nemainās. Ja h_1 palielina līdz 18.3 mm , tad virziendarbības diagrammā izzūd sānu minimumi.

5.2.3 Septiņu elementu ESPAR antena

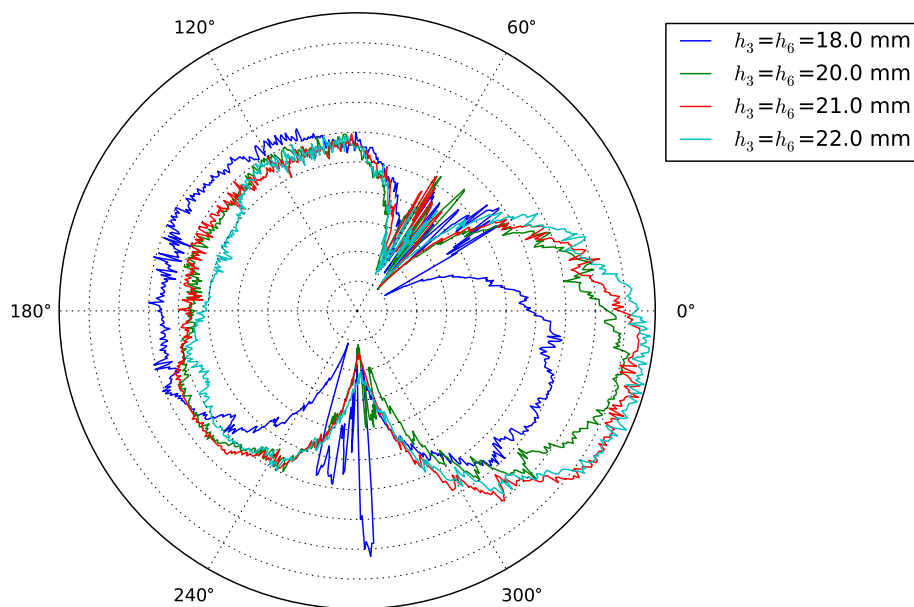
Septiņu elementu ESPAR antenu, sk. 5.5 attēlu, izmantojot simetriju, var reducēt uz divim variantiem:

- Antenas pasīvos elementus izvēlās tā, lai $h_1 = h_2$, $h_3 = h_6$ un $h_4 = h_5$. Tad antenas simetrijas plakne un tās virziendarbības maksimums ir 30° no "1" virzienā uz "2". Tātad šai konfigurācijai ir trīs ģeometriski parametri un tapēc tālāk to sauksim par 3-simetriju.
- Antenas pasīvos elementus izvēlās tā, lai $h_2 = h_6$ un $h_3 = h_5$. Tad antenas simetrijas plakne un tās virziendarbības maksimums ir elementa "1" virzienā. Savukārt šai konfigurācijai ir četri ģeometriski parametri un tapēc tālāk to sauksim par 4-simetriju.

Sāksim ar 3-simetrisku antenu. Loģiski būtu uzskatīt, ka šajā gadījumā parametru $h_1 = h_2$ un $h_4 = h_5$ vērtībām jābūt tuvām (5.2) dotajiem lielumiem. Lai par to pārliecinātos, tika veikta mērījumu sērija dažādām antenām un tās rezultāti atspoguļoti 5.6. zīmējumā. Vispirms tika veikti eksperimenti ar trīs elementu antenu, kurai ir divi blakusesoši pasīvie elementi. Kā redzams attēlā, šādai antenai ir labāka virziendarbība nekā divelementu ESPAR antenai ar $h_1 = 17.2\text{ mm}$ un $r_p = 14\text{ mm}$. Pievienojot papildus atstarojošus pasīvos



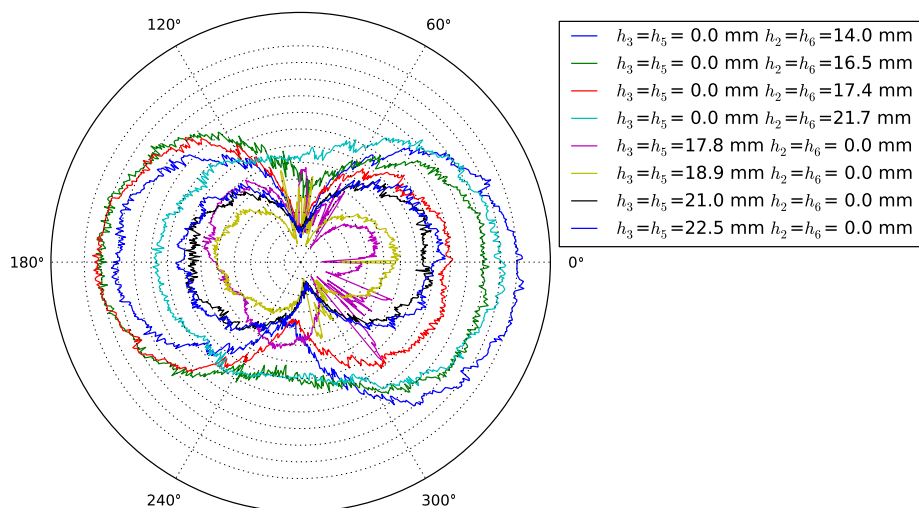
Att. 5.6: Dažādu ESPAR antenu virziendarbības diagrammas sagatavojot septiņu elementu ESPAR antenu ar 3-simetriju. Visos mērījums $h_a = 18.3\text{ mm}$ un $r_p = 14\text{ mm}$.



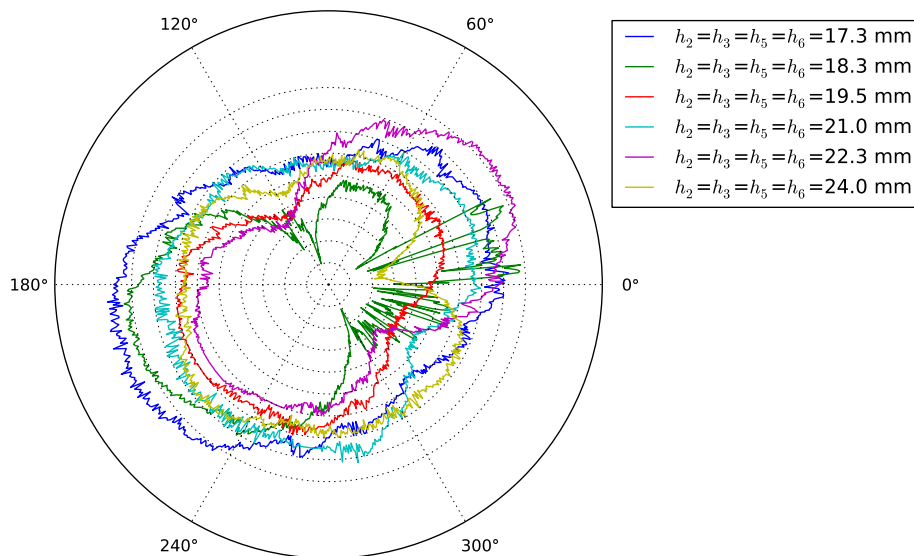
Att. 5.7: Septiņu elementu ESPAR antenas ar 3-simetriju virziendarbības diagramma atkarībā no $h_3 = h_6$. Visos mērījums $h_a = 18.3\text{ mm}$, $h_1 = h_2 = 17.3\text{ mm}$, $h_4 = h_5 = 22.3\text{ mm}$ un $r_p = 14\text{ mm}$.

elementus $h_4 = h_5 = 22.3\text{ mm}$, virziendarbības diagramma mainās, riskētu teikt, ka uzlabojas.

Septiņu elementu ESPAR antenas ar 3-simetriju virziendarbības diagramma atkarībā no $h_3 = h_6$ dota 5.7. zīmējumā. Kā redzams, šo elementu pagarināšana pasliktina virziendarbības diagrammu. Diemžēl netika veikti mērījumi ar īsākiem $h_3 = h_6$, kas varētu būt ļoti interesanti.

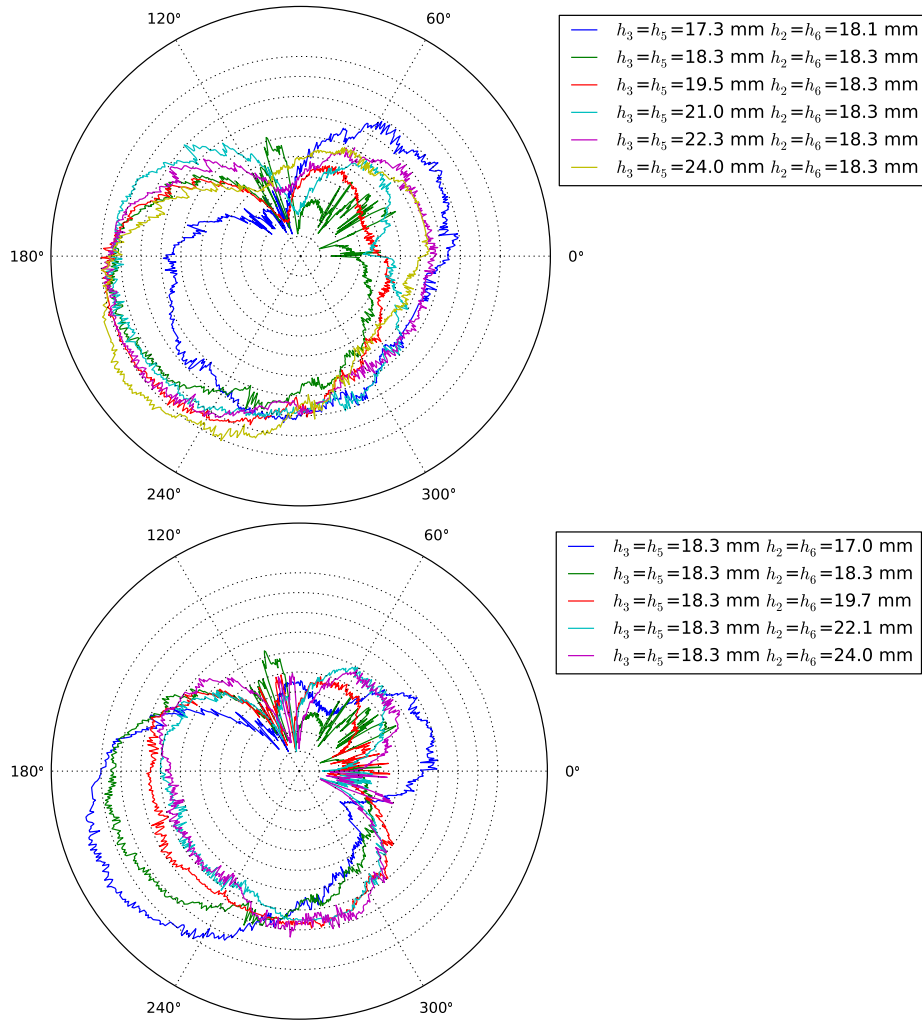


Att. 5.8: Piecu elementu ESPAR antenas skice un virziendarbības diagramma atkarībā. Visos mērījums $h_a = 18.3 \text{ mm}$, $h_1 = 17.3 \text{ mm}$, $h_4 = 22.3 \text{ mm}$ un $r_p = 14 \text{ mm}$.



Att. 5.9: Septiņu elementu ESPAR antenas virziendarbības diagrammas atkarībā no $h_1 = h_2 = h_3 = h_6$. Visos mērījumos $h_a = 18.3 \text{ mm}$, $h_4 = 22.3 \text{ mm}$ un $h_1 = 17.3 \text{ mm}$.

Līdzīgi sagatavošanās mēģinājumi tika veikti antenai ar 4-simetriju. Par sākuma konfigurāciju tiek ņemta trīs elementu ESPAR antena ar (5.2) atbilstošām vērtībām. Tā tika papildināta līdz piecu elementu antenai un tika veiktas divas mērījumu sērijas, blakuselementus novietojot pie īsākā vai garākā no elementiem, 5.8 zīmējums.



Att. 5.10: Septiņu elementu ESPAR antenas virziendarbības diagrammas. Visos mērījumos $h_a = 18.3 \text{ mm}$, $h_1 = 17.3 \text{ mm}$, $h_4 = 22.3 \text{ mm}$, un $r_p = 14 \text{ mm}$.

Vēl viena konfigurācija septiņu elementu ESPAR antenai ar 4-simetriju, kuru ir iespējams izpētīt mainot tikai vienu parametru, ir $h_1 = h_2 = h_3 = h_6$. Atbilstošās virziendarbības diagrammas dotas 5.9 zīmējumā.

Virziendarbības diagrammas, kas septiņu elementu ESPAR antenai ar 4-simetriju tika iegūtas mainot divus parametrus, uzrādīja, ka ļoti laba virziendarbība ir antenai ar $h_1 = h_2 = h_3 = h_6 = 18.3 \text{ mm}$. Tapēc 5.10 zīmējumā doti rezultāti tikai tiem mērījumiem, kur vai nu $h_1 = h_2 = 18.3 \text{ mm}$ vai arī $h_3 = h_6 = 18.3 \text{ mm}$.

5.2.4 Antena ar spiestās plates līniju

Lai pārbaudītu izvēlētas ģeometrijas spēju vadīt antenas virziendarbību, tika veikti mērījumi ar trīs elementu ESPAR antenu, kuras pasīvais elements noslogots

ar reaktīvo pretestību X_1 , sk. 5.11 zīmējumu. Mainīgu reaktīvo pretestību konstruējām izmantojot montāžās plates celiņu, kura garumu l mērījumu laikā pakāpeniski samazina. Mērījumi tika veikti divām dažādām līnijām:

Viņu pretestība	Z_0	$50\ \Omega$	$36\ \Omega$
Līnijas platums	W	2.7 mm	4.3 mm
Līnijas garums, kas atbilst $\lambda/4$	$l_{0.5}$	16.57 mm	16.15 mm

Zīmējuma 5.11 labajā pusē redzamas šīs līnijas pēc mērījumu pabeigšanas līnijai ar viļņu pretestību $50\ \Omega$ un pēc gandrīz pabeigtiem mērījumiem $36\ \Omega$ līnijai. Līnijas reaktīvo pretestību atkarībā no tās garuma var aprēķināt izmantojot formulu, [1]

$$X = -\frac{Z_0}{\tan \frac{2\pi l}{\lambda}} \quad (5.3)$$

Tas nozīmē, ka $l = l_{0.5}$ pasīvais elements ir īsslēgts. Ja $l < l_{0.5}$, tad slodze ir kapacitīva, ja $l > l_{0.5}$, tad induktīva. Mērījumu rezultātos, kas doti 5.11 zīmējumā, redzams, kā mainās virziendarbības diagramma atkarībā no līnijas garuma.

5.3 Kopsavilkums un secinājumi

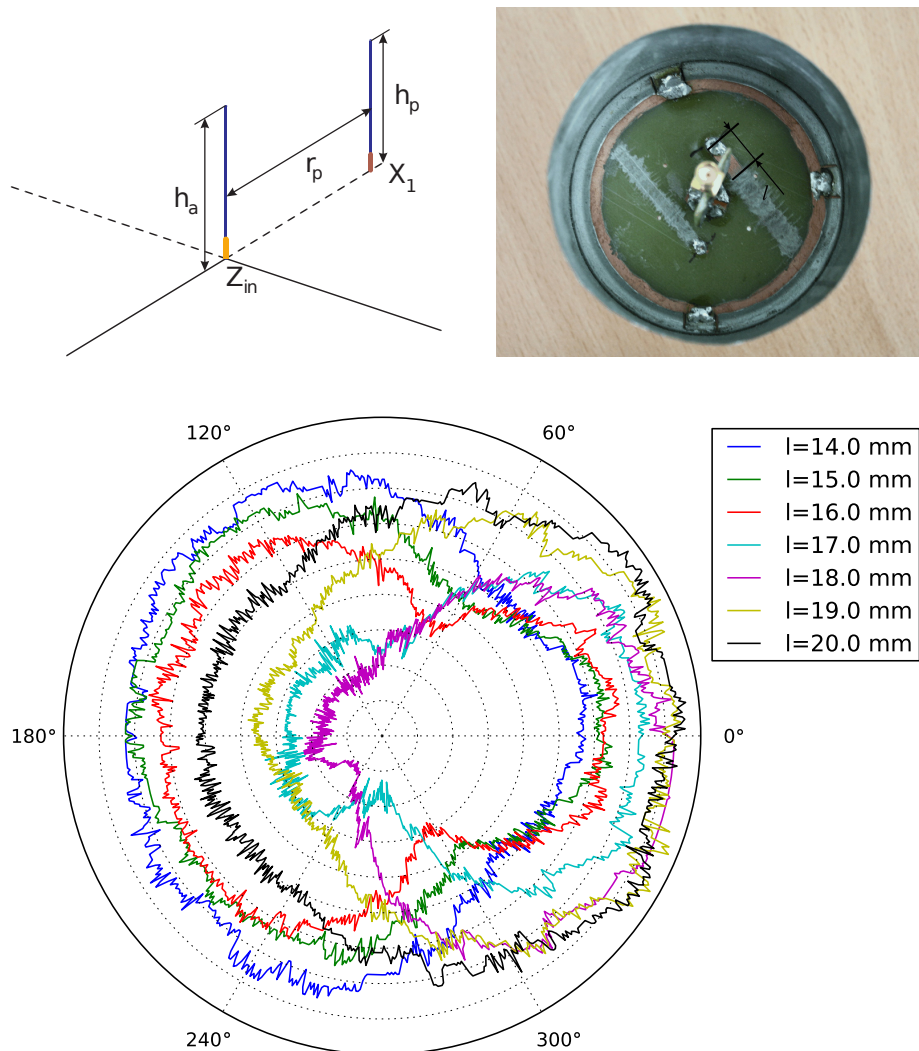
- Veicot mērījums ar pašu vienkāršāko ESPAR antenu, kas sastāv no viena aktīvā un viena pasīva elementa, tika noskaidrots ka optimāla r_p vērtība ir 14 mm;
- Tika izgatavota un optimizēta trīs elementu ESPAR antena, kur aktīvais un divi pasīvie elementi novietoti uz vienas līnijas. Attālumus starp elementiem ir iepriekšējā mērījumu solī noteiktais. Kā optimālie atzīti sekojoši pasīvo elementu garumi $h_1 = 17.3\text{ mm}$ un $h_4 = 22.3\text{ mm}$.
- Tika izgatavota un pēc tam veikti mērījumi septiņu elementu ESPAR antenai gan 3-simetrijas gan arī 4-simetrijas variantam.
- Tika veikti ilustrējoši eksperimenti ar trīs elementu ESPAR antenu, kuras pasīvais elements noslogots ar reaktīvo pretestību, kuru veido spiestās plates līnija. Iegūta kvalitatīva un kvantitatīva sakritība ar prognozēto.

5.4 Nākotnes perspektīvas

Nepieciešams salāgot iegūtās antenas ar standarta $50\ \Omega$ līniju un veikt mērījumus, lai objektīvi salīdzinātu to veiktspējas.

Literatūra

- [1] D. M. Pozar. *Microwave engineering*. J. Wiley, 2005.
- [2] K. Prieditis, I. Drikis, and L. Selavo. Passive elements array antenna for wireless sensor networks. In *ACM SenSys 2010*, Zurich, 2010.



Att. 5.11: Trīs elementu ESPAR testantena skice, pasīvā elementa slodzes reaktīvo pretestību veidojošo līniju foto kā arī antenas virziendarbības mērījumu rezultāti. $h_a = 18.3$ mm, $h_1 = 17.3$ mm un $r_p = 14$ mm.

- [3] R. Schlub, J. Lu, and T. Ohira. Seven-element ground skirt monopole espar antenna design from a genetic algorithm and the finite element method. *IEEE Trans. Antennas Propag.*, 51:3033–3039, 2003.
- [4] E. Spindler. *Antennen*. WEB Verlag Technik, Berlin, 1988.

Nodaļa 6

SenSigA grupas atskaite

Anotācija

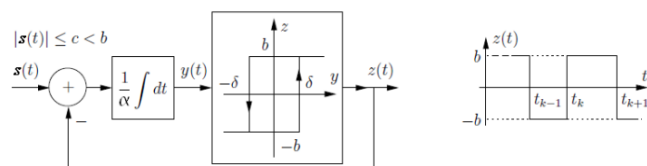
Nodaļā aprakstīts algoritms signālu atjaunošanai reālā laikā pēc to pārveidošanas ar Asinhrono Sigma-Delta Modulatoru (ASDM). Pārveidojuma rezultātā iegūtais ASDM trigeru izejas signāls tiek pārraidīts un tālāk uztverts pārveidotā signāla atjaunošanai no iegūtās diskreta laika secības.

6.1 Ievads

Asinhronais sigma-delta modulators (ASDM) analogo ieejas signālu pārveido divu līmeņu binārā izejas signālā, kura pāreju pozīcijas no viena līmeņa uz otru dod ieejas signāla atjaunošanai nepieciešamo laika informāciju [1],[2]. Atšķirībā no klasiskā analoģ-ciparu pārveidojuma ASDM gadījumā kodētas tiek nevis signāla nolašu, bet gan laika momentu vērtības, no kurām iegūstamas signāla integrālās vērtības un pēc apstrādes arī pats pārveidotais signāls. Lai šādu pieeju sekmīgi pielietotu praksē, svarīga ir signālu atjaunošana reālā laikā.

6.2 Ar ASDM pārveidotu signālu atjaunošana

ASDM blokshēma parādīta 6.1 attēlā [1]. Pārveidojuma rezultātā frekvenču



Att. 6.1: ASDM blokshēma

joslā līdz F_{max} ierobežots signāls $s(t)$ tiek attēlots ar diskreta laika secību t_k , $k \in \mathbb{Z}$, kas sakrīt ar pārveidotāja izejas signāla $z(t)$ līmeņu b un $-b$ pārslēgumu vietām. Atjaunošanas uzdevums ir no dotās secības t_k iegūt signāla $s(t)$ analogo formu.

6.2.1 Atjaunošana visā signāla garumā

Signāla $s(t)$ atjaunošanu veic, nosakot koeficientu c_n vērtības izteiksmē

$$\int_{-\infty}^t s(u)du = \sum_{n \in Z} c_n g(t - t_n), \quad (6.1)$$

kur $g(t) = 2F_{max} \text{sinc}(2\pi F_{max} t)$ un t_n sakrīt ar intervālu $[t_k, t_{k+1}]$ viduspunktiem $(t_k + t_{k+1})/2$.

No ASDM blokshēmas seko

$$\int_{t_k}^{t_{k+1}} s(u)du = (-1)^k [2\alpha\delta - b(t_{k+1} - t_k)], \quad (6.2)$$

kur vienādojuma kreiso pusi ievērojot (6.1) var izteikt

$$\int_{t_k}^{t_{k+1}} s(u)du = \sum_{n \in Z} c_n (g(t_{k+1} - t_n) - g(t_k - t_n)) \quad (6.3)$$

No (6.2) un (6.3) izriet koeficientu c_n izteiksme

$$\mathbf{c} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{P} \mathbf{q} = \mathbf{G}^+ \mathbf{P} \mathbf{q}, \quad (6.4)$$

kur kolonnas vektora $\mathbf{q}$ un matricu $\mathbf{G}$ un $\mathbf{P}$ elementi

$$q_k = (-1)^k [2\alpha\delta - b(t_{k+1} - t_k)], \quad (6.5)$$

$$g_{k,n} = \int_{t_k}^{t_{k+1}} g(u - t_n) du \quad (6.6)$$

un

$$p_{k,n} = \begin{cases} -1, & \text{jā } k \leq n \\ 0, & \text{jā } k > n \end{cases} \quad (6.7)$$

ir zināmi. Lai paātrinātu matricas $\mathbf{G}$ pseidoinversiju, funkcijas $g(t)$ vietā lieto periodisku tās aproksimāciju

$$\hat{g}(t) = \beta \sum_{m=-M}^M e^{jm \frac{2\pi F_{max}}{M} t}, \quad (6.8)$$

kur $\lim_{M \rightarrow \infty} \hat{g}(t) = g(t)$, $\beta = 2F_{max}/(2M + 1)$ un M/F_{max} sakrīt ar funkcijas atkārtosšanās periodu [3]. Rezultātā izteiksmes (6.4) vietā iegūst

$$\mathbf{b} = \beta \mathbf{A}^+ \mathbf{X} \mathbf{D} \mathbf{P} \mathbf{q}, \quad (6.9)$$

kur matricai

$$\mathbf{A} = \beta \mathbf{X} \mathbf{D} \mathbf{X}^H \quad (6.10)$$

piemīt Toeplisa un Ermita tipa matricu struktūra, kas nozīmē tās ātru pseidoinversiju, matricas $\mathbf{X}$ elementi ir

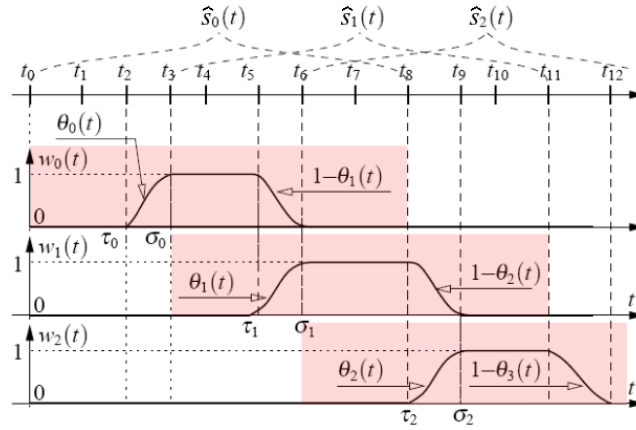
$$x_{m,k} = e^{-jm \frac{2\pi F_{max}}{M} t_k} \quad (6.11)$$

un $\mathbf{D} = \text{diag}(t_{k+1} - t_k)$. No koeficientu b_m ($m = -M, \dots, M$) vērtībām atrod signāla atjaunoto formu

$$\hat{s}(t) = \frac{j2\pi F_{max}}{M} \sum_{m=-M}^M m b_m e^{jm \frac{2\pi F_{max}}{M} t} \quad (6.12)$$

6.2.2 Atjaunošana reālā laikā

Iepriekš apskatītā signālu atjaunošana no visiem pārveidojuma rezultātā iegūtajiem laika momentiem t_k nenotiek reālā laikā, jo pirms atjaunošanas nepieciešama saņemto datu uzglabāšana līdz pārraides beigām. Reāla laika pielietojumos signālus atjauno vienlaicīgi ar datu saņemšanu, tāpēc šoreiz atjaunošanu veic mazos laika intervālos $t \in [t_{mJ}, t_{mJ+L}]$, kur $m = 0, 1, 2, \dots$ apzīmē intervāla kārtas numuru un J nosaka pārslēgumu vietu skaitu, pēc kuru saņemšanas uzsākta tiek nākamā intervāla signāla fragmenta atjaunošana [4] (6.2 attēls).



Att. 6.2: Atjaunošana intervālos, kas satur $L + 1 = 9$ pārslēgumu vietas t_k un seko ik pēc $J = 3$ saņemtajiem pārslēgumiem

Intervālu galapunktos atjaunošanas precizitāte samazinās, tāpēc atjaunotais signāls $\hat{s}_m(t)$ tiek reizināts ar intervālam atbilstošo loga funkciju

$$w_m(t) = \begin{cases} 0, & \text{ja } t \notin (\tau_m, \sigma_{m+1}], \\ \theta_m(t), & \text{ja } t \in (\tau_m, \sigma_m], \\ 1, & \text{ja } t \notin (\sigma_m, \tau_{m+1}], \\ 1 - \theta_{m+1}(t), & \text{ja } t \in (\tau_{m+1}, \sigma_{m+1}], \end{cases} \quad (6.13)$$

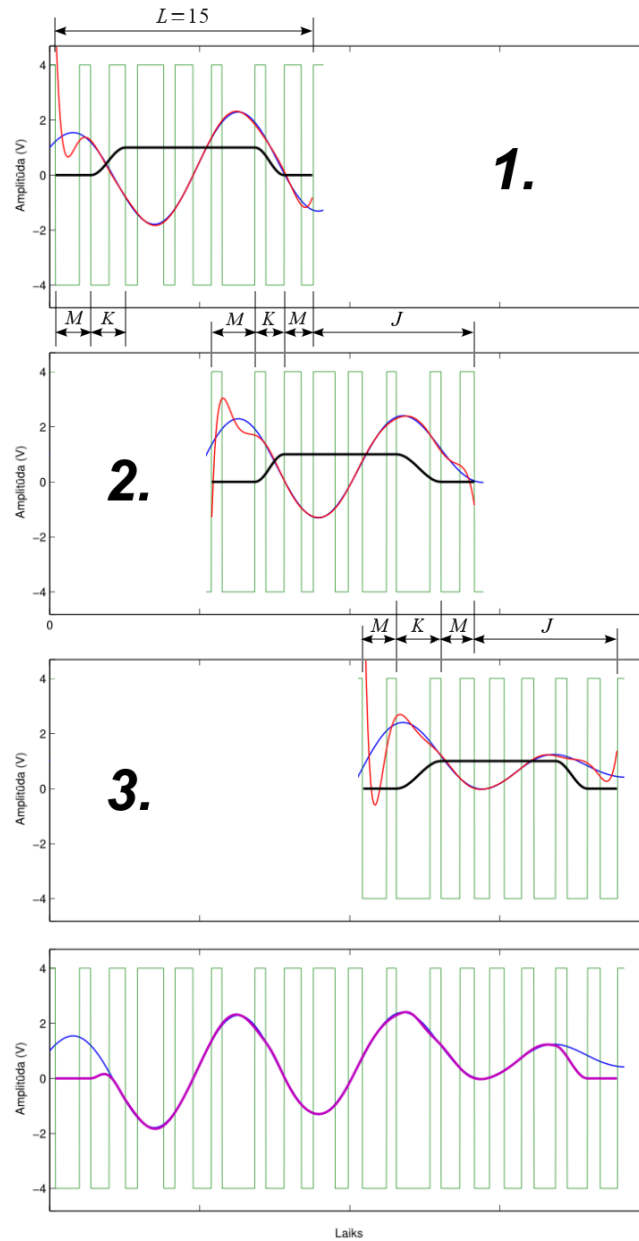
kur $\tau_m = t_{mJ+M}$, $\sigma_m = t_{mJ+M+K}$ un

$$\theta_m(t) = \sin^2 \frac{\pi(t - \tau_m)}{2(\sigma_m - \tau_m)} \quad (6.14)$$

Pēc reizināšanas signāla fragments no nulles atšķiras tikai intervāla vidusdaļā, tāpēc nākamā intervāla izvēlas pēc $J = L - 2M - K$ jaunu pārslēgumu vietu saņemšanas, tādējādi nodrošinot intervālu pārklāšanos (6.3 attēls). Saskaitot visus fragmentus kopā, iegūst atjaunoto signālu visā tā garumā

$$\hat{s}(t) = \sum_{m \in \mathbb{Z}} \hat{s}_m(t) w_m(t), \quad (6.15)$$

kuru tālāk filtrē ar zemo frekvenču filtru sākotnējā signāla frekvenču joslas atjaunošanai pirms tā reizināšanas ar loga funkciju.

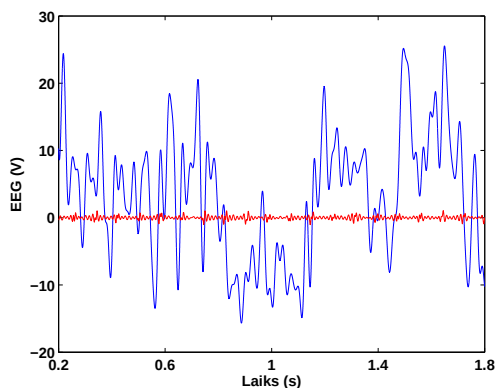


Att. 6.3: Trīs secīgiem intervāliem atbilstošo signāla fragmentu atjaunošana (zaļā līnija – ASDM trigeru izejas signāls, sarkanā līnija – atjaunotais fragments pirms reizināšanas ar loga funkciju (melnā līnija), zila līnija – sākotnējais signāls pirms pārveidošanas) un atjaunoto fragmentu summārais signāls

6.3 Rezultāti un secinājumi

Algoritma pārbaudei atjaunots tika EEG signāls, kura 1.6 sekunžu fragments parādīts 6.4 attēlā ar zilo līniju. Signāls pirms pārveidošanas tika filtrēts ar 49

Hz zemo frekvenču filtru. Maksimālais attālums starp divām secīgām ASDM trigeru pārslēgumu vietām nepārsniedza 6.7 ms, kas ir pietiekams nosacījums frekvenču joslā līdz 74 Hz ierobežotu signālu atjaunošanai.



Att. 6.4: Oriģinālais EEG (zilā līnija) un kļūdas (sarkanā līnija) signāli

Parametri tika izvēlēti $L = 20$, $M = 1$, $K = 2$ un $J = L - 2M - K = 16$, kas nozīmē, ka ik pēc 16 laika momentu t_k saņemšanas sākas jauna signāla fragmenta atjaunošana. Dotajā EEG piemērā 16 ASDM trigeru pārslēgumi tika ģenerēti vidēji 70 ms, bet signāls tika atjaunots vidēji 40 ms, tātad signāla atjaunošana reālā laikā ir iespējama.

Literatūra

- [1] A. A. Lazar, E. K. Simonyi, and L. T. Toth. Time Encoding of Bandlimited Signals, an Overview. 2005.
- [2] C. J. Kikkert and D. J. Miller. Asynchronous delta sigma modulation. In *Proceedings of the IREE Australia*, volume 36, pages 83–88, 1975.
- [3] A. A. Lazar, E. K. Simonyi, and L. T. Toth. Fast recovery algorithms for time encoded bandlimited signals. In *Proceedings of ICASSP'2005*, 2004.
- [4] A. A. Lazar, E. K. Simonyi, and L. T. Toth. A real-time algorithm for time decoding machines. In *Proceedings of EUSIPCO'2006*, September 2006.