

KLAIPĖDOS UNIVERSITETAS

Jūrų technikos fakultetas

Informatikos inžinerijos katedra

**ADAPTYVIOS INTERMODALINIŲ KONTEINERIŲ
KROVOS VALDYMO POSISTEMĖS KŪRIMAS,
TAIKANT AUTOMATINIO VAIZDO ATPAŽINIMO
ALGORITMUS**

**DEVELOPMENT OF AN ADAPTIVE INTERMODAL
CONTAINER HANDLING CONTROL SUBSYSTEM
BASED ON AUTOMATIC VISION RECOGNITION
ALGORITHMS**

Magistro baigiamasis darbas

Autorius

TMII-11 gr. Vytautas Beniušis

Vadovas

Prof. Dr. Arūnas Andziulis

Klaipėda, 2013

ANOTACIJA

Darbe nagrinėjama pramoninių procesų intelektualųjų valdymo informacinių sistemų problematika, išskiriant pagrindinius jų privalumus ir trūkumus. Pateikti srities mokslininkų atliktų tyrimų analizės rezultatai, leidžiantys nustatyti nagrinėjamos problematikos sprendimo variantus. Darbe aprašomas intelektualios intermodalinių konteinerių transportavimo krano pozicionavimo posistemės, automatinio konteinerio padėjimo taikant automatinį vaizdų atpažinimo algoritmus, modelis. Pateikti vaizdo atpažinimo bei valdymo sistemos algoritmai, kurių pagrindu sukurtas programinis kodas. Apmokintas ir įdiegtas neuroninis tinklas optimaliai valdantis elektros variklius skirtus konteinerių griebtuvo vertikaliam pozicionavimui.

PAGRINDINIAI ŽODŽIAI: intermodaliniai konteineriai, automatinis vaizdo atpažinimas, neuroninis tinklas, adaptyvi valdymo sistema.

ABSTRACT

This work focuses on the problems that arise in industrial processes being controlled by IT systems by evaluating the pros and cons of them. Analysis of a research done by scientists from similar fields enabled us to identify the possible solutions for analyzed problems.

A model of a positioning subsystem for intermodal container handling crane, applying automatic computer vision algorithms is being described in this work. Automatic computer vision algorithms were presented and based on them program code was written for the model. Neural-fuzzy network was created and installed into subsystem, so it can control container crane grabber vertical position.

KEYWORDS: Intermodal Containers, Automatic Image Recognition, Neural Network, adaptive Control System.

SUMMARY

The most important point of the container handling at the container terminal is container landing onto the platform (truck, trailer or car of the train), whereas excessive vibration is caused at that moment, this vibration and kicks causes container cargo status change. Also at that moment a loud noise is caused, which bothers tenants of the round territory about the terminal. This problem is particularly actual in Klaipeda, as the harbor area is about to residential areas and the people living around the harbor complain about the huge noise during handling procedure. So far, the owners of the port plans to build a sound-absorbing fences but this only slightly reduce that noise. The most effective way to reduce noise, is accurate positioning of the container grabber and gently landing it to the surface, causing a lot less noise than before. In other countries, this problem is less actual, because container handling terminals are usually in remote areas, where no one can hear noise caused by container handling procedure. And this is the reason why no other scientists are working on this problem.

This paper presents the container crane grabber positioning subsystem that allows the exact positioning of the container-auto-landing onto the platform, applying computer vision algorithms. Constructed subsystem consists of two separate parts: an automatic image recognition system and the adaptive control system, which is based on neural network with fuzzy interface. This network is trained by using learning algorithms so it can easily control container crane motors and adopt to changing conditions (container weight, platform height). Functional algorithms is proposed and based on them computer programs is written. Electric circuits is also created and described, that allows testing and validation of this subsystem.

Acknowledgements

The author thank the Projects LLII-061 “Development of Joint Research and Training Centre in High Technology Area” and LLIV-215 “JRTC Extension in Area of Development of Distributed Real-Time Signal Processing and Control Systems” for the possibility to complete a scientific research.

TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS

Algoritmas (angl. <i>algorithm</i>)	tam tikra veiksmų seka, kurią būtina atlikti norint pasiekti užsibrėžtą tikslą, dažniausiai prieš programuojant sudaromas, kad būtų lengviau suvokiamas procesas.
ANFIS	adaptyvi neuroninio tinklo su neraiškiosios logikos sąsaja sistema
C++	aukšto lygio tekstinė, objektinė programavimo kalba, dažniausiai naudojama kompiuterinių programų kūrimui, o taip pat gali būti naudojama ir mikrovaldiklių programavimui.
Išėjimas (angl. <i>output</i>)	tai mikrovaldiklio modulis prie kurio jungiami vykdymo įrenginiai.
Įėjimas (angl. <i>input</i>)	tai mikrovaldiklio modulis prie kurio jungiami jutikliai.
Mikroprocesorius	procesorius, realizuotas, kaip mikroschema ar mikroschemų rinkinys. Pastaruoju atveju aritmetinės logikos įrenginys gali būti vienoje mikroschemoje, o registrai – kitoje.
Mikrovaldiklis	vienkristalis prietaisas, turintis savyje visus įrenginius, būtinus minimalios konfigūracijos valdymo schemai sukurti: procesorių, duomenų atmintį, komandų atmintį, taktinių impulsų generatorių, programuojamas įvedimo ir išvedimo schemas ir t.t.
PLV	programuojamas loginis valdiklis
PWM	Signalų pločio moduliacija.
Programa (angl. <i>program</i>)	algoritmo išraiška tikslią kalbą, suprantama skaičiavimo mašinai. Tai yra programinės įrangos pavyzdys. Dauguma programų yra sudarytos iš instrukcijų sekos, nusakančios jos darbą.
Programinė įranga (angl. <i>software</i>)	kompiuterio vykdomų instrukcijų seka, skirta tam tikriems veiksams atlikti. Dažniausiai tokia įranga sukuriamą

	naudojant programavimo kalbas, o vėliau kompiliuojant ar interpretuojant kodą.
Prototipas (<i>angl. Prototype</i>)	pavyzdys, pagal kurį gaminami, kuriami kiti tos rūšies daiktai.
Sąsaja (<i>angl. interface</i>)	tam tikras susitarimas tarp dviejų programinių ar techninių įrenginių. Sąsajos yra skirtos abstrakčiai aprašyti apsikeitimą duomenimis tarp atitinkamų sistemos komponentų, kad vienam komponentui nereikėtų žinoti nieko daugiau apie kitą komponentą. Komponentu gali būti įrenginys, programos modulis, programa, klasės objektas.
Sistema (<i>angl. System</i>)	aibė susijusių elementų, ryšys tarp kurių yra glaudesnis nei tarp jų ir sistemą supančių elementų.
Techninė įranga (<i>angl. hardware</i>)	informacijos apdorojimo sistemos fizinių komponentų visuma arba tos visumos dalis.
Testavimas (<i>angl. Test</i>)	signalų rinkinys arba paprogramės, programos skaičiavimo blokų, mazgų, programinės įrangos veikimui tikrinti (kontrolinis), gedimo vietai nustatyti (diagnostinis).
Valdiklis	tai mikroprocesorius ir jam priklausančių įrenginių sistema, skirta pramonės procesų automatizavimui.

TURINYS

LENTELIŲ SĄRAŠAS	9
ILIUSTRACIJŲ SĄRAŠAS	10
ILIUSTRACIJŲ SĄRAŠAS	10
ĮVADAS	13
ĮVADAS	13
1. ANALITINĖ DALIS	15
1.1. Automatinių vaizdo atpažinimo algoritmų apžvalga.....	15
1.1.1. Konteinerių atpažinimo algoritmų analizė	15
1.1.2. Atstumo iki konteinerio matavimo algoritmų apžvalga	20
1.1.3. Vertikalių bei horizontalių viršūnių radimas	24
1.2. Konteinerių krovos krano valdymo sistemų apžvalga.....	27
1.2.1. Prisitaikanti neraiškaus tinklo valdymo sistema nuolatinės srovės variklio valdymui ...	27
1.2.2. Prisitaikanti valdymo sistema paremta skatinamuoju mokymusi	29
2. METODINĖ DALIS	31
2.1. Vaizdo atpažinimo algoritmai	31
2.1.1. Hough transformacija	31
2.1.2. Homogeniškumo operatorius, Sobelio briaunų aptikimas.....	32
2.1.3. Panašieji trikampiai	34
2.2. Valdymo sistemų metodai	35
2.2.1. Neraiškosios logikos valdymo sistemos.....	35
2.2.2. Dirbtinių neuroninių tinklų (DNT) taikymas valdymo sistemoms.....	37
2.2.3. Perceptronas, jo mokymas	37
2.3. Programinės įrangos apžvalga	38
2.3.1. NI Vision Builder	38
2.3.2. Embarcadero RAD Studio	39
2.3.3. AVRStudio 6	40
2.3.4. Konteinerių tipai ir specifinės savybės.....	42
2.3.5. Konteineriniai sunkvežimiai ir traukinių vagonai	43
2.3.6. Mikrovaldiklis, naudojamas valdymo bloke	43
3. Projektavimo dalis	48
3.1. Konteinerių krovos valdymo sistema	48
3.2. Vaizdo atpažinimo sistemos projektavimas	50
3.3. Adaptyvios konteinerių krovos valdymo posistemės programinis modelis	51

3.4. valdymo sistemos blokų architektūra	54
3.4.1. Centrinis valdymo blokas	54
3.4.2. Sistemos įtampos valdymo blokas.....	55
3.4.3. Variklių valdymo blokas	56
3.4.4. Programavimo blokas	57
3.4.5. Sąsajos su kompiuteriu blokas.....	57
4. Eksperimentinė dalis	59
4.1. Konteinerio aptikimas.....	59
4.2. Bevielės Bluetooth ryšys	62
4.3. Adaptyvios valdymo posistemės testavimas	64
IŠVADOS	67
Tobulinimo perspektyvos	67
LITERATŪRA	68
PRIEDAI	1
1. Bevielės sąsajos programinis kodas	1
2. Neuroninio-neraiškaus tinklo realizavimo ir apmokymo programinis kodas	4
3. Elektroninė programatoriaus „USB Asp“ schema	18
4. Mokslinės publikacijos	19
5. Kompaktinis diskas	44

LENTELIŲ SĄRAŠAS

Lentelė 1 Įvairių tipų konteineriai [28].....	42
Lentelė 2 „ <i>Atmega32</i> “ mikrovaldiklio parametrai.....	45

ILIUSTRACIJŲ SĄRAŠAS

1 pav. Linijos suradimas pasinaudojus Hough transformacija [1].	16
2 pav. Konteinerio skylių suradimas [1].	16
3 pav. Griebtuvo siūbavimo matavimai [6].	18
4 pav. ACCR struktūrinė schema [2].	19
5 pav. Automobilių valstybinių numerių pavyzdžiai 1 eilutės ir 2 [4].	19
6 pav. Konteinerių kodų paveikslėlių pavyzdžiai. [2].	20
7 pav. Pozicijos aptikimo sistemos vaizdas [6].	21
8 pav. Griebtuvo aukščio matavimai [6].	21
9 pav. Atstumo iki konteinerio dviem vaizdo kameromis geometrinė schema [1].	23
10 pav. Šviesos stiprumo kitimas konteinerio identifikacijos numerio zonoje [2].	25
11 pav. Konteinerio vaizdo filtravimas su horizontaliuoju aukšto dažnio filtru [2].	26
12 pav. Skirtingu paros metu fiksuojamas vaizdas [1].	27
13 pav. Prisitaikančio neuroninio-neraiškaus tinklo architektūra [16].	28
14 pav. ANFIS apmokymo metu kintantys svoriai[16].	29
15 pav. Taškai P ir Q(x,y) koordinačių erdvėje[37]	32
16 pav. Taškai P ir Q(x,y) koordinačių erdvėje[9,37]	32
17 pav. Homogeniškumo operatoriaus veikimas: a) veikimo principas; b) paveikslo fragmentas[8,37]	32
18 pav. Homogeniškumo operatoriaus II veikimas: a) veikimo principas; b) paveikslo fragmentas[37]	33
19 pav. Panašiųjų trikampių geometrinė schema.	35
20 pav. Neraiškiosios sprendimų priėmimo sistemos principinė diagrama [23]	35
21 pav. Neraiškiosios logikos įėjimo (temperatūros) funkcijos, dar kitaip vadinama fuzifikacija...	36
22 pav. Neraiškiosios išvedimo sistemos diagrama [21]	36
23 pav. Perceptronas [29].	38
24 pav. Programinio paketo „NI Vision Builder“ programos darbinis langas.	39

25 pav. Programinio paketo „Embarcadero RAD Studio“ programos darbinis langas	40
26 pav. Programinio paketo „AVR Studio 6“ programos darbinis langas.....	41
27 pav. Įvairūs konteinerių tipai [28].....	42
28 pav. Konteinerinė priekaba [30].....	43
29 pav. Konteinerinis vagonas [31].	43
30 pav. Įvairiausių mikrovaldiklių pavyzdžiai [10]	45
31 pav. Pasirinkto mikrovaldiklio išvesties įvesties kanalų reikšmės [11]	46
32 pav. Intermodalinių konteinerių krovos krano valdymo diagrama	48
33 pav. Adaptyvios konteinerių krovos krano valdymo posistemės schema.....	49
34 pav. Vaizdo atpažinimo programos struktūra	50
35 pav. Pritaikantios valdymo posistemės neuroninio-neraiškiojo tinklo struktūra.....	52
36 pav. Įvesties narystės funkcijos greičio klaidai X.....	52
37 pav. Konteinerių terminalo valdymo schemos blokinė diagrama.....	54
38 pav. Centrinio valdymo bloko elektroninė schema.....	55
39 pav. Įtampos valdymo blokas.....	56
40 pav. Variklių valdymo blokas	56
41 pav. Sąsajos su kompiuteriu blokas	57
42 pav. Sąsajos su kompiuteriu blokas	58
43 pav. Bevielės sąsajos su kompiuteriu blokas	58
44 pav. Vaizdas konvertuotas į nespaltvotą vaizdą.....	59
45 pav. Vaizdas gautas po Sobel briaunų paieškos filtravimo.....	60
46 pav. Konteinerio sienos iškilimų radimas	60
47 pav. Hough transformacija konteinerio šono linijų suradimui.....	61
48 pav. Hough transformacija konteinerio pagrindo suradimui	62
49 pav. Bevielės sąsajos testavimo NI Vision builder aplinkoje algoritmas	62
50 pav. Bevielės sąsajos testavimo NI Vision builded aplinkoje grafinis programos kodas.....	63
51 pav. Duomenų apsikeitimo tarp valdymo posistemės ir kompiuterio laiko grafikas	63

52 pav. Bluetooth sąsajos blokas	64
53 pav. Centrinis valdymo blokas	65
54 pav. Neuroninio-neraiškaus tinklo apmokymo rezultatų grafikas	65
55 pav. Neuroninio-neraiškaus tinklo prognozavimo paklaidos	66

IVADAS

Sparčiai vystantis informacinėms technologijoms, atitinkamai plečiasi ir jų taikymo spektras. Informacinės sistemos diegiamos transporto, logistikos, įvairiose pramonės srityse turi užtikrinti efektyvų valdymą, mažesnius broko kiekius, tikslesnius atliekamus matavimus bei gaunamus rezultatus ir pan. Kadangi mūsų šalis – tranzitinė, ypatingai daug dėmesio turėtų būti skiriama ne tik standartinių, bet ir specialių – intermodalinių konteinerių automatizuoto krovimo į traukinius/laivus terminalų sistemų valdymui ir monitoringui [1,2,6,45]. Silpnai išvystyta infrastruktūra neleidžia atlikti pilno konteinerių krovos automatizavimo [1,2], bei konteineriuose gabenamų produktų, prekių būsenos bei valdymo sistemos monitoringo [46].

Kaip rodo statistika, per paskutinius 10 metų intermodalinių konteinerių paklausa įvairių krovinių transportavimui visame pasaulyje stipriai išaugo [2,47]. Taip pat padidėjo gabenamų krovinių kiekis ir spektras [51], dėl to atsirado būtinybė stebėti ne tik krovinių judėjimą, bet pašalinius poveikius įtakančius gabenamą krovinį, tokius kaip: drėgmė, temperatūra, vibracija ir pan. Todėl konteinerių krovos terminalų technologinių procesų valdymui yra keliami vis didesni reikalavimai [48]. Vienas iš svarbiausių sektorių, kuriame turėtų būti atliekama ypatinga kontrolė, tai krovos darbus atliekančio krano pozicionavimas [2,49], kurio netikslus valdymas gali pažeisti krovinį pagal anksčiau jau minėtus pašalinius poveikius.

Svarbiausias momentas konteinerių krovos metu yra konteinerio padėjimas ant platformos(sunkvežimio priekabos ar traukinio vagono), kadangi šiuo metu yra sukeliamas didelė vibracija dėl smūgių, kas veikia konteinerio viduje esančio krovinio būklės pakitimą [1]. Taip pat šiuo momentu yra sukeliamas didelis triukšmas. Ši problema yra ypač aktuali Klaipėdoje, kadangi uosto teritorija ribojasi su gyvenamosiomis zonomis ir aplink uosta gyvenantys žmonės skundžiasi dėl didžiulio triukšmo krovos metu. Kol kas uosto savininkai planuoja statyti garsą sugeriančias tvoras tačiau tai tik truputį sumažins keliamą triukšmą. Efektyviausia krovos metu sukeliamą triukšmą mažinti tiksliau pozicionuojant konteinerio griebtuvą konteinerio padėjimo metu ir švelniai padėti konteinerį sukeliant daug mažesnę triukšmą. Kitose pasaulio šalyse ši problema yra mažesnė kadangi krovos terminalai dažniausiai randasi atokesnėse vietovėse, kur konteineriu krovos metu sukeliamo triukšmo niekas negirdi.

Mokslininkai atlieka tyrimus susijusius su konteinerių krovos proceso valdymu, taikydami įvairius atstumo jutiklius [38]. Taip pat tokių sistemų kūrimui pradedama naudoti automatinį vaizdų atpažinimą [1]. Apjungus tokią sistemą su intelektualiomis technologijomis būtų galima sumažinti krovos metu sukeliamą triukšmą.

Darbo tikslas – sukurti adaptyvią konteinerinio krano pozicionavimo posistemę taikant automatinio vaizdo atpažinimo algoritmus.

Tiksliui pasiekti keliama **uždaviniai**:

- 1) Atlikti išsamią nagrinėjamos srities literatūros analizę, nustatyti problemines sritis ir galimus sprendimo variantus.
- 2) Suprojektuoti ir realizuoti adaptyvią konteinerių krovos krano valdymo posistemę, taikant tinkamiausius algoritmus.
- 3) Atlikti posistemės validavimą bei testavimą, sukuriant tam reikalingus algoritmus.

1. ANALITINĖ DALIS

Šiame skyriuje apžvelgsime su vaizdo atpažinimo sistemomis ir jų algoritmais atliekamus tyrimus, jų svarbą ir naudą pramonėje. Taip pat aptariamos egzistuojančios valdymo sistemos skirtos konteinerių terminalo valdymui. Išanalizuotos variklių valdymo sistemos bei mokslininkų atliekami tyrimai.

1.1. Automatinių vaizdo atpažinimo algoritmų apžvalga

Beveik visuose pasaulio konteinerių terminaluose konteinerių kranai yra valdomi rankiniu būdu. Tokiu būdu atsiranda žmogiškasis faktorius, galimos klaidos dėl darbuotojų nuovargiu, atidumo trūkumo ar aplaidumo. Šiuolaikinės technologijos tokios, kaip vaizdo kameros bei procesoriai galintys apdoroti vaizdo kamerų fiksuojamus duomenis ir išduoti atitinkamus signalus, leidžia sumažinti žmogiškąjį faktorių ir automatizuoti tokias sistemas, kaip konteinerių krovos krano valdymas. Automatiškai valdant konteinerių krovos kraną yra svarbu tiksliai bei greitai aptikti norimą perkelti konteinerį [2,6,1].

1.1.1. Konteinerių atpažinimo algoritmų analizė

Padėto konteinerio atpažinimas: Mokslininkas Hee-Joo Yoon kartu su Young-Shul Hwang pasinaudodami vaizdo atpažinimo sistema *Stereo Vision*, sukūrė konteinerių aptikimo ir pagriebimo sistemą. Ši sistema padeda automatizuoti dalį krovos proceso mažinant žmogiškąjį faktorių ir pagreitina bei optimizuoja krovos procesą[1, 12].

Konteinerio aptikimui bendrame vaizde yra naudojamos žinomos nekintančios faktai [34]:

Konteinerio viršus yra stačiakampio formos;

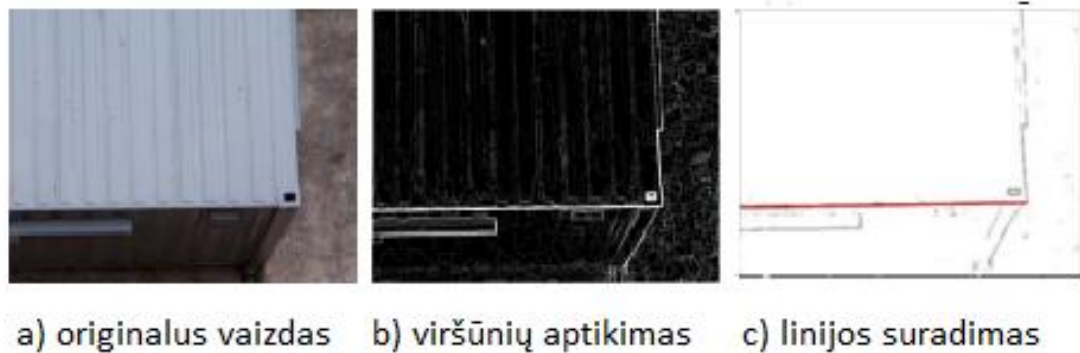
Konteinerio paviršiaus danga yra profilinė, t.y. visos konteinerio sienos turi pasikartojančias lygiagrečias linijas. kurios yra aptinkamos naudojant Hough transformaciją;

Konteinerio viršuje viršuje ir apačioje yra kvadratinės skylės konteinerio paėmimui bei fiksavimui ant platformos.

Lygiagrečių tiesių linijų aptikimui yra naudojama **Hough transformacija** [7, 9, 12]. Kiekviena aptikta linija xy – plokštumoje gali būti aprašoma:

$$x \cos \theta + y \sin \theta = \rho \quad (1)$$

čia, θ – Hauth transformacijos filtro linijos kampas su x ašimi, ρ – artimiausias atstumas iki surastos linijos.



1 pav. Linijos suradimas pasinaudojus Hough transformacija [1].

1 paveiksle pavaizduotas konteinerio krašto linijos suradimo procesas pasinaudojant Hough transformacija. 1 a – pavaizduotas originalus vaizdas užfiksuotas vienos vaizdo kameros. 1 b – pavaizduotas vaizdas po viršūnių arba kontūrų aptikimo pasinaudojant Sobel briaunų radimo operatorių [8]. Šiame vaizde gaunama daug briaunų, tarp jų ir konteinerio ribas, kurios ir yra reikalingos šiai sistemai. 1 c – pavaizduota raudona linija po Hough transformacijos, kuri yra viršutinė konteinerio sienos išilginė riba.



2 pav. Konteinerio skylių suradimas [1].

Dar vienas visų konteinerių išvaizdos specifinis bruožas yra skylės visuose keturiuose viršaus sienos kampuose. Bendrame vaizde visos skylės yra tamsesnės spalvos nei aplinka, dėl mažiau šviesos gaunančio paviršiaus. Todėl skylės galima aptikti naudojant intensyvumo ribą, pašalinant visus taškus esančius žemiau nustatytos ribos. Ši riba yra pasirenkama apskaičiavus vidutini vaizdo intensyvumą. 2 paveiksle matomas vaizdo rezultatas pašalinus vaizdo taškus su mažesniu intensyvumu nei nustatytoji riba.

Suradę konteinerio ribas bei aptikę keturias kampuose esančias skylės, bei išmatavę konteinerio aukščio bei pločio santykį autoriai priskiria vaizdą konteinerio klasei [10]. Tokiu būdu

yra surandamas konteineris ir jo regionas. Visi minėti veiksmai yra atliekami su dviem kameromis analogiškai. Konteinerio aptikimui abiejų kamerų aptikta teritorija turi atitikti visus konteinerio bruožus ir priskirta konteinerio klasei.

Padėto konteinerio ir griebtuvo atpažinimas: Analogiškai kaip ir padėto konteinerio atpažinimas yra svarbu atpažinti ir pakeltą konteinerį su griebtuvu. Mokslininkai, kaip H.Kawai ir Y.Choi pagriebto konteinerio pozicijos aptikimui ir griebtuvo tikslesniam pozicionavimui naudoja optinį vaizdo jutiklį, kuris padeda tiksliau nustatyti konteinerio poziciją, pasisukimo kampą, siūbavimą bei atstumą iki konteinerio vežimėlio [6].

Norint sumažinti konteinerio ir griebtuvo svyravimus, pirmiausia reikia nustatyti realia griebtuvo poziciją. Procesorius apdoroja iš vaizdo kameros gautus duomenis, ir nustato griebtuvo padėties pokytį realiu laiku. 3 paveikslo kairiame vaizde punktyru pavaizduota standartinė konteinerio pozicija kurioje jis būtų idealiomis sąlygomis be jokių svyravimų. O_s koordinatė yra idealios konteinerio pozicijos centras. $L1$ ir $L2$ ant griebtuvo viršaus esantys vaizdo žymenų centrai. D tai tikrasis atstumas tarp vaizdo žymenų centrų, o d vaizdo jutiklio užfiksuotas atstumas tarp žymenų centrų. $X1, X2, Y1, Y2$ atitinkamai žymenų koordinatės nuo O_s pagal vaizdo jutiklio užfiksuotą vaizdą. X_s ir Y_s tikrasis konteinerio centro pokytis. Žinant šias pastarąsias koordinates galima apskaičiuoti realių konteinerio pozicijos pasikeitimą.

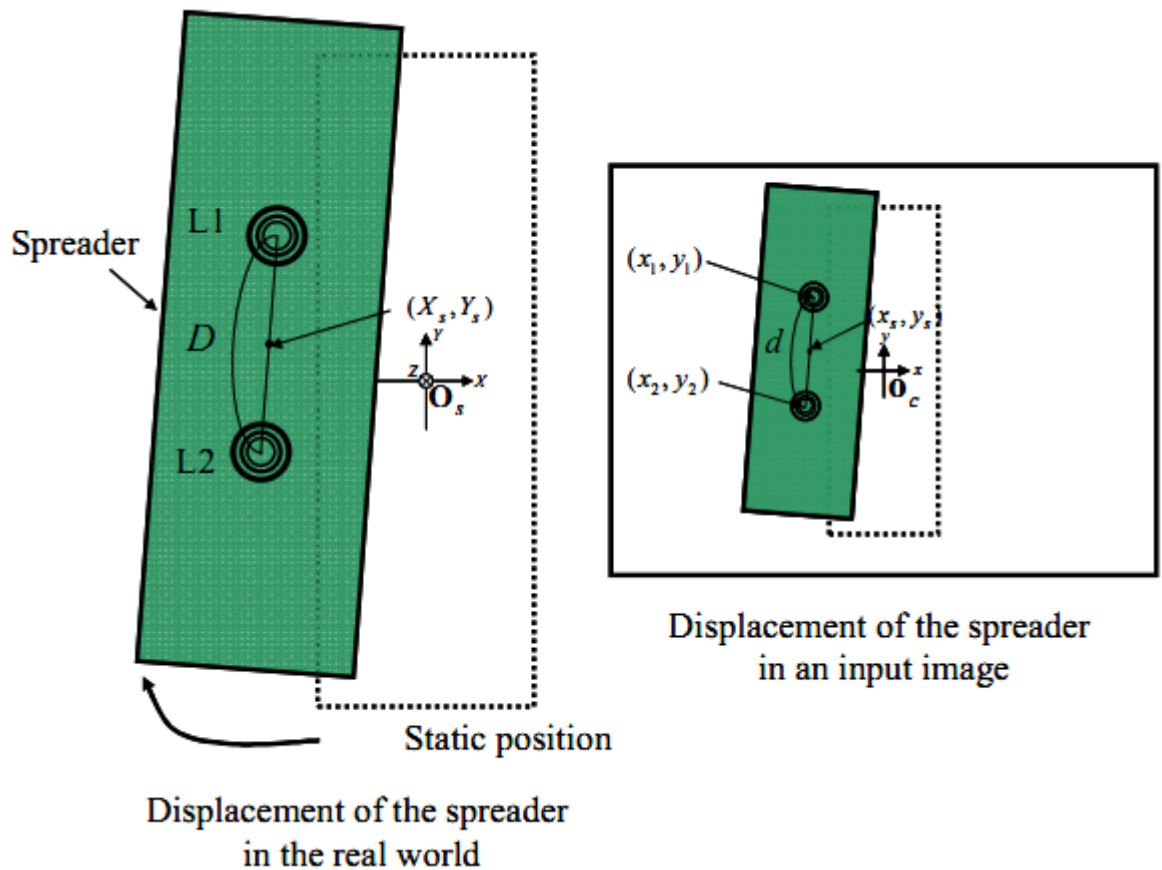
Pozicijos pasikeitimas skaičiuojamas lyginant tikrąjį atstumą D tarp žymenų centrų $L1$ ir $L2$ su vaizdo jutiklio realiu laiku išmatuotu atstumu d . Šio dydžio apskaičiavimui naudojamos Pitagoro teoremą pavaizduotą 2 formulę. Pagal 3 formulę apskaičiuojame konteinerio centro pokytį pagal vaizdo jutiklio duomenis.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

$$(x_s, y_s) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \quad (3)$$

Turėdami šiuos duomenis galime apskaičiuoti tikrąjį konteinerio griebtuvo centro pokytį:

$$(X_s, Y_s) = \left(\frac{D}{d} x_s, \frac{D}{d} y_s \right) \quad (4)$$



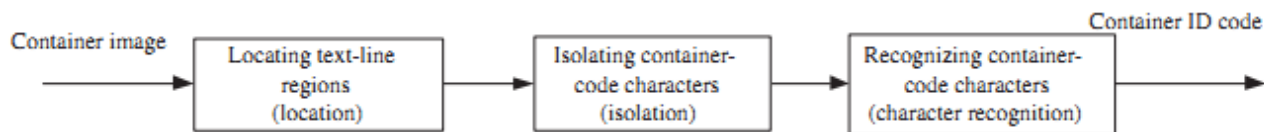
3 pav. Griebtuvo siūbavimo matavimai [6].

Konteinerio identifikacijos numerio atpažinimas: Kadangi transportuojamų konteinerių kiekiai nuolat didėja konteinerio identifikacijos numerio atpažinimas tapo esminis aspektas konteinerių paskirstyme. Mokslininkai Wei Wu su Zheng Liu nagrinėja konteinerių identifikavimo numerių atpažinimo sistemą bei su vaizdų atpažinimu išskylančias problemas [2]. Autorių teigimu tradiciniai identifikacijos numerio atpažinimo metodai veikia lėtai ir atpažinimo metu daro daug klaidų, todėl sukūrė automatinio konteinerių identifikacijos numerių atpažinimo sistemą (ACCR(4 pav)). Lyginant su kitais metodais tokiais, kaip bevieliai sensoriniai tinklai arba radijo dažnio identifikacija (RFID), siūlomosios kompiuterine rega paremtos ACCR sistemos naudojimas yra efektyvus metodas paskirstant konteinerius.

Konteinerių identifikacijos numeriai yra koduojami pagal ISO-6346 standartą, pagal kurį standartinis identifikacijos numeris susideda iš vienuolikos ženklų iš kurių keturių didžiųjų lotyniškų raidžių, šešių skaitmenų ir vieno kontrolinio ženklo. Anot autorių šių 11 ženklų koduotės užtenka sudaryti unikaliam konteinerio identifikacijos numeriui, tačiau galimi ir papildomi ženklai arba numeriai.

Autorių ACCR sistema susideda iš 3 pagrindinių blokų:

Tekstinės eilutės regiono aptikimas;
 Identifikacijos numerio simbolių izoliavimas;
 Simbolių atpažinimas(1 pav);



4 pav. ACCR struktūrinė schema [2].

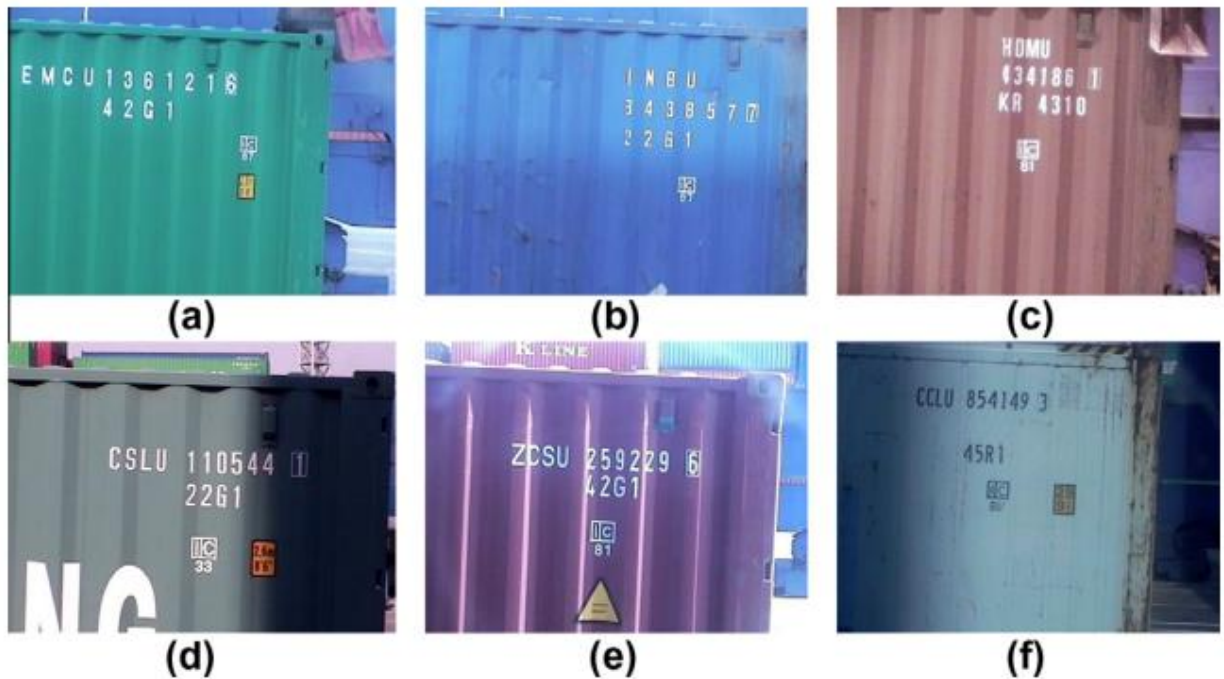
Nors konteinerio identifikacijos numerio atpažinimo struktūrinė schema susideda iš tokių pat blokų, kaip ir automobilio valstybinio numerio atpažinimo schemas (lokalizavimas, izoliavimas ir atpažinimas), tačiau numerių atpažinime yra viena spalva [3], bei numerių plokštelė yra kampuota ir turi standartinį dydį [4] kas labai palengvina atpažinimo procesą (5 pav). Be to atpažįstant konteinerių identifikacijos numerius iškyla problema, dėl nelygaus konteinerio paviršiaus, dėl kurio keičiasi simbolių atspalviai, krinta šešėliai.



5 pav. Automobilių valstybinių numerių pavyzdžiai 1 eilutės ir 2 [4].

Mokslininkų paminėtos dažniausiai išskylančios problemos atpažįstant konteinerių identifikacijos numerius:

- Konteinerio kodas gali būti skirtingose pozicijose;
- Kodas gali būti skirtingų spalvų;
- Gali skirtis kodo šrifto dydis ir pats šriftas;
- Skiriasi kodo eilučių kiekis (gali būti vienoje eilutėje, dviejose ar net trejose eilutėse);
- Skiriasi kodo eilučių lygiavimas (gali būti lygiuota pagal dešinę, pagal kairę arba centruota);
- Maži tarpai tarp simbolių (6 pav);



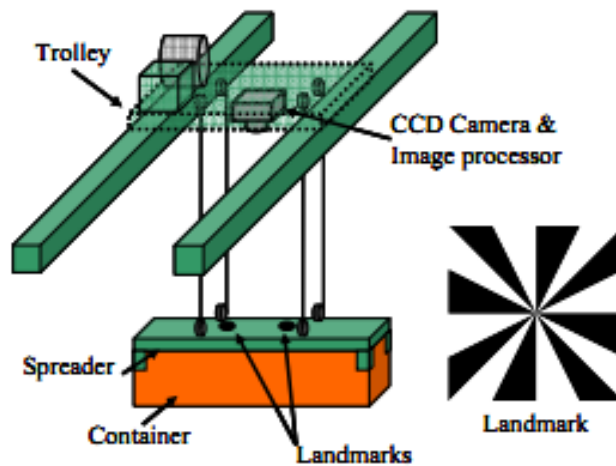
6 pav. Konteinerių kodų paveikslėlių pavyzdžiai. [2]

1.1.2. Atstumo iki konteinerio matavimo algoritmų apžvalga

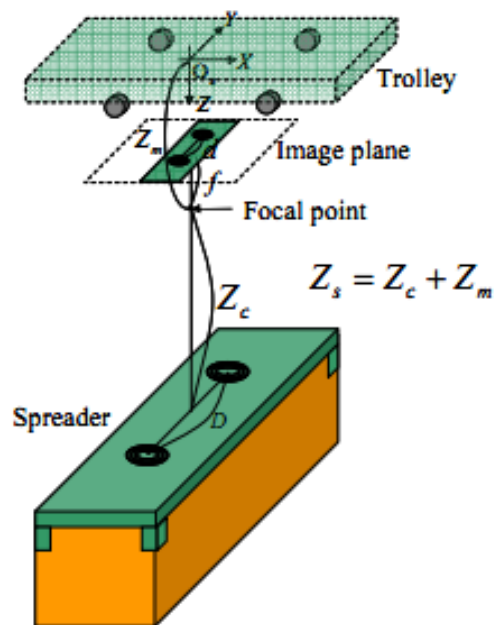
Atpažinus konteinerį ir norint jį pagriebti reikalinga nustatyti koks griebtuvo atstumas iki atpažintojo konteinerio. Tai galima atlikti naudojantis įvairiais atstumo sensoriais tokiais kaip lazeriai, ultragarsiniai sensoriai [32], tačiau šitie minėti sensoriai matuoja tik atstumą ir nepasako iki ko tas atstumas yra, todėl geriau naudotis jau turimomis priemonėmis t.y. vaizdo kamera[2,6,1]. Atstumą matuoti su vaizdo kamera galima naudojantis viena kamera [6] arba dviejų kamerų sistema [1].

Matavimai viena vaizdo kamera: Mokslininkai H.Kawai ir Y.Choi atstumą iki konteinerio matuoja su viena vaizdo kamera pritvirtinta krano viršuje. Ant griebtuvo viršaus yra pritvirtinti du orientyrai – vaizdo žymenys, kuriuos vaizdo jutiklis lengvai išskiria iš supančios aplinkos [6].

Sistema susideda iš dviejų pagrindinių komponentų, tai vaizdo jutiklio bei sukurto vaizdo žymens ant griebtuvo viršaus (7 pav.). Sukurtas vaizdo žymuo yra suprojektuotas taip, kad būtų lengvai išskiriamas iš aplinkos, būtų matomas skirtingais atstumais bei aptinkamas iš skirtingų kampų. Vaizdo jutiklis yra video kamera, kurios duomenis apdoroja procesorius. Šis procesorius apskaičiuoja atstumą iki kiekvieno vaizdo žymens ir paskaičiuoja horizontalų griebtuvo pakrypimą išilgai griebtuvo.



7 pav. Pozicijos aptikimo sistemos vaizdas [6].



8 pav. Griebtuvo aukščio matavimai [6].

Apskaičiavus žinant tikrąjį atstumą tarp žymenų D bei apskaičiavus atstumą tarp žymenų vaizde užfiksuotame vaizdo kameros d pasinaudoję panašųjų trikampių metodą apskaičiuojame griebtuvo aukštį (8 pav):

$$Z_s = Z_m + Z_c \quad (5)$$

$$Z_c = \frac{D}{d} f$$

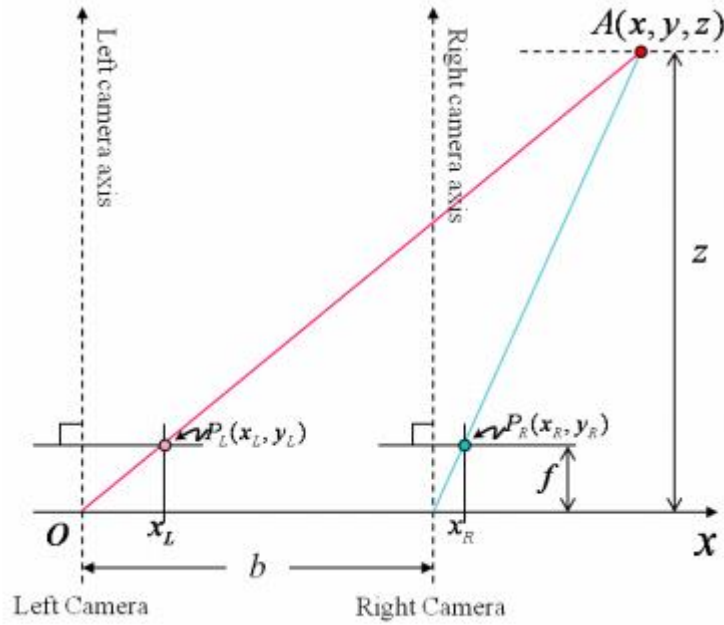
(6)

čia, Z_s - griebtuvo aukštis, Z_m ir Z_c atstumas nuo kameros iki židinio ir atstumas nuo židinio iki griebtuvo padėties, f kameros židinio nuotolis.

Prieš atliekant minėtus skaičiavimus reikia aptikti žymenų padėtį. Nekintant atstumui iki vaizdo objekto galima sumažinti paieškos zoną tačiau šis metodas netinkamas, nes atstumas pastoviai kinta. Dėl šios priežasties vaizdo žymenų reikia ieškoti visame fiksuojamame vaizdo plote, tam reikia naudoti kitus metodus. Vienas iš paplitusių metodų yra šablonų naudojimas. Tokiu atveju turimas vaizdo šablonas yra lyginamas su visame vaizde esančiais objektais, turint atitinkamus leistinus nuokrypius nuo šablono. Šio metodo trūkumas yra didžiulis laiko poreikis, nes reikia skanuoti visą vaizdą. Dar vienas šio metodo trūkumas yra ne atsparumas skirtingam apšvietimui.

Vietoje šio metodo galima naudoti atsparesnį apšvietimo skirtumams vektorinio kodo koreliacijos metodą (VCC). Šis metodas apskaičiuoja vaizdų panašumą lygindamas vaizdo vektorinius kodus, vietoje atskirų taškų spalvos intensyvumo lyginimo. Metodas apskaičiuoja gradientus pagal greta esančių 8 taškų intensyvumą kiekvienam taškui su šablonu, taip gaudamas nesutapimo laipsnį. Gradientai gali turėti 3 dviejų bitų reikšmes (kryptis) 00 kai intensyvumas visomis kryptimis nekinta, 01 kai intensyvumas didėja x ir y ašimis, ir 10 kai intensyvumas mažėja x ir y ašimis.

Matavimai dviem vaizdo kamerom: Kadangi vaizdas yra fiksuojamas dviem vaizdo kamerom, autoriai vietoje papildomų atstumo jutiklių siūlo naudoti trianguliaciją. 9 paveiksle matomas geometrinis sistemos vaizdas, kur $A(x,y,z)$ konteinerio pozicija b – atstumas tarp kairiosios ir dešinėsios kamerų, f - kamerų židinių atstumas, P_L ir P_R atitinkamai kairiosios ir dešinėsios kameros fiksuojamas konteinerio vaizdai esantys vienoje linijoje ($x_L = x_R$) dėl sutampančio židinio nuotolio. Autoriai įtraukia nesutapimo dydį d kuris atsiranda dėl skirtingų kamerų pozicijų (x_L ir x_R t.y. vaizdas yra fiksuojamas iš skirtingų kampų.



9 pav. Atstumo iki konteinerio dviem vaizdo kameromis geometrinė schema [1].

Sistema pasirenka konkretų vaizdo tašką gaunamą abiejuose kamerų užfiksuotuose vaizduose, kuris yra spindulių OP_L ir OP_R susikirtimo taškas. Pasinaudojus panašiųjų trikampių teorija [11] išsivedame formules 7 ir 8:

$$\frac{x_L}{f} = \frac{x}{z} \quad (7)$$

$$\frac{x_R}{f} = \frac{x - b}{z} \quad (8)$$

Suvedę abi formules gauname atstumo nuo konteinerio griebtuvo iki konteinerio apskaičiavimo formulę:

$$z = \frac{bf}{x_L - x_R} = \frac{bf}{d} \quad (9)$$

Atstumas tarp kamerų b ir kamerų židinių nuotolis f yra žinomi ir yra konstanta nes kameros yra nejudamoje padėtyje ir židinio nuotolis nėra keičiamas. Iš 9 formulės matome, kad atstumas iki taško A mažėja didėjant skirtumui d ir atvirkščiai, kai skirtumas mažėja – atstumas didėja. Norint prijungti konteinerio griebtuvą prie konteinerio reikia surasti atstumą iki konteinerio centro, kuris yra apskaičiuojamas žinant konteinerio skylių koordinates.

Minėtų mokslininkų naudojamos sistemos puikiai tvarkosi su konteinerio pagriebimu bei pozicijos suradimu. Tačiau konteinerių krovos metu tai tėra vienas iš daugelio vykdomų procesų. Pakėlus konteinerį jį reikia ir padėti. Šioje sistemoje naudojamos kameros su jų pozicija, konteinerio padėjimui visiškai netinka, kadangi paėmus konteinerį, kameros visiškai užsidengia ir nebemato aplinkos. Šią problemą galima išspręsti kameras tvirtinant kitoje vietoje, kur visos krovos metu būtų matomas ir konteinerio griebtuvas ir konteineris, ir vieta kurioje reikia padėti konteinerį.

1.1.3. Vertikalių bei horizontalių viršūnių radimas

Tekstinės eilutės regiono aptikimas yra pirmasis autorių siūlomos ACCR sistemos funkcinis blokas, kurio pagrindinė funkcija išskirti ant konteinerio esančią teritoriją su tekstu iš viso konteinerio paviršiaus. Pagrindinės problemos, tiksliai aptinkant teksto zoną, su kuriomis autoriai susiduria yra prasta vaizdo kokybė, kintantis apšvietimas, konteinerio nelygumai, krintantys šešėliai, bei įvairūs atspindžiai. Atsižvelgiant į tai, kad konteinerio ir identifikacijos numerio spalvos skiriasi viena nuo kitos ir pačios spalvos nustatymas nėra aktualus, vaizdas yra transformuojamas iš spalvoto vaizdo į bespalvį. Vaizdo transformacijai yra naudojama 10 formulė:

$$L = 0.299R + 0.587G + 0.114B \quad (10)$$

čia, R – raudona, G – žalia, B – mėlyna – spalvoto paveikslo atspalviai;

Atsižvelgiant į tai, kad identifikacijos numeriai turi stiprias vertikalias tekstūras ir spalvos pokytis, pereinant nuo konteinerio paviršiaus prie teksto ženklų, yra labai didelis ir identifikacijos numerio zonoje pasitaiko daug dažniau, nei ant viso konteinerio paviršiaus, autoriai siūlo zoną, kurioje dažnai pasitaiko horizontalūs tekstūros pokyčiai, priskirti teksto eilutės regionui. Vertikalių tekstūrų viršūnių aptikimui yra naudojamas horizontalus aukšto dažnio filtras:

$$I_{vg} = I * H \quad (11)$$

čia, I ir I_{vg} atitinkamai originalios ir vertikalios vaizdo tekstūrų viršūnės, daugybos operatorius atitinka konvoliucijos operaciją, H – horizontalus aukšto dažnio filtras kuris išreiškiamas:

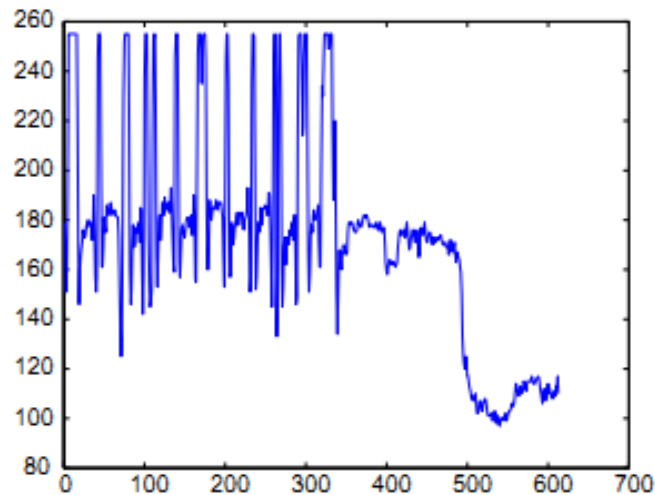
$$\frac{1}{n} \underbrace{[-1, \dots, -1, k, -1, \dots, -1]}_n \quad (12)$$

kur, n yra horizontalaus aukšto dažnio filtro ilgis (kiekis šiuo atveju 13), o $k = 1 - n$, filtracijos rezultatas pateiktas 11 paveiksle.

Po filtracijos, kiekvienas taškas viršijantis atitinkamą slenkstį yra traktuojamas, kaip pagrindinis taškas. Pasinaudojus sekančia funkcija autoriai gauna binarinį konteinerio vaizdą, kuriame pagrindiniams taškams yra suteikiama 1 reikšmė, o kitiems 0, atitinkamai vaizdas yra baltas arba juodas.

$$I_{vb}(x,y) = \begin{cases} 0, & I_{vg}(x,y) < \beta M_v(x,y) \\ 1, & I_{vg}(x,y) > \beta M_v(x,y) \end{cases} \quad (13)$$

čia, M_v yra vidutinė vaizdo šviesos stiprumo reikšmė gaunama vidurkinant viso vaizdo šviesos stiprumą. $M_v(x,y)$ atitinka tam tikros zonos, kurioje yra taškas (coordinates x,y) stiprumo vidurkio reikšmę. Šviesumo stiprumo vidurkis išvedamas atitinkamoms zonoms, o ne visam vaizdui, dėl galimų apšvietimo pokyčių vaizde. β koeficientas, kurio reikšmė autoriai priskiria 0.1, ją nustatė bandymų metu.



10 pav. Šviesos stiprumo kitimas konteinerio identifikacijos numerio zonoje [2].

Po filtracijos reikalinga pašalinti visas netinkamas zonas, vadinamas ne simbolių zonas. Pirmiausia yra matuojami baltų zonų aukščiai ir atmetamos visos zonos, kurių aukštis yra mažesnis už galimą konteinerio identifikacijos simbolio aukštį. Taip pat yra atmetamos visos zonos kurių aukštis yra didesnis už maksimalų galimą konteinerio identifikacijos numerio simbolio aukštį.

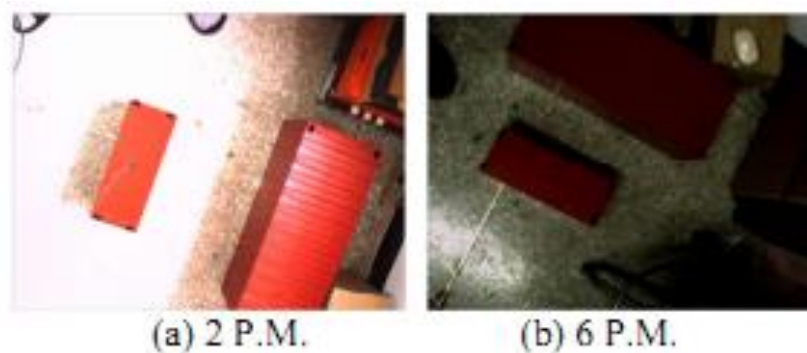


11 pav. Konteinerio vaizdo filtravimas su horizontaliuoju aukšto dažnio filtru [2].

Atmetus nesimbolines zonas yra analizuojamas likęs vaizdas. Pirmiausia surandami šuoliai iš juodos į baltą ir tada į juoda. Vienas šuolis prasideda pasikeitimu iš juodos į altą ir užsibaigia pasikeitimu iš baltos į juodą. Suradus visus šuolius yra matuojami šuolių plotai. Pirmiausia atmetami siauri šuoliai kurių plotis neviršija minimalaus galimo pločio. Tada tikrinama, kiek šuolių seka vienas po kito. Nagrinėjamu atveju vienoje tekstinėje zonoje turi būti bent 3 vienas po kito sekantys šuoliai.

Simbolinės zonos atskyrimui, nuo atspindžių zonos yra pasinaudojama Ostu segmentavimo metodu, pritaikant jį kiekvienai zonai atskirai [5]. Analizuojant šviesios spalvos ryškiai apšviestus konteinerius išryškėja šio metodo trūkumas – sunkiai išskiriamas simbolis iš fono. Be to sunkiau atpažįstami simboliai kai vaizdas paimamas nuo judančio konteinerio. Šiuos trūkumus būtų galima bandyti šalinti į sistemą įtraukiant daugiau kamerų pastatytų skirtingais kampais. Tokiu atveju būtų galima analizuoti kelis to paties konteinerio vaizdus ir juos sulyginus gauti kur kas geresnius rezultatus.

Naudojant videokameras lauko sąlygomis t.y. konteinerių terminaluose, reikia atsižvelgti į tai kad fiksuojamo vaizdo kokybei labai didelę įtaką daro kintančios oro sąlygos, apšvietimas, tamsa, krituliai. Aplinkos įtakos sumažinimui galima padidinti kameros užlaikymo laiką. Užlaikymo laiko didinimas didina fiksuojamo vaizdo ryškumą ir vaizdas gaunamas aiškesnis. Skirtingu paros laiku taip pat yra gaunami skirtingo ryškumo vaizdai (12 pav.).



12 pav. Skirtingu paros metu fiksuojamas vaizdas [1].

Šią problemą galima išspręsti priklausomai nuo gaunamo vaizdo ryškumo keičiant užlaikymo laiką, taip gaunant automatiškai susikonfigūruojančią sistemą ir sumažinant šviesos įtaką vaizdo kokybei.

1.2. Konteinerių krovos krano valdymo sistemų apžvalga

1.2.1. Prisitaikanti neraiškaus tinklo valdymo sistema nuolatinės srovės variklio valdymui

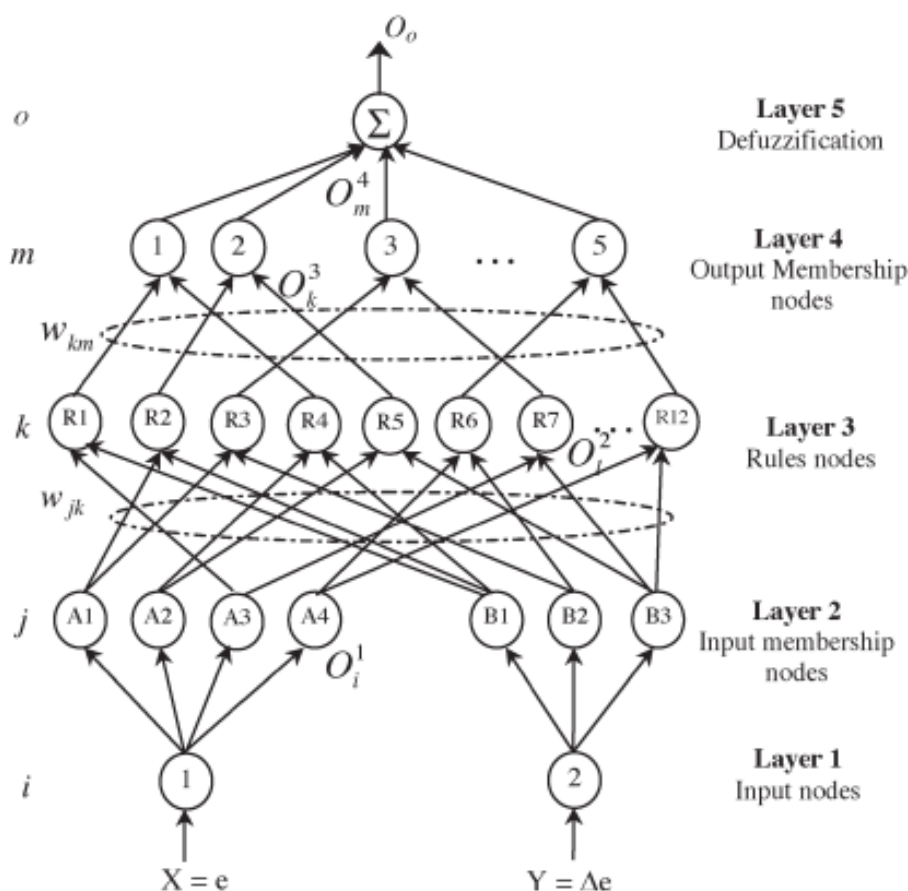
Automatizuojant konteinerių terminalą reikalinga ne tik automatiškai pasinaudojus vaizdų atpažinimu surasti konteinerį, griebtuvą bei platformą, tačiau reikalinga ir pagal gautus iš vaizdo duomenis atitinkamai valdyti variklius. Kol kas visi krano varikliai yra valdomi rankiniu būdu operatoriaus. Konteinerio padėjimas priklauso nuo operatoriaus reakcijos, be to reikalinga gautus duomenis atvaizduoti ekrane, kas dar labiau sulėtina realiu teisingai valdyti kraną.

Mokslininkas Ahmed Rubaai kartu su Abdul R. Ofoli nagrinėja nuolatinės srovės variklių valdymo sistemą ir kuria automatinį valdymą, pasitelkdami intelektualius metodus tokius kaip neuroninis tinklas ir neraiškioji logika. Autoriai apjungia neraiškiosios logikos ir neuroninio tinklo struktūrą sukurdami prisitaikančio neuroninio tinklo su neraiškiaja sąsaja sistemą (ANFIS), kuri apjungia tiek neuroninio tiek ir neraiškiosios logikos galimybes į vieną. Autoriai taip pat aprašo šios sistemos apmokymo algoritmą, kuris pagal sugeneruotą mokymo duomenų aibę koreguoja sistemos svorius, kad pagal atitinkamą įvestį būtų formuojamas kuo tikslesnis išėjimo signalas.

Adaptyvaus neuroninio-neraiškaus tinklo struktūra: ANFIS struktūra yra paimama iš neuroninio tinklo, svorių naudojimas taip pat paimamas iš neuroninio tinklo. Neraiškioji logika čia prisijungia norint suformuoti ANFIS įėjimą fuzifikuojant kintamuosius, bei išvedant išvesties signalus išraiškinimo metodais(13 pav.).

Autorių pateiktoje ANFIS sistemoje pirmiausia, neraiškiosios logikos valdymo struktūrai yra parenkamos lingvistinės taisyklės. Tada, skaitmeniniai duomenys(sugeneruota mokymų aibė) yra panaudojami šios sistemos apmokymui. Įėjimo narystės funkcijų skaičius yra pasirinktas, kad

apimtų visą galimą įvesties erdvę.



13 pav. *Prisitaikančio neuroninio-neraiškaus tinklo architektūra [16].*

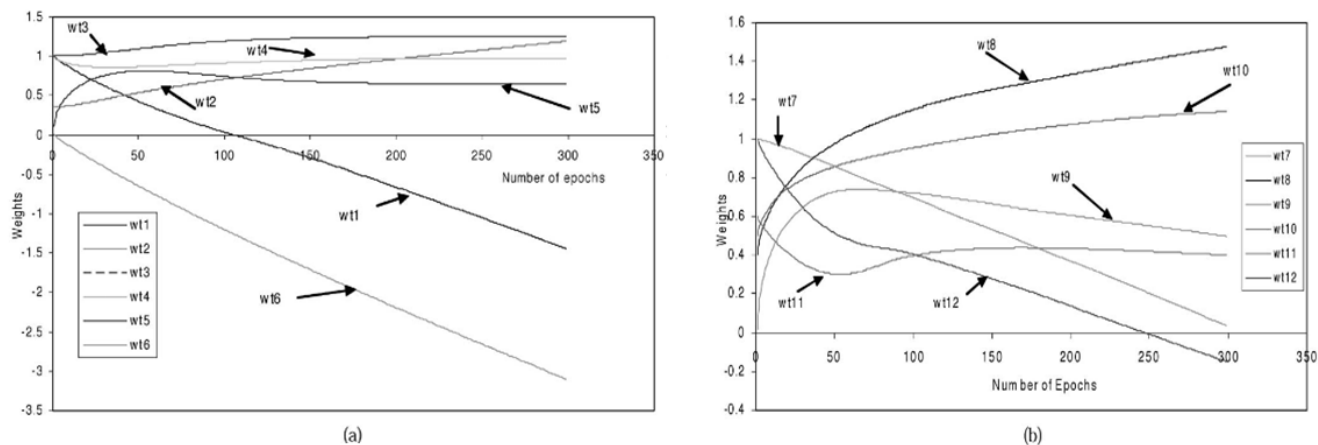
Autoriai ANFIS suprojektavo taip, kad projektuotojas norėdamas pritaikyti ANFIS savo sistemoje, lengvai suprastų vidinius sluoksnius, vadinamus paslėptaisiais sluoksniais, ir suprastu jų tarpusavio sąsają. Pateiktas ANFIS yra daugiasluoksnis neuroninio tinklo valdiklis susidedantis iš keletos sluoksnių:

- Įvesties sluoksnis;
- Įvesties narystės funkcijų sluoksnis;
- Taisyklių sluoksnis;
- Išvesties narystės funkcijų sluoksnis;
- Išraiškinimo sluoksnis;

Apmokymo algoritmas koreguoja ANFIS svorius, esančius tarp 3 ir 4 sluoksnio priklausomai nuo to kurie svoriai dalyvavo einamosios iteracijos išvesties signalo skaičiavime. Svoriai koreguojami remiantis grįžtamojo ryšio pagalba tikrinant realų rezultatą su apmokymo aibėje esančiu siektinu rezultatu.

Svoriai jungiantys taisyklių ir išvesties narystės sluoksnius, yra apmokomi, kad pastebėtų

sistemos dinamiką ir atitinkamai keistų rezultatą. Svorių kitimas apmokymo metu, priklausomai nuo iteracijų skaičiaus yra pateiktas 14 pav. Pažymėti svoriai, w_{t5} ir w_{t6} buvo nustatyti prie labai mažo skaičiaus 0.001 su pradine prielaida, kad tie svoriai yra perteklinai prieš prasidedant mokymui. Tačiau, neuro-fuzzy sistema ištaiso šias prielaidas. Svoris w_{t5} nukreipiamas į teigiamą vertę, kuri nurodo sužadinimo taisyklės svorį, kai tuo metu w_{t6} nukreipiamas į neigiamą skaičių, kuris slopina taisyklės svorį. Kai kurie iš svorių pakeičia ženklą per mokomąjį procesą. Svoris w_{t1} pradžioje buvo nustatytas teigiamas, bet konvergavo į neigiamą vertę, tai taisyklės svorį daro slopinantį.



14 pav. ANFIS apmokymo metu kintantys svoriai[16].

Apmokymas vykdomas tol kol mokymo metu išmatuota paklaida sumažėja iki 0,01 reikšmės. Tačiau ši paklaida yra kvadratinė ir variklių valdymui yra gana didelė, o sistemos apmokymas iki 10 kartų mažesnės paklaidos užimtų labai daug laiko. Tokios sistemos apmokymą geriausia būtų vykdyti sistemos veikimo įrenginiuose metu.

1.2.2. Prisitaikanti valdymo sistema paremta skatinamuoju mokymusi

Didėjant žmonių populiacijai didėja ir elektros energijos poreikis. Atsižvelgiant į tai kad neatsinaujinančių energijos šaltinių visam laikui neužteks, vis labiau ir labiau yra kreipiamas dėmesys į atsinaujinančius elektros energijos šaltinius. Vienas iš populiariausių tokių šaltinių yra vėjas, kurio kinetinė energija per vėjo malūno mentes yra verčiama sukamuoju judesiu ir per generatorių paverčiama elektros energija. Pagrindinis energijos gavimo iš atsinaujinančių energijos šaltinių privalumas yra tas, kad į aplinką nėra išmetama kenksmingų medžiagų ar dujų, kaip deginant kurą, arba radioaktyvių atliekų, kaip atominėse elektrinėse.

Yra siūloma daugybė schemų ir būdų minėtos energijos išgavimui, tačiau vis dar didžiausias trūkumas yra didžiulis išgaunamos elektros energijos netiesiškumas. Todėl yra reikalingos labai sudėtingos netiesinės valdymo sistemos, kurių efektyvumas labai priklauso nuo valdymo sistemos

tikslumo ir veikimo greičio. Daugybė mokslininkų bando sukurti valdymo sistemas elektros energijos iš vėjo optimaliam generavimui. Pavyzdžiu Kanellos ir Hatziaargyriou[18], Yong-tong ir Cheng-zi[19], kaip ir Zhao-da su kitais [20] naudoja neuroninius tinklus kaip galingą įrankį netiesinės valdymo strategijos kūrimui. Dažniausiai tokiose sistemose yra naudojamas daugiasluoksnis perceptronas (MLP). Tačiau minėtų autorių siūlomas sistemas iš pradžių reikia apmokinti ir tada naudoti pačioje vėjo jėgainėje.

Mokslininkai M.Sedighizadeh ir A.Rezadeh siūlo intelektualią skatinamąjį mokymų sistemą vėjo energijos generavimo sistemą (WECS) , kuri galėtų realiu laiku save apmokinti ir optimaliai valdyti vėjo jėgainę. Naudojant skatinamąjį mokymą autoriai šalia neuroninio tinklo įdiegia agentą aktorių/kritiką, kuris realiu laiku tiria aplinką ir pagal sukauptus duomenis modifikuoja neuroninio tinklo svorius. Toks mokymosi principas veikia gyvūnų bei žmonių smegenyse.

2. METODINĖ DALIS

2.1. Vaizdo atpažinimo algoritmai

2.1.1. Hough transformacija

Sukurtos posistemės automatinio vaizdo atpažinimo dalyje, naudojama *Hough* transformacija (angl. *Hough transform*) paveiksle suranda figūrą pagal surastus briaunų taškus. Šiuo būdu galima rasti pirmines formas, kurios aprašomos lygtimis arba polinomais. Nustačius dėsningumą (aproksimavus) pasirinkta forma, galima tikslinti taškų vietas kartu patikslinant detektuotą kontūrą (briaunas).

Su Hough transformacijos metodu galime per duotą taškų aibę išbrėžti liniją. Žinant du taškus per juos galima išbrėžti liniją, kuri aprašoma:

$$y = b + mx \quad (14)$$

kur m ir b – tiesės lygties koeficientai. Jei turime du taškus, galima rasti minėtas konstantas ir gauti lygties koeficientus. Tiesė taip pat gali būti aprašyta tašku (m, b) specialioje koeficientų koordinačių erdvėje. Kiekvienas taškas šioje erdvėje atvaizduoja skirtingas tieses. Duotas taškas $P(x_1, y_1)$ gali priklausyti bet kuriai tiesei, kurios koeficientai m ir b tinka lygybei:

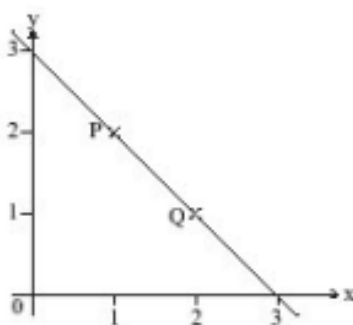
$$y_1 = b + mx_1 \quad (15)$$

Jei x_1 ir y_1 – konstantos, tai yra tiesė (m, b) koeficientų erdvėje:

$$b = -x_1 m + y_1 \quad (16)$$

Ši tiesė atvaizduoja visas tieses (x, y) koordinačių erdvėje, kurios gali eiti per tašką (x_1, y_1) . Kiekvienas taškas ir (x, y) koordinačių erdvės yra atvaizduojamas linija (m, b) koordinačių erdvėje. Skirtingi taškai bus atvaizduoti skirtingomis linijomis. Kolineariųjų taškų tiesės (m, b) koordinačių erdvėje susikirs tame pačiame taške. Šio susikirtimo koordinatės yra tiesės, kuriai priklauso taškai, lygties koeficientai. Taigi susikirtimo tašką (m, b) koordinačių erdvėje galima pavaizduoti linija (x, y) koordinačių erdvėje.

Taškai $P(1,2)$ bei $Q(2,1)$ priklauso tai pačiai tiesei $y = b + mx$, kurios koeficientai: $m = -1$, $b = 3$ (15 pav).



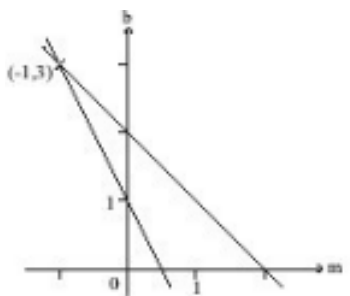
15 pav. Taškai P ir Q(x,y) koordinačių erdvėje[37]

Šie du taškai duoda dvi tieses (m, b) koordinačių erdvėje (16 pav), kurių tiesės lygtys yra tokios:

$$b_P = 2 - m_P \quad (17)$$

$$b_Q = 1 - m_Q \quad (18)$$

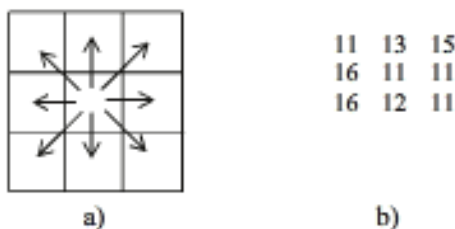
Šių dviejų tiesių susikirtimo taškas (m,b) koordinačių erdvėje turi koordinates (-1, 3), susikirtimo taško koordinatės ir yra taškų ir (x, y) koordinačių sistemos tiesės lygties koeficientai



16 pav. Taškai P ir Q(x,y) koordinačių erdvėje[9,37]

2.1.2. Homogeniškumo operatorius, Sobelio briaunų aptikimas

Paprasčiausio ir greičiausio briaunos radimo pirmais etapas – apskaičiuoti vidurinio ir šalia esančių 8 kaimyninių taškų skirtumus (vadinamus homogeniškumo operatoriumi, angl. *homogenous operator*), kai lango dydis 3x3 (17 pav.).

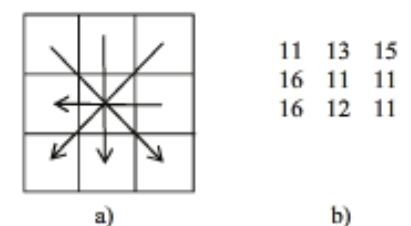


17 pav. Homogeniškumo operatoriaus veikimas: a) veikimo principas; b) paveikslo fragmentas[8,37]

Antrajame etape iš gautųjų reikšmių modulių išrenkamas didžiausias gautas skirtumas, kuris ir tampa naująja taško reikšme:

$$\text{nauja reikšmė} = \max \{|11-11|, |11-13|, |11-16|, |11-16|, |11-11|, |11-16|, |11-12|, |11-11|\} = 5 \quad (19)$$

Dar greitesnis veikimas gaunamas skaičiuojant ne 8, o tik 4 priešingų kaimyninių taškų skirtumus: viršutinio dešiniojo kampo su apatiniu kairiuoju kampu, viršutinio kairiojo su apatiniu dešiniuoju, vidurinio stulpelio viršutinio su vidurinio stulpelio apatiniu ir vidurinės eilės dešiniojo su vidurinės eilės kairiuoju (18 pav.).



18 pav. Homogeniškumo operatoriaus II veikimas: a) veikimo principas; b) paveikslo fragmentas[37]

$$\text{nauja reikšmė} = \max \{|11-11|, |13-12|, |15-16|, |11-16|\} = 5 \quad (20)$$

Sobelio briaunų radimo operatorius yra jautrus (t. y. geriau nustatantis) ir gerai aptinka objektų kontūrus. Sobel 3x3 šablonas:

X kryptimi:

-1 -2 -1

0 0 0

1 2 1

Y kryptimi:

-1 0 1

-2 0 2

-1 0 1

Jei pradinis paveikslas yra:

0 0 0 0 0 2 0 3 3

0 0 0 1 0 0 0 2 4 2

0 0 2 0 2 4 3 3 2 3

0 0 1 3 3 4 3 3 3 3

0 1 0 4 3 3 2 4 3 2

0 0 1 2 3 3 4 4 4 3

Tai suminis rezultatas su Sobel operatoriumi bus:

4 6 4 10 14 12 14 4

6 8 10 20 16 12 6 0

4 10 14 10 2 4 2 4

2 12 12 2 2 4 8 8

Išskirtos briaunos, kai pritaikytas slenkstis yra ≥ 12 :

0 0 0 0 1 1 1 1

0 0 0 1 1 1 0 0

0 0 1 0 0 0 0 0

0 1 1 0 0 0 0 0

Taikant Sobelio operatorių ir pritaikius atitinkamą slenkstį iš nespaltvoto vaizdo yra gaunamas

juodai baltas vaizdas (binarinis), kuriame baltai žymimi taškai atpažinti, kaip kontūras, o juodai žymimos visos kitos nekontūrinės linijos.

2.1.3. Panašieji trikampiai

Taip pat mano posistemėje atstumo iki kontainerio radimui bei, taškų skaičiaus vertimui matmenimis yra naudojamos panašiųjų trikampių taisyklės. Jei vieno trikampio kampai yra lygūs atitinkamiems kito trikampio kampams, o jo kraštinės yra proporcingos atitinkamoms kito trikampio kraštinėms, tai sakome, kad pirmasis trikampis panašus į antrąjį. Trikampių panašumo požymiai[11]:

- Jeigu vieno trikampio du kampai atitinkamai lygūs kito trikampio dviem kampams, tai trikampiai yra panašūs
- Jeigu vieno trikampio dvi kraštinės proporcingos kito trikampio dviem kraštinėms ir kampai tarp tų kraštinių lygūs, tai trikampiai yra panašūs.
- Jeigu vieno trikampio visos trys kraštinės proporcingos kito trikampio kraštinėms, tai trikampiai yra panašūs.
- Panašiųjų trikampių perimetrų santykis lygus panašumo koeficientui. Panašiųjų trikampių plotų santykis lygus panašumo koeficiento kvadratui.

Žinodami atitinkamus vieno trikampio duomenis ir tik dalį kito trikampio, galime surasti visus kitus mums reikalingus duomenis. Pasinaudoję panašiųjų trikampių savybėmis galime paskaičiuoti kokiu atstumu yra vaizdo kamera nuo atpažinto kontainerio. 19 paveiksle pavaizduota geometrinė schema kurios O taške yra vaizdo kamera $A1 - B1$ tikrasis objekto ilgis(y), o $A2 - B2$ tai vaizdo kamera gautas objekto ilgis (x), $F1$ – kameros židinio nuotolis, $F2$ – atstumas nuo vaizdo kameros gaunamo vaizdo iki tikrojo objekto, F – atstumas nuo vaizdo kameros iki objekto.

Žinomi dydžiai: tikrasis objekto ilgis(y), vaizdo kamera gautas objekto ilgis(x) vaizdo kameros židinio nuotolis($F1$);

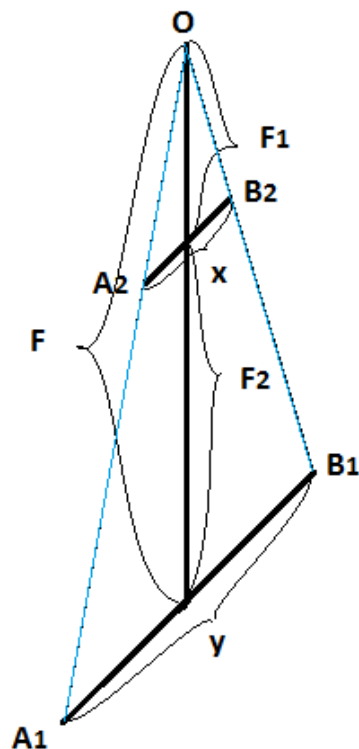
Ieškomas dydis: Atstumas nuo vaizdo kameros iki objekto(F);

Pagal panašiųjų trikampių dėsninę sąlyką:

$$\frac{F}{F1} = \frac{y}{x} \quad (19)$$

Apskaičiuojame tikrąjį atstumą iki objekto:

$$F = \frac{y}{x} F1 \quad (20)$$

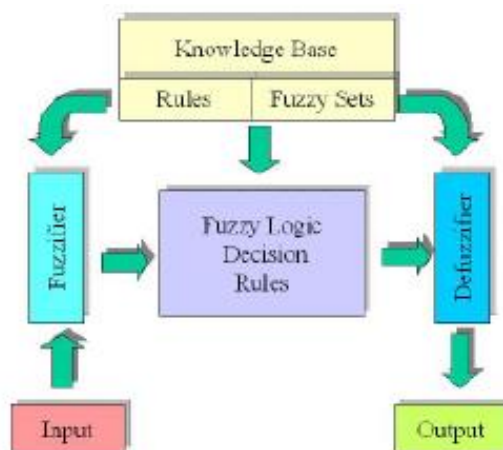


19 pav. Panašųjų trikampių geometrinė schema.

2.2. Valdymo sistemų metodai

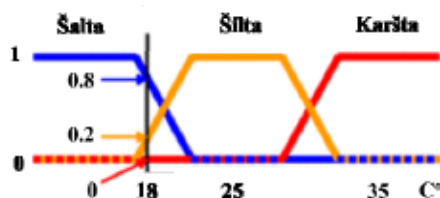
2.2.1. Neraiškosios logikos valdymo sistemos

20 paveiksle yra pavaizduota neraiškosios logikos sprendimų priėmimo sistema, kuri ir yra laikoma neraiškiaja valdymo sistema (angl., *Fuzzy Control System*), toliau FCS. 20 paveiksle matome FCS sudedamąsias dalis: įėjimas, įėjimo fuzifikacija, žinių bazė sudaryta iš taisyklių ir neraiškiųjų aibių, neraiškosios logikos sprendimų priėmimo blokas, defuzifikacija, išėjimas[22].



20 pav. Neraiškosios sprendimų priėmimo sistemos principinė diagrama [23]

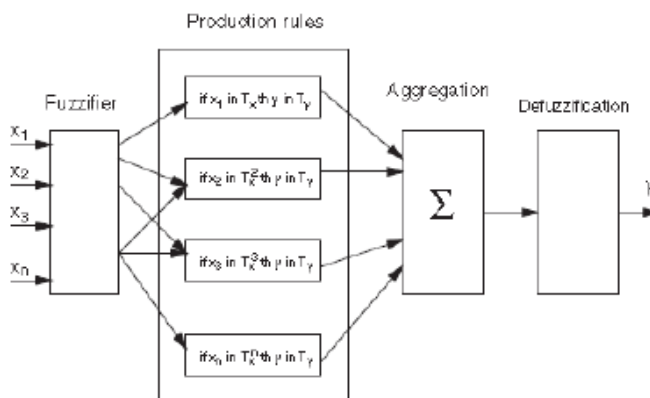
Įėjimai gali būti tolydieji dydžiai, kuriuos nusako tam tikros funkcijos ribose [0, 1]. 21 paveiksle pateiktas neraiškiosios logikos fuzifikacijos procesas, kur pateikiamos 3 temperatūros funkcijos: šalta, šilta ir karšta. Šiuo atveju temperatūra yra įėjimas, o fuzifikacijos proceso tikslas yra nustatyti kiekvienos funkcijos reikšmę pagal įėjimo temperatūrą. Pvz., turime temperatūrą 18Co. Ties 18Co išvedus statmenį x ašiai, gauname susikirtimo taškus su temperatūros funkcijomis.



21 pav. Neraiškiosios logikos įėjimo (temperatūros) funkcijos, dar kitaip vadinama fuzifikacija

Tie susikirtimo taškai nusako kiekvienos funkcijos reikšmę: $\text{šalta}=0,8$, $\text{šilta}=0,2$, $\text{karšta}=0$. Šios reikšmės yra vadinamos neraiškiais dydžiais. Tokiu būdu kiekvienai funkcijai sudaroma neraiškioji aibė, kur poruojama įėjimo reikšmė su atitinkančia neraiškia reikšme pagal funkciją: $\tilde{A} = \{x, \text{šalta}_A(x) \mid x \in X\}$. čia x – reiškia įėjimo temperatūros reikšmė, – funkcijos $\text{šalta}_A(x)$ neraiški reikšmė nuo x , X – reiškia temperatūros reikšmių aibė.

Taisyklės aiškiai nusako ryšį tarp įėjimo kintamųjų ir išėjimo. Čia naudojama *IF/THEN* kalba. Pvz., *IF* $x=\text{šalta}$ *AND* $x=\text{šilta}$ *THEN* $\text{orotem}=\text{vidutiniškaišilta}$. Tokius išvedimus atlieka *neraiškiosios logikos sprendimų taisyklių blokas*. Neraiškioji išvedimo sistema pateikta 22 pav. Esant daugiau nei vienai taisyklei, toliau vykdomas *agregavimo procesas*, kuris pagal kiekvienos taisyklės padarytus išvedimus apskaičiuoja kiekvieno kintamojo neraiškiosios aibės elementus. Po agregavimo vykdomas defuzifikacijos procesas, kuris transformuoja neraiškiosios aibės elementus į reiškiosius skaičius [21].



22 pav. Neraiškiosios išvedimo sistemos diagrama [21]

2.2.2. Dirbtinių neuroninių tinklų (DNT) taikymas valdymo sistemoms

Mano sukurtoje posistemėje naudojami dirbtiniai neuroniniai tinklai (DNT, *artificial neural networks*, ANN) yra taikomi ne tik klasifikavimo, klasterizavimo, funkcijų aproksimavimo, prognozavimo, optimizavimo uždaviniams spręsti, bet ir daugiamačių duomenų (objektų) matmenų skaičiui mažinti, tuo pačiu ir daugiamačiams duomenims (objektams) vizualizuoti. Dirbtiniai neuroniniai tinklai dažnai padeda atskleisti daugiamačių duomenų savybes, kurių negalima pastebėti klasikiniiais daugiamačių duomenų vizualizavimo metodais. Šiuo metu yra sukurta įvairių dirbtinių neuroninių tinklų ir jų mokymo algoritmų daugiamačiams duomenims vizualizuoti [24].

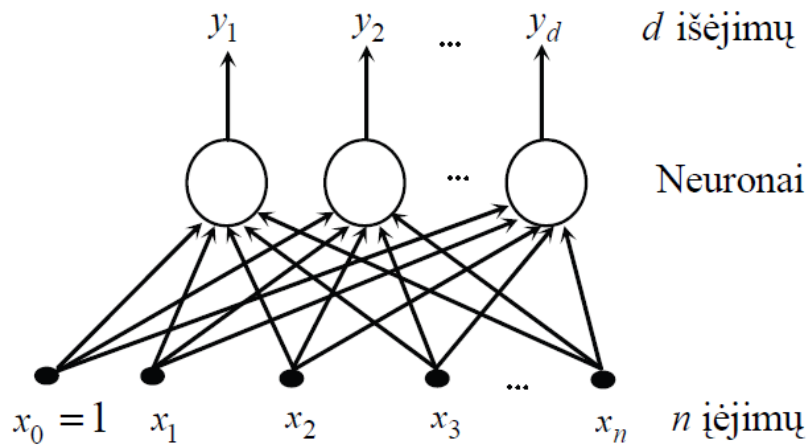
DNT sudaromi iš daugelio tarpusavyje sujungtų paprastų skaičiavimo elementų, kurie tarpusavyje sujungti skirtingo stiprumo jungtimis (sinapsėmis). Mokymosi metu smegenyse keičiasi jungčių, siejančių neuronus, stiprumas. Toks jungčių stiprumo kitimas būdingas ir DNT. DNT mokyti naudojami duomenų pavyzdžiai. Mokymo metu, duomenims veikiant dirbtinį neuroninį tinklą, specialiais algoritmais iteratyviai keičiami jungčių stiprumo koeficientai, vadinami svoriais [25]. Informacija, reikalinga konkrečiam uždaviniui spręsti, sukaupiama svorių vertėse.

Pirmiausias uždavinys kylantis kuriant dirbtinį neuroninį tinklą yra tinkamo dirbtinio neuroninio tinklo tipo pasirinkimas duotajam uždaviniui. Yra nemažai dirbtinių neuroninių tinklų tipų, tačiau juos galima suskirstyti į keturias pagrindines grupes, tiesioginio sklidimo (angl. *feed-forward*), rekurentiniai (angl. *recurrent*), save organizuojantys (angl. *self-organising*) ir hibridiniai, kurie turi kelių tipų viename dirbtiniame neuroniniame tinkle.

Kadangi nėra tiksliai nustatytų sąsajų tarp žmogaus fiziologinių parametrų ir jo emocijų, kiekvieno žmogaus fiziologinis atsakas emocijai yra skirtingas, bei tas faktas jog žmogus gali išsitreniruoti kontroliuoti savo fiziologinius parametrus, pvz. budistai vienuoliai meditacijos pagalba sulėtina savo širdies plakimą beveik iki jos sustojimo, taip pat žmonės ne visada žino ar supranta kaip jaučiasi. Visi šie kriterijai gerokai pasunkina matavimo duomenų rinkimą, emocijų atpažinimą.

2.2.3. Perceptronas, jo mokymas

Kadangi mano posistemėje naudojamas neuroninis tinklas, jis turi būti apmokintas. Paprasčiausios architektūros yra vadinamieji tiesioginio sklidimo (feedforward) neuroniniai tinklai, kuriuose galimos tik vienkryptės jungtys į priekį (unidirectional forward) iš įėjimų į išėjimus. Paprasčiausias tokio tipo neuroninis tinklas yra perceptronas, sudarytas iš vieno sluoksnio d neuronų, sujungtų su n įėjimų (23 pav.) [29].



23 pav. Perceptronas [29]

Neuronų skaičius d yra lygus išėjimų skaičiui. Kai kurie autoriai jau vieną neuroną vadina perceptronu, tada $d = 1$. Kiekvienas perceptrono išėjimas y_j , $1, \dots, d = \dots$, yra įėjimų $x_1, x_2, \dots, x_n$ funkcija, kuri skaičiuojama pagal šią formulę:

$$y_j = f(a_j) = f\left(\sum_{k=0}^n w_{jk} x_k\right), j = 1, \dots, d, \quad (26)$$

čia w_{jk} yra jungties iš k -ojo įėjimo į j -ąjį neuroną svoris, x_0 – papildomas įėjimas, įprastai $x_0 = 1$ [36].

2.3. Programinės įrangos apžvalga

Kadangi sukurtoji posistemė susideda iš kelių skirtingų dalių (valdymo blokas, vaizdo atpažinimo blokas, bei apmokymo dalis), todėl buvo naudojama skirtinga programinė įranga, atliekanti specifinius veiksmus. Iš daugybės galimų pasirinkti variantų buvo išsirinktos trys programos:

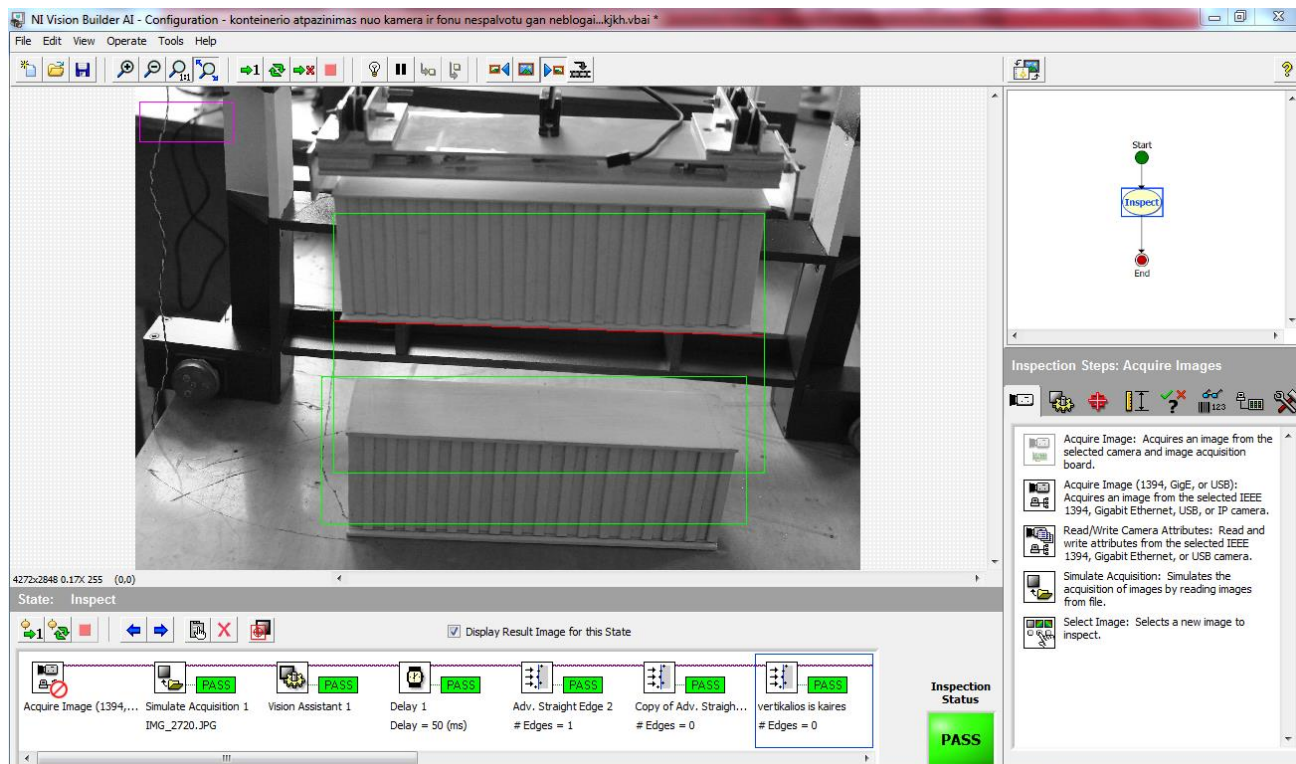
Ni Vision Builder – vaizdo atpažinimo programos kūrimui;

Embarcadero RAD Studio – apmokymo algoritmo programavimui bei neuroninio tinklo apmokymui;

AVR Studio 6 – neuroninio tinklo programavimui bei įdiegimui į valdymo bloką.

2.3.1. NI Vision Builder

Vaizdo atpažinimui ir apdorojimui buvo pasirinkta naudoti kompanijos *National Instruments* siūlomą programinį paketą *NI Vision Builder*. Šis programinis paketas suteikia galimybę apdoroti tiek nuotraukas tiek ir tiesiogiai iš vaizdo kamerų siunčiamą vaizdą. Taip pat turi galimybę tiesiogiai komunikuoti su išoriniais įrenginiais per RS232 sąsają. Lengvas duomenų perdavimas kitiems įrenginiams padeda sukurtą programinį paketą lengvai įdiegti į valdymo sistemas [34].



24 pav. Programinio paketo „NI Vision Builder“ programos darbinis langas

24 paveiksle pateiktas programinio paketo langas susidedantis iš pagrindinių dalių:

Darbinis langas – centre esantis langas kuriame matomas pasirinktinai arba sistemos algoritmo struktūra arba apdorojamo paveikslo vaizdas.

Laukimo langas – dešinėje pusėje viršuje esantis langas, kuriame rodomas laukimo būsenoje esantis arba paveikslo vaizdas arba programos algoritmas. Priklausomai nuo to kuris vaizdas yra įkeltas į darbinę dalį.

Meniu juosta – viršutinėje lango dalyje esanti meniu juosta su skubiųjų funkcijų eilute, kurioje įkelti dažniausiai naudojami įrankiai, tokie kaip išsaugojimo, priartinimo, vykdymo ir kiti įrankiai.

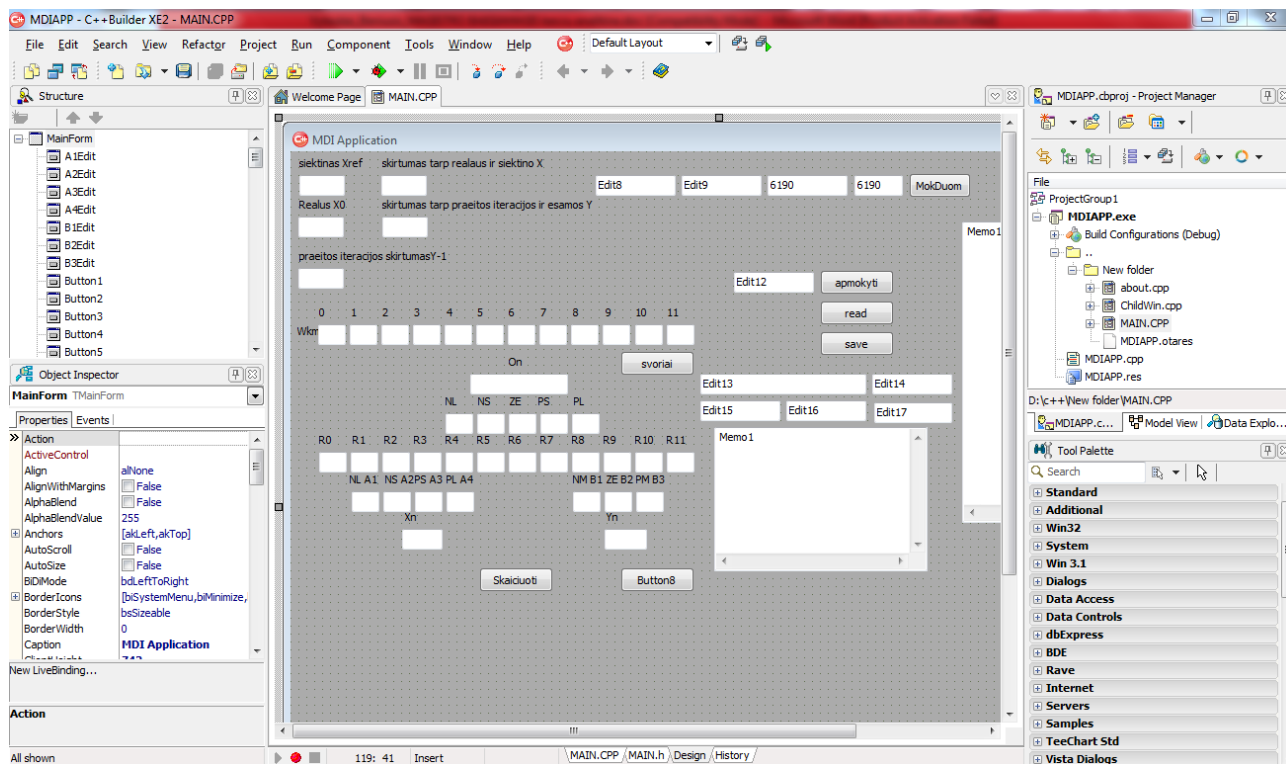
Žingsnių langas – lango apatinėje dešinėje pusėje esantis langas, kuriame pateikti visi programos įrankiai (sąsajos blokai, vaizdo filtravimo funkcijos, defektų aptikimo algoritmai bei kiti).

Būsenos langas – Programos lango apačioje esanti juosta, kurioje vaizduojamos visos vykdomos funkcijos ir koks algoritmas yra vykdomas. Taip pat čia atvaizduojama, koks algoritmas yra įvykdytas teisingai o koks neteisingai.

2.3.2. Embarcadero RAD Studio

Konteinerių valdymo posistemės valdymo bloko programinės įrangos apmokymui bei

testavimui buvo suprojektuota atskira programa. Šios programos rašymui bei kompiliavimui buvo pasirinkta naudoti Embarcadero kompanijos siūlomą produktą „Embarcadero RAD Studio“. Šis produktas suteikia galimybę c++ programavimo kalba kuriamai programai sukurti sąsają tarp žmogaus ir kompiuterio.

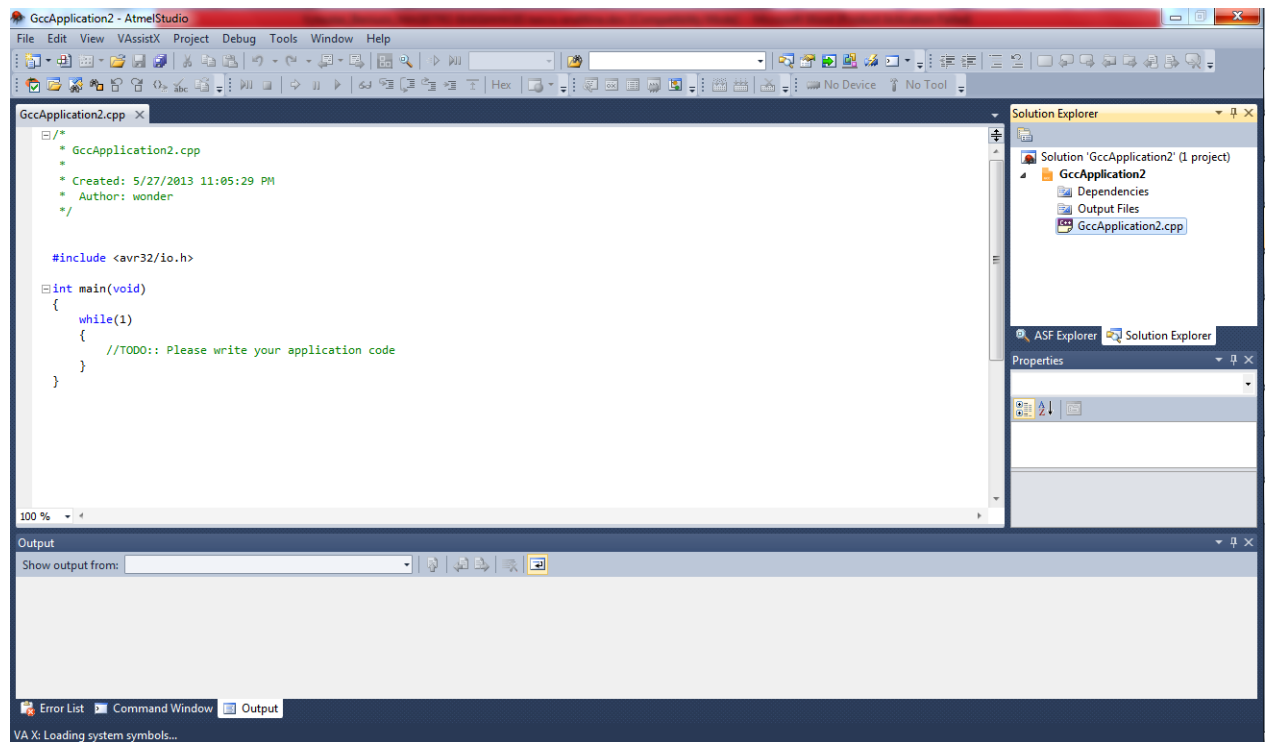


25 pav. Programinio paketo „Embarcadero RAD Studio“ programos darbinis langas

Programos viršuje yra meniu, bei dažniausiai naudojamų funkcijų juosta (25 pav.). Kairėje lango pusėje yra naudojamų objektų(viršuje) ir pasirinkto objekto funkcijų(apacioje) langai. Dešinėje lango pusėje yra programos failų (viršuje) struktūros bei galimų naudoti funkcijų-objektų (apacioje) langai.

2.3.3. AVRStudio 6

Mikrovaldiklio programos kompiliavimui buvo pasirinktas programinis paketas „AvrStudio 6“. Ši programinė įranga programuotojo sukurtą programos kodą paverčia šešioliktainiu mašininu kodu. Pagal standartą šis programinis paketas turi assembler kalbos kompiliatorių, tačiau nuo savo pirmtakų skiriasi tuom, kad savyje turi ir c++ kalbos kompiliatorių (26 pav.). Taip pat šis programinis paketas gali ne tik AVR tipo mikrovaldikliams rašomą kodą bet ir ARM procesoriams [36].



26 pav. Programinio paketo „AVR Studio 6“ programos darbinis langas

Su assembler parašyta programa veikia greičiau ir ekonomiškiau, tačiau c++ yra aukštesnio lygio programinė kalba ir su ja lengviau bei greičiau vyksta programavimas. Pasirenkant programavimo kalbą reikia atsižvelgti į siekiamą programos greitaveiką, sukompiliuoto kodo užimamą vietą mikrovaldiklyje, programavimo laiką bei programavimo patogumą. Nuo visų anksčiau minėtų reikalavimų priklauso kokia programavimo kalba yra parankesnė ir ką reikia pasirinkti. Kuriamos kontenerių valdymo posistemės programos kūrimui buvo pasirinkta c++ kalba, kadangi svarbiau už greitaveiką buvo programavimo patogumas bei laikas, nes pasirinktasis mikrovaldiklis „Atmega 32“ turi pakankamai atminties ir yra labai greitas.

Programinio paketo darbinė aplinką yra paprasta ir patogi dirbti programuotojui. Lango viršuje yra patogiai išdėstyti visi reikalingi įrankiai. Dešinėje pusėje pateikti esamo projekto funkcijų rinkiniai. Taip pat dešinėje lango pusėje išdėstytos visų galimų mikrovaldiklio įėjimo bei išėjimo ir kitų registrų būsenos. Vidurinėje dalyje yra pagrindinis langas, kuriame rašoma kuriama programa. Programinis paketas turi įdiegtą programavimo kalbos stilistiką ir specialiams elementams suteikia atitinkamas spalvas. Tai labai palengvina programos rašymą bei koregavimą. Lango apačioje pateikiamas kompiliavimo rezultatas, atvaizduojamos iškilusios problemos ir klaidos.

Dar viena svarbi šios programos funkcija yra galimybė tiesiogiai sukompiliuotą kodą įkelti į mikrovaldiklį. Sistema palaiko daug programatorių todėl nereikia papildomos programinės įrangos. Techninė įranga

2.3.4. Konteinerių tipai ir specifinės savybės

Pasaulyje naudojama daug įvairiausių konteinerių [34], kurie vienas nuo kito skiriasi savo matmenimis, spalva, svoriu [28]. 27 paveiksle bei 1 lentelėje pateikti įvairių konteinerių pavyzdžiai.



27 pav. Įvairūs konteinerių tipai [28]

Nors konteineriai skiriasi savo matmenimis, tačiau jie turi specifinių bruožų, pagal kuriuos naudojant įvairius vaizdo atpažinimo algoritmus galima juos atpažinti ir išskirti iš aplinkos. Pirmiausia konteineriai yra stačiakampiai gretasieniai. Ilgio ir pločio santykis yra nuo 1:3 iki 1:2 arba nuo 1: 6 iki 1:4. Minėtas ilgio ir pločio santykis padeda atmesti daugumą aplinkos objektų iš bendro vaizdo. Taip pat konteinerių sienos yra gaminamos iš profilinių metalų t.y. turi daug tiesių lygiagrečių linijų, kurios yra stačios viršaus ir apačios linijoms. Taip pat viršutinėje bei apačioje yra konteinerio tvirtinimo skylės ant viršaus ir apačios po 4 visuose kampuose. Visas šias savybes sudėjus, galime atskirti konteinerį nuo kitų aplinkos objektų.

Lentelė 1 Įvairių tipų konteineriai [28]

Tipas	Tara, kg	Kroviny, kg	Ilgis, m	Plotis, m	Aukštis, m
20 Pėdų standartinis	2200	21700	6.06	2.44	2.59
20 pėdų izoliuotas	2450	17760	6.06	2.44	2.44
20 Pėdų platforma	1940	21520	6.06	2.44	0.233
20 pėdų šaldytuvas	3500	24000	5.49	2.25	2.2
40 Pėdų standartas	3980	2650	12.19	2.44	2.59
40 Pėdų šaldytuvas	4350	26300	12.19	2.44	2.28
40 Pėdų High Cube	4150	26300	12.19	2.44	2.90

Iš kelių skirtingų konteinerių tipų nagrinėjimui buvo pasirinktas standartinio 40 pėdų konteinerio 30 kartų sumažinta versija.

2.3.5. Konteineriniai sunkvežimiai ir traukinių vagonai

Konteineriams vežti yra specialios vilkikų priekabos bei traukinių vagonai. Tai kaip ir standartinės priekabos atrodantys tačiau turinčios 4 konteinerių tvirtinimo užraktus 4 priekabos kampuose. Šie spygliai yra pagrindinis bruožas padedantis atpažinti konteinerio priekabą bendrame vaizde.



28 pav. Konteinerinė priekaba [30].

Analogiškus tvirtinimo užraktus turi ir konteineriniai vagonai. Todėl naudojant tuos pačius atpažinimo algoritmus galime atpažinti tiek ir konteinerines sunkvežimio priekabas tiek ir konteinerinius vagonus. Tokio vagono pavyzdys pateiktas 28 paveiksle.



29 pav. Konteinerinis vagonas [31].

2.3.6. Mikrovaldiklis, naudojamas valdymo bloke

Mikrovaldiklis - tai tarsi mažas kompiuteris, savyje turintis procesorių, vidinę atmintį (FLASH), bitinę atmintį (EEPROM) [11]. Vidinė atmintis skirta programuotojo sukurtai programai saugoti, o bitinė – leidžiančia programos vyksmo metu išsaugoti tam tikrus duomenis ir nuskaityti

po to kai buvo pertrauktas sistemos maitinimas. Mikrovaldikliai taip pat turi įvesties bei išvesties kanalus, skaitliukus, bei kitus įrenginius reikalingus įvairioms užduotims atlikti. integruotus viename korpuse. Pagrindinis mikrovaldiklio tikslas yra valdyti tam tikrą sistemą pagal anksčiau programuotojo sukurtą ir įrašytą programą.

Mikroprocesoriaus taktinis dažnis priklauso nuo vidinio arba naudojamo išorinio rezonatoriaus. Rezonatorius pulsacijos sužadina skaitiklius, kurie skaičiuoja impulsų skaičių, kuris yra saugomas skaitliukų bloko laikinoje atmintyje. Persipildžius atminties blokui yra sužadinamas pertraukčių valdymo blokas, kurio formuojamus duomenis apdoroja mikroprocesorius. Lygiagrečios bei nuosekliosios sąsajų blokai gavę naujus duomenis taip pat sužadina pertraukčių bloką ir persiunčia gautus duomenis per duomenų magistralę. Mikroprocesorius visus gautus duomenis apdoroja pagal tai kokia programa yra išsaugota „Flash“ atmintyje ir formuoja specialios paskirties signalus išvesties/įvesties bloke. Atsiradus poreikiui duomenys yra saugomi „EEPROM“ atmintyje, kur informacija išlieka po mikrovaldiklio perkrovimo.

Funkciniu požiūriu, parenkant mikrovaldiklius, reikia atsižvelgti į tokius aspektus :

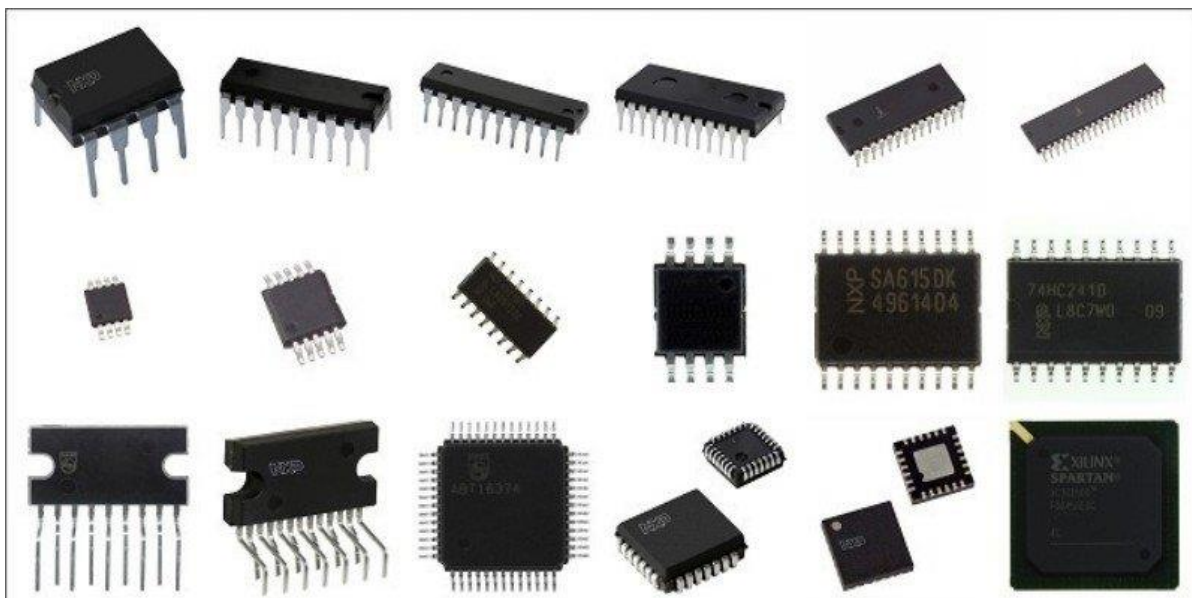
Atminties, skirtos vartotojo programoms ir duomenims saugoti, talpa. Priklausomai nuo mikrovaldiklio tipo ir gamintojo ji gali būti nuo vieno kilobaito iki dešimčių megabaitų

Sistemos maksimalus taktinis dažnis. Skirtingi mikrovaldikliai dirba skirtingu taktiniu dažniu nuo kurio priklauso vieno ciklo vykdymo greitis (vienos komandos atlikimas), todėl šis parametras yra itin svarbus.

Bitinė atmintis (ang. Bit memory) arba vėliavėlės. Tai vartotojui skirta atmintis, į kurią galima įrašyti, saugoti, ir iš kurios galima nuskaityti atskirų bitų vertes. Talpa gali siekti nuo kelių šimtų iki tūkstančių bitų. //parašyti bitų sistemą ten 8 ar 16 ar 32 gražiai

Skaitiklių (ang. Counters) ir laiko daviklių (ang. Timers) skaičių. Šis parametras taip pat yra labai svarbus, kadangi nuo jo priklauso kiek „techninių“ PWM kanalų bus mikrovaldiklyje. Skaitliukų kiekis svyruoja nuo 1 iki 16, priklausomai nuo mikrovaldiklio rūšies.

Yra daug skirtingų rūšių mikrovaldiklių tiekiamų skirtingų gamintojų, tokių kaip Microchip, Atmel, Motorola, Freescale Semiconductor, Analog Devices, Texas Instruments, STMicroelectronics ir daugelis kitų (30 pav). Pagrindiniai skirtumai tarp mikrovaldiklių yra įėjimų bei išėjimų kojų kiekis, maksimalus taktinis dažnis, programinės valdiklio atminties skirtos vartotojo sukurtai programai talpinti. Tai leidžia vartotojui pasirinkti tinkamiausią mikrovaldiklį atitinkantį jo poreikius. Kruopštus pasirinkimas gali žymiai pakeisti galutinę sistemos kainą, kadangi mikrovaldiklių kainos svyruoja nuo kelių litų iki keliasdešimties tūkstančių.



30 pav. Įvairiausių mikrovaldiklių pavyzdžiai [10]

Patys populiariausi ir lengvai konfigūruojami mikrovaldikliai yra tiekiami Microchip ir Atmel gamintojų. Atmel produkcija atlieką viską ką gali atlikti ir Microchip, tačiau greičiau, geriau ir yra pigesni. Todėl buvo nuspręsta rinktis vieną iš Atmel produktų. Mūsų keliamus reikalavimus geriausiai atitiko „AVR“ serijos mikrovaldiklis „Atmega32“ (pakankamai įvesties ir išvesties kanalų, tinkamas greitis, skaitliukų kiekis tinkamas), pagrindiniai parametrai pateikti 2 lentelėje.

Lentelė 2 „Atmega32“ mikrovaldiklio parametrai

Ei Nr.	Parametras	Dydis
1.	„Flash“ atmintis (Kbytes)	32
2.	„EEPROM“ atmintis (Bytes)	1024
3.	Maksimalus taktinis dažnis (MHz)	16
4.	Palaikomos sąsajos	SPI/TWI/UART
5.	Analoginiai įvesties kanalai	8
6.	Skaitliukai	3
7.	Diskretieji įvesties išvesties kanalai	32

Techniniu požiūriu, parenkant mikrovaldiklius, reikia atsižvelgti į tokius aspektus:

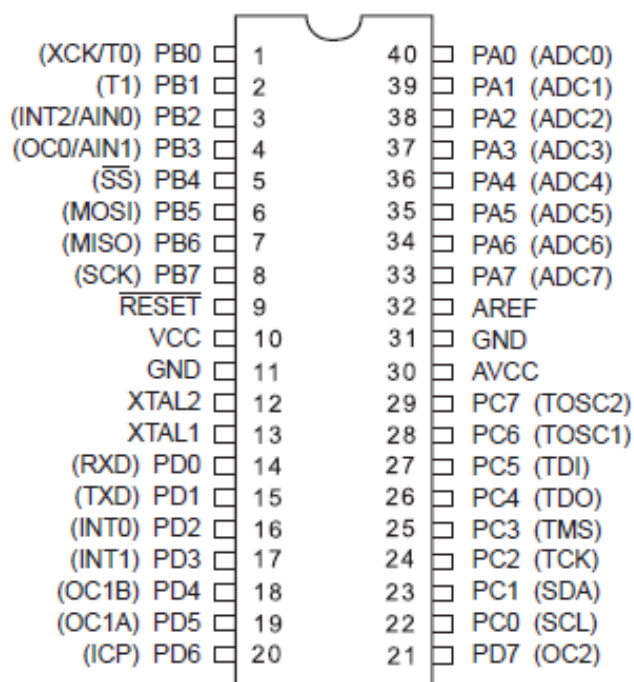
Diskrečiųjų įėjimų ir išėjimų skaičius. Tai mikrovaldiklyje esančių išėjimo/įėjimo kanalų kiekis, kitaip sakant – kontaktų kiekis. Dažniausiai šis skaičius būna nurodomas maksimalus gamintojo integruotas kiekis, t.y. neišplėtus papildomais moduliais. Kanalų kiekis dažniausiai svyruoja nuo kelių vienetų iki kelių šimtų (31 pav).

Analoginių įėjimų skaičius. Kitaip dar vadinamas analoginis-skaitmeninis keitiklis, naudojamas analoginių signalų atvaizdavimui skaitmeniniu formatu. Įėjimų skaičius gali būti nuo kelių iki keliasdešimties.

Ryšio sąsaja (angl. Interface). Skirta komunikavimui su kitais mikrovaldikliais arba duomenų įvesties prietaisais. Priklausomai nuo mikrovaldiklio tipo gali būti naudojamos įvairių rūšių sąsajos: RS232, TWI, JTAG, ISP.

Periferija – tai papildomi moduliai, integruoti į tą patį kristalą. Tai gali būti blokai palaikantys įvairius komunikacijos standartus (RS232, RS485, CAN, USB, I2C), analoginių signalų keitikliai (ASK, SAK), kontroleriai palengvinantys LCD displejų valdymą

Techniniu požiūriu valdiklio tipas ir konfigūracija pasirenkama pagal diskrečiųjų ir analoginių įėjimų bei išėjimų skaičių, greitaveiką, konfigūravimo ir plėtimo bei proceso įrangos apjungimo į komunikacijų tinklą galimybes [10].



31 pav. Pasirinkto mikrovaldiklio išvesties įvesties kanalų reikšmės [11]

Kuriamos konteinerių krovos valdymo sistemos pozicionavimui naudojami keturi nuolatinės srovės varikliai. Šių variklių valdymo schemos naudoja po du diskrečiuosius mikrovaldiklio išėjimus. Visom keturioms schemom valdyti reikalinga 8 išėjimo kanalai. Taigi pasirinktasis mikrovaldiklis turi turėti mažiausiai 8 diskrečiuosius išėjimus. [11]

Modelio lokalizacijai buvo pasirinkta naudoti analoginį signalą išduodančius optinius atstumo jutiklius. Kiekvienas jų naudoja po vieną analoginį mikrovaldiklio įėjimą. Visų jutiklių nuskaitymui

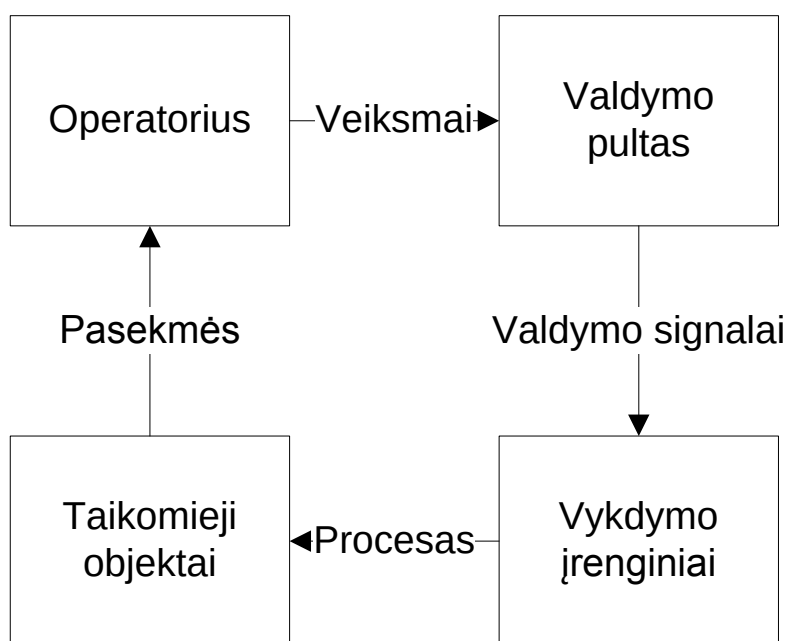
reikia 4 analoginių įėjimų.

Mikrovaldiklis turi būti užprogramuojamas programatoriumi „USBasp“, kuris naudoja ISP sąsają. Testavimui bei rankiniam valdymui per RS232 ryšio sąsają, naudojamas kompiuteris, todėl mikrovaldiklis turi turėti abi minėtų tipų sąsajas.

3. PROJEKTAVIMO DALIS

3.1. Konteinerių kranų krovos valdymo sistema

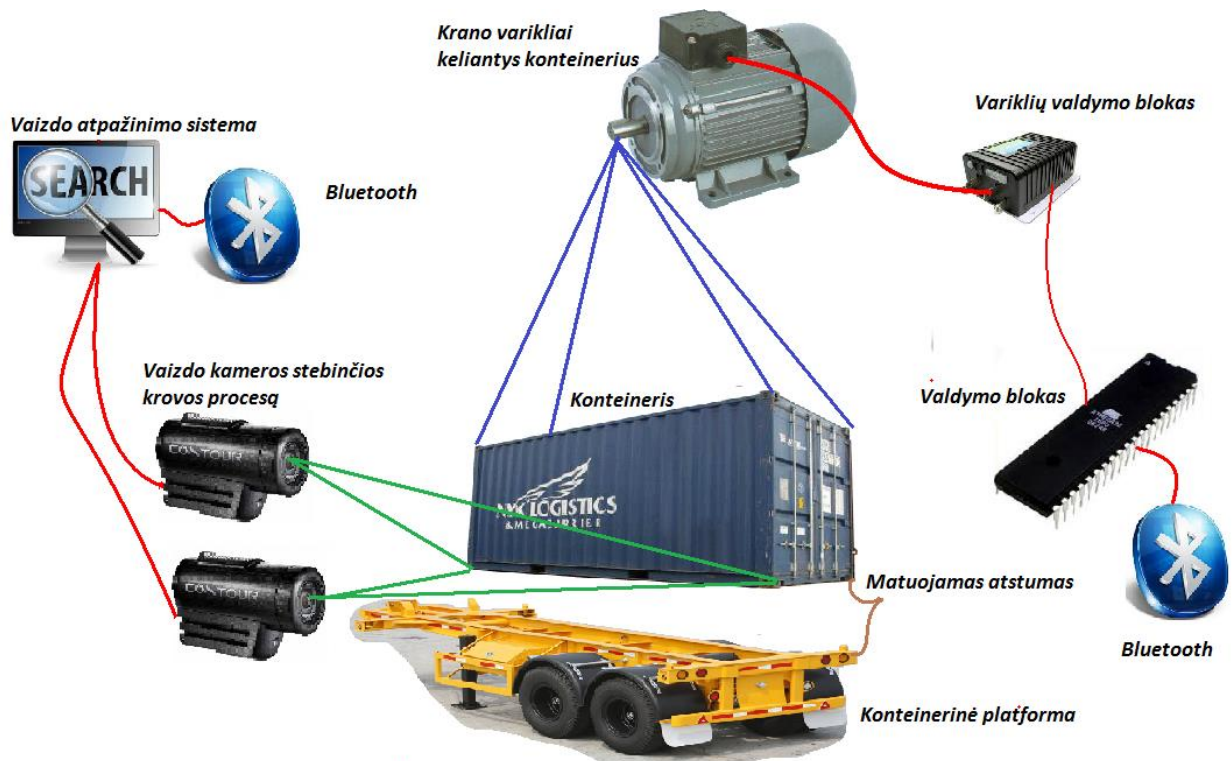
Šiuolaikiniuose įrenginiuose fizinį darbą atlieką varikliai, hidraulika ir kiti vykdymo įtaisai. Visi šie įtaisai vienaip ar kitaip yra valdomi. Daugumoje įrenginių jau įdiegtos automatinio valdymo sistemos, kurios valdo vykdymo įtaisus su minimalia operatorių kontrole. Vis dažniau yra diegiamos ir intelektualios valdymo sistemos, gebančios tiksliau ir efektyviau valdyti įtaisus. Konteinerių krovos kranas, kaip ir kiti įrenginiai turi būti valdomas. Šiomis dienomis valdymą atlieka operatoriai. Į pagalbą pasitelkę valdymo pultą operatoriai valdo variklius valdymo svirčių pagalba, taip transportuodami konteinerius iš vienos iš laivų ant konteinerinių priekabų bei traukinių vagonų. Visą proceso eigą stebi operatorius ir reaguodamas į vykstančius veiksmus valdo vykdymo įtaisus (32 pav).



32 pav. *Intermodalinių konteinerių krovos kranų valdymo diagrama*

Tokia valdymo sistema nėra efektyvi, kadangi visas krovimo procesas priklauso nuo operatoriaus susitelkimo į užduotį. Griebtuvo pozicionavimo tikslumas taip pat priklauso nuo operatoriaus, kadangi valdymo pultas yra toli nuo transportuojamo konteinerio. Todėl operatorius turi akylai stebėti konteinerį ir tuo pačiu metu valdyti variklius. Svarbus momentas yra konteinerio padėjimas ant transportavimo platformos, traukinio vagono ar vilkiko priekabos. Kadangi operatoriaus kabina yra gana toli nuo konteinerio padėjimo vietos, iškyla problema tiksliai padėti konteinerį, nesukeliant didelio triukšmo. Operatoriui praradus budrumą, transportavimas bus

netikslus ir atliekant masiškus transportavimo darbus bus sukiamas labai didelis triukšmas. Ši problema yra labai aktuali Klaipėdos konteinerių krovos terminale „Smeltėje“ todėl būtina sukurti sistemą, kurios valdymas kuo mažiau priklausytų nuo operatoriaus ir leistų tiksliau valdyti krano variklius dedant konteinerį ant platformos ir sukeltų kuo mažesnę triukšmą.

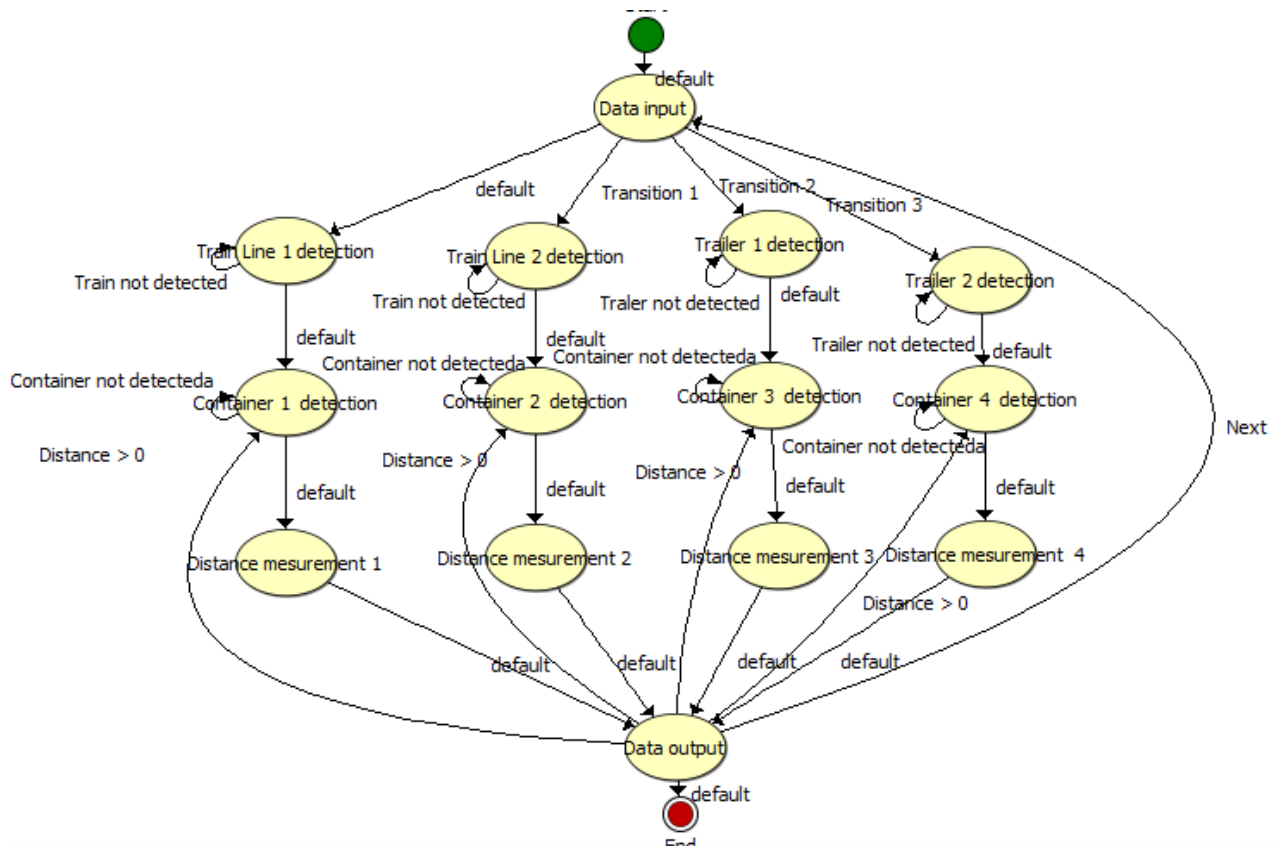


33 pav. *Adaptyvios konteinerių krovos krano valdymo posistemės schema*

33 paveiksle pateikta sukurta Adaptyvi valdymo sistema, paremta automatiniu vaizdo atpažinimo algoritmais. Konteinerio nuleidimo proceso metu konteineris yra leidžiamas lygiagrečiai platformai ant kurios norima padėti konteinerį. Vaizdo kameros nuolat filmuoja krovos procesą ir perduoda vaizdą kompiuteriui, kuris atpažįsta konteinerius bei platformą. Atpažinus konteinerius realiu laiku yra apskaičiuojamas likęs atstumas tarp konteinerio ir platformos ant kurios bus padėtas konteineris. Šie duomenys yra perduodami valdymo blokui per bluetooth sąsają. Valdymo blokas adaptyvaus neuroninio-neraiškaus tinklo pagalba apskaičiuoja kokių greičiu turi sukstis arba sustoti konteinerių krano varikliai ir išduoda atitinkamą signalą variklių valdymo blokui. Šio tinklo pagalba yra tiksliai nuleidžiamas konteineris.

3.2. Vaizdo atpažinimo sistemos projektavimas

Sukurtosios posistemės pirmoji dalis yra vaizdo atpažinimo blokas, kuris susideda iš keturių pagrindinių krypčių, priklausomai nuo to, kurioje linijoje vyksta krovos procesas. Pirmiausia yra gaunama informacija iš valdymo bloko apie krovos liniją. Atitinkamai pasirenkamas Paieškos regionas. Pirmiausia aptinkamas kraunamasis konteineris, konteinerio aptikimui yra naudojami *Sobelio* briaunų aptikimo operatorius ir *Hough* transformacija skirta aptikti konteinerio sienos vertikalioms linijoms bei konteinerio pagrindo linijai. Aptikus konteinerį yra ieškoma platforma/konteineris ant kurio bus dedamas kraunamasis konteineris. Aptikus abu objektus yra išmatuojamas likęs atstumas ir perduodamas valdyymo blokui.



34 pav. Vaizdo atpažinimo programos struktūra

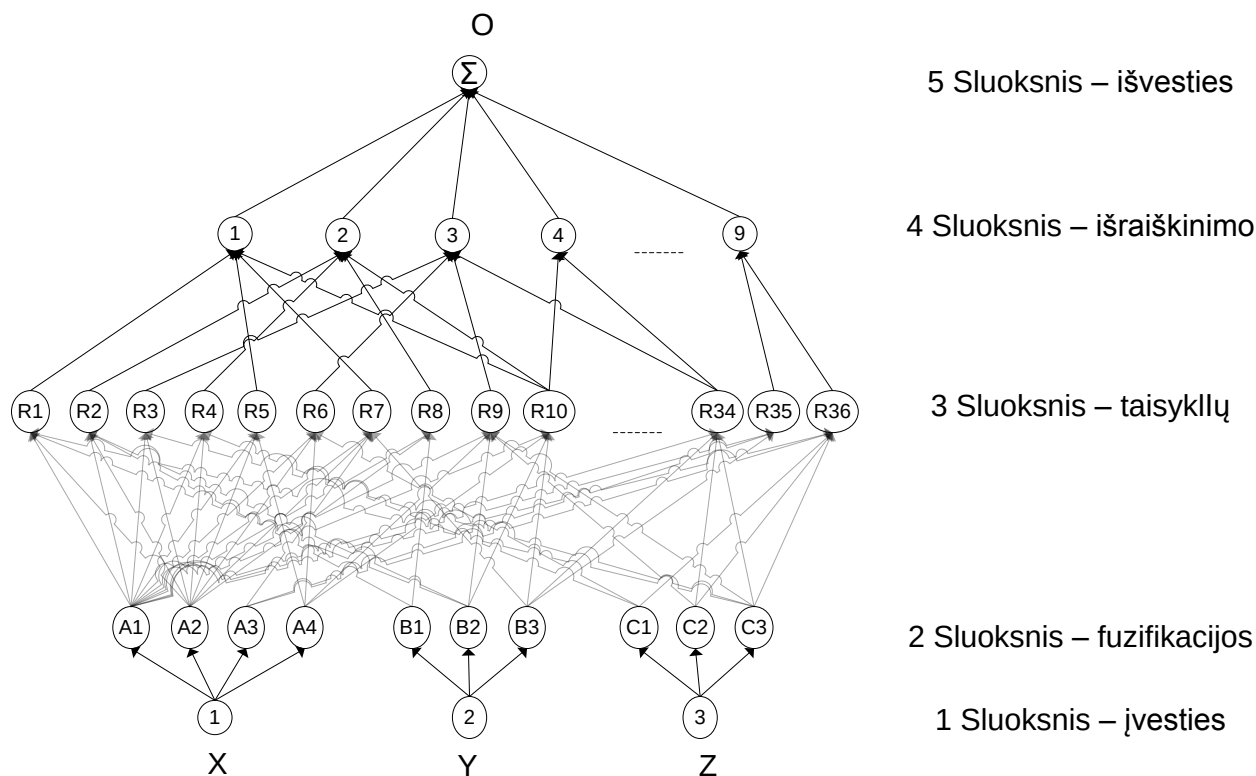
Neatlikus nors vieno veiksmo veiksmas yra kartojamas tol kol atliekamas teisingai ir aptiktas objektas yra priskirtas konteinerio klasei. Programa vykdoma tol, kol likęs atstumas netampa 0. Baigus programos vykdymą yra išsiunčiami duomenys valdyimo blokui apie programos vykdymo pabaigą ir programa pereina į laukimo funkciją kol iš valdymo bloko bus gauti nauji duomenys apie naują krovos procesą.

3.3. Adaptyvios konteinerių krovos valdymo posistemės programinis modelis

Sistema parašyta C++ programavimo kalba, kuri yra c kalbos išplėtimas apibrėžiantis failų rinkinius bei specialių procesorių suteikiančių C kalbai objektinio programavimo galimybes. Programavimo kalba C++ yra priskiriama prie mikrovaldiklių tekstinių programavimo kalbų, kurią reglamentuoja IEC 148822 standartas, kuris pavirtinas 1998 metais. Šia kalba yra aprašomi mikrovaldiklio įėjimai bei išėjimai. Naudojant specialius algoritmus galima aprašyti funkcijas bei jų kintamuosius įtakojančius, svarbius procesus.

Konteinerių krovos valdymo sistemos programinio modelio koncepcija ir struktūra apibrėžia realaus modelio matavimo, registravimo ir vykdymo įtaisus. Kadangi nuo kiekvieno matavimo ir vykdymo įrenginio priklauso sistemos programiniai elementai, būtina apžvelgti kiekvieno iš jų specifikacijas bei programavimo galimybes.

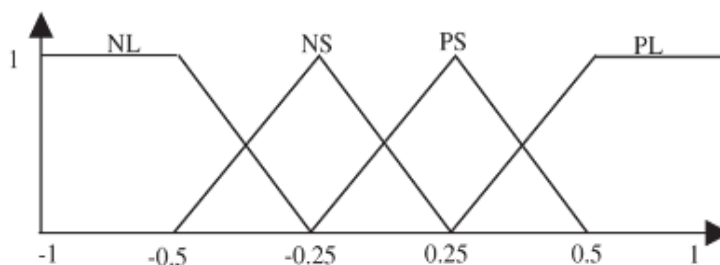
Keturi įėjimo signalai apibrėžia atpažinimo įrenginius bei registruoja jų signalus, kuriuos išsaugo atmintyje matavimų pavidalu. Vėliau gauta informacija apdorojama naudojant pagrindinės programos funkcijas, penkis funkcijų rinkinius ir kintamuosius, kuriuose saugomos atitinkamos reikšmės. Funkcijų rinkiniuose yra saugomos pagalbinės funkcijos, kurios atsižvelgdamos į sistemos kintamuosius apdoroja matavimų rezultatus. Nuo kiekvienos funkcijos priklauso išėjimo signalų būsenos, kurias taip pat gali įtakoti ir kintamieji. Visų šių elementų pagalba yra kontroliuojami devyni išėjimo signalai, valdantys keturis nuolatinės srovės variklius. Atsižvelgiant į sistemos kintamuosius yra formuojami valdymo signalai per išeities kanalus perduodami valdymo įrenginiams.



35 pav. *Prisitaikančios valdymo posistemės neuroninio-neraiškiojo tinklo struktūra*

35 paveiksle pateiktas suprojektuotos posistemės neuroninio-neraiškiojo tinklo struktūra. Pirmasis tinklo sluoksnis dar vadinamas įvesties sluoksnis atlieka perdavimo funkciją. Į tinklą yra paduodami trys išmatuoti dydžiai tai X – variklio sukimosi greičio klaida (skirtumas tarp siektino dydžio ir realaus), Y – klaidos pokytis, skaičiuojama kiek pakito klaida per 1 iteraciją, Z – atstumas tarp konteinerių.

Antrasis sluoksnis vykdo fuzifikacijos funkciją, kurios metu yra gaunami lingvistiniai kintamieji. 36 paveiksle pateiktos narystės funkcijos skirtos lingvistinių kintamųjų gavimui iš greičio klaidos. Trikampės narystės funkcijos pasirinktos dėl skaičiavimų patogumo.



36 pav. *Įvesties narystės funkcijos greičio klaidai X*

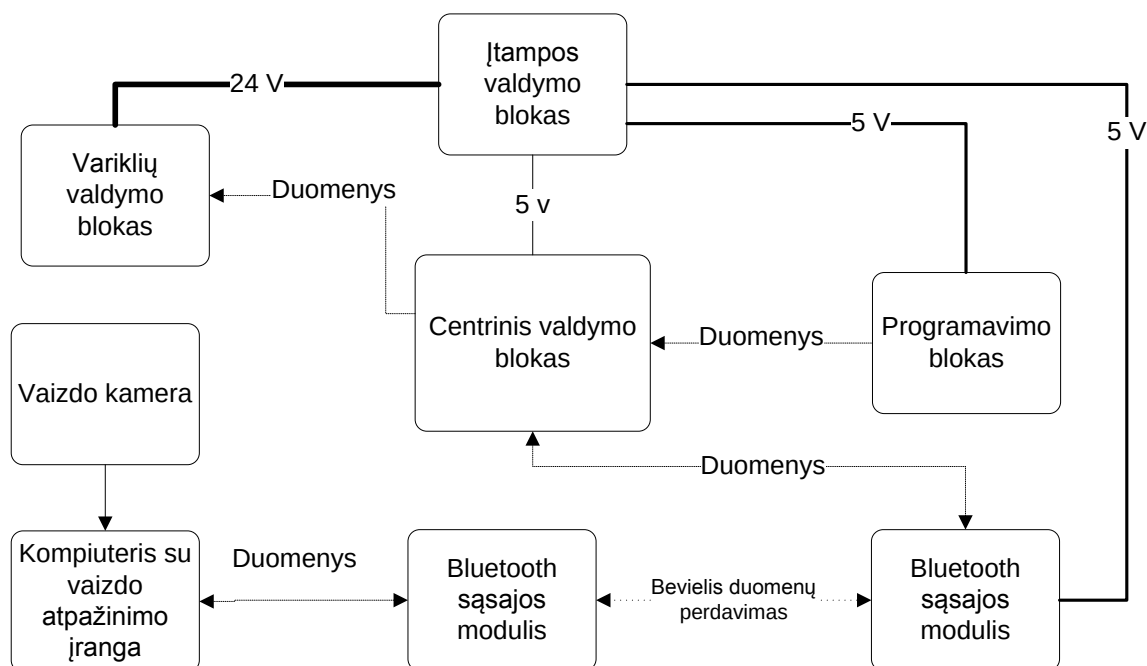
Trečiasis tinklo sluoksnis vadinamas taisyklių sluoksniu, kadangi jame yra atliekami

skaičiavimai nulemiantys išvesties rezultatą. Kiekvienas šio sluoksnio neuronas gaunamas sudauginus tris narius iš antrojo sluoksnio, bei padauginus iš atitinkamo svorio, kuris apmokymo metu yra koreguojamas.

Ketvirtasis sluoksnis gaunamas trečio sluoksnio elementus veikiant atitinkamais svoriais. Iš visų į 4 sluoksnio neuroną perduodamų reikšmių iš 3sluoksnio yra išrenkama maksimali reikšmė ir taip gaunamas 4 sluoksnio neurono išėjimas.

Penktame sluoksnyje yra sumuojami visi 4 sluoksnio išėjimai ir taip gaunama išvestis. Apmokymo metu yra sudaryta apmokymo programa, kuri pagal apmokymo duomenis tikrina kiek skiriasi apmokymo duomenyse nurodytas siektinas išėjimas nuo realaus išduodamo. Gryžtamojo ryšio pagalba yra keičiami svoriai tų neuronų, kurie per 4 sluoksnį pateko į išėjimo sumos narius. Apmokymo greitis ir laikas per kurį bus pasiektas norimas tikslumas yra reguliuojami keičiant apmokymų žingsnį, kuris yra pridedamas prie svorių apmokymo metu. Žingsnio pasirinkimas yra svarbus momentas, kadangi pasirinkus per didelį apmokymo žingsnį galime peršokti per siektiną paklaidą, o sumažinę žingsnį apmokymo procesas gali trukti labai ilgai.

3.4. valdymo sistemos blokų architektūra



37 pav. Konteinerių terminalo valdymo schemos blokinė diagrama

Įtampos valdymo blokas siunčia maitinimo signalą variklių valdymo blokui. Pastarasis aktyvuoja gavęs valdymo signalą iš centrinio valdymo bloko. Analoginių atstumo jutiklių blokas maitinamas 5 V įtampa. Jis išmatuotus duomenis persiunčia centriniam valdymo blokui.

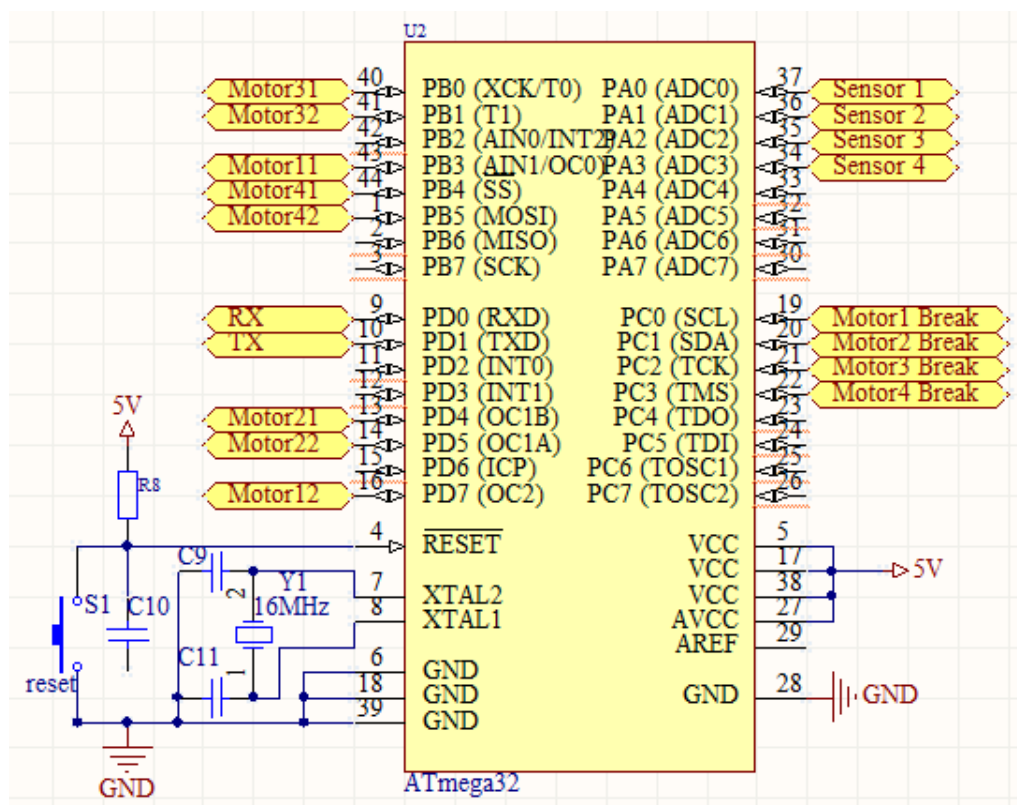
Sąsajos su kompiuteriu blokas gauna 5 V įtampą ir yra aktyvuojamas kompiuterio arba centrinio valdymo bloko, kai yra persiunčiami duomenys. Taip pat šio bloko pagalba yra įjungiamas programavimo blokas, perrašant valdiklio programą. Šiam blokui turi būti tiekama 5 V nominalo srovė.

Centriniam valdymo blokui tiekama 5 V įtampa. Šis blokas valdo apjungia visus kitus valdymo blokus ir atlieka visu reikalingus skaičiavimus.

3.4.1. Centrinis valdymo blokas

Centrinis valdymo blokas ir jo pagrindiniai elementai pavaizduoti 38 paveiksle. Vienas iš svarbiausių jo sudedamųjų komponentų yra mikrovaldiklis (atmega32), atsakingas už visos intermodalinių konteinerių krovos valdymo sistemos veikimą ir kontrolę. Schemoje pavaizduotos iš valdiklio išeinančios duomenų magistralės veda į variklių valdymo schemas bei sensorius. Šios magistralės reikalingos tam, kad mikrovaldiklis galėtų surinkti išmatuotus parametrus ir apdorojės

juos išduoti komandas vykdymo įrenginiams. Sekantis svarbus elementas yra kvarcinis rezonatorius su dviem kondensatoriais leidžiantis mikrovaldikliui dirbti 16 MHz taktiniu dažniu.



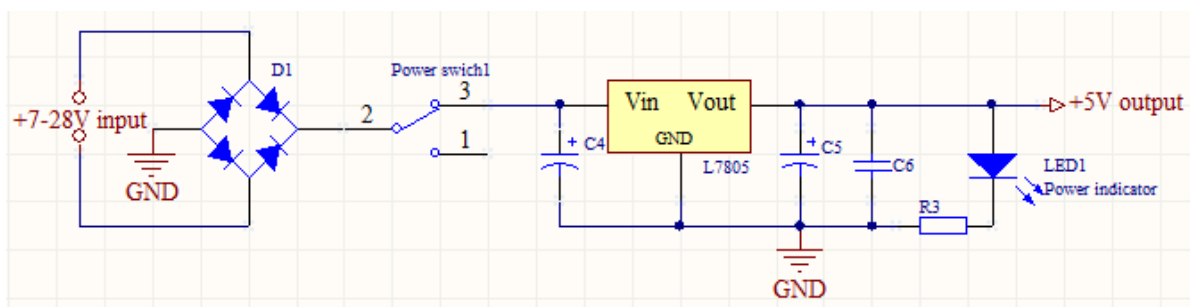
38 pav. Centrinio valdymo bloko elektroninė schema

Paskutinis elementas yra sistemos pakartotinio paleidimo mygtukas. Prie mikrovaldiklio mygtukas prijungiamas prie 5 V įtampos per 10 K Ω varžą ir prie „GND“ per 0.1 μ F kondensatorių. Paspaudus šį mygtuką yra nuvedamas 5 V maitinimas į „GND“ ir vietoje loginio „1“ į „Reset“ koją yra siunčiamas loginis „0“. Loginio signalo pasikeitimas įtakoja išorinės pertraukties vykdymą ir visa mikrovaldiklio programa yra paleidžiama iš pradžių. Ši funkcija naudojama, kai sistema dėl trukdžių išsiderina ir atlieka blogus skaičiavimus.

3.4.2. Sistemos įtampos valdymo blokas

Šis blokas yra atsakingas už tai, kad sistemai būtų tiekiamas tolygi 5V nuolatinės srovės įtampa. D1 diodinis tiltelis apsaugo sistemą nuo galimo atvirkščio poliškumo pajungimo (pajungus mikrovaldiklio maitinimą atvirkščiu poliškumu, mikrovaldiklio kristalas sudega). Bloko maitinimo įtampa gali svyruoti nuo 7 iki 28V, jos konvertavimui į 5V yra naudojamas įtampos reguliatorius – L7805 išduodantis iki 1 A srovę (39 pav.). Mažų svyravimų pašalinimui yra naudojami elektrolitiniai kondensatoriai C4 ir C5, iš mikrovaldiklio sklindantiems svyravimams mažinti naudojamas keramikinis kondensatorius C6. LED1 lemputė yra naudojama srovės indikacijai, įjungus jungiklį lemputė įsižiebia. Lemputės, bei apsaugai nuo galimo trumpo jungimo naudojama

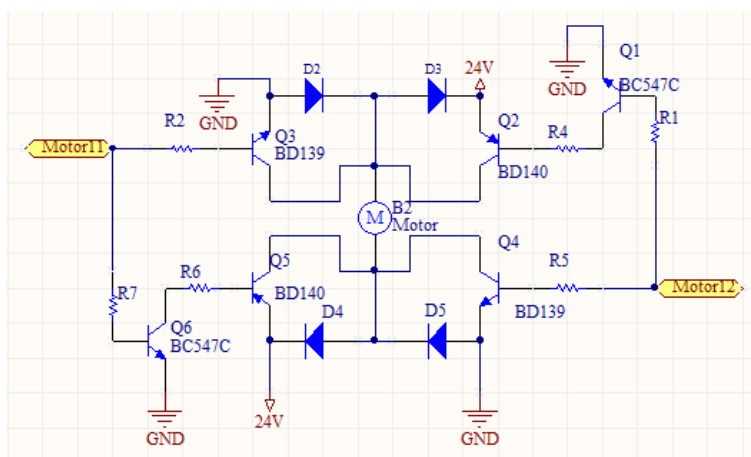
varža R3. Variklių valdymo schemoms maitinimas yra tiekiamas iš kito nuolatinės srovės šaltinio, kadangi varikliams reikalinga didesnė įtampa ir daugiau srovės.



39 pav. Įtampos valdymo blokas

3.4.3. Variklių valdymo blokas

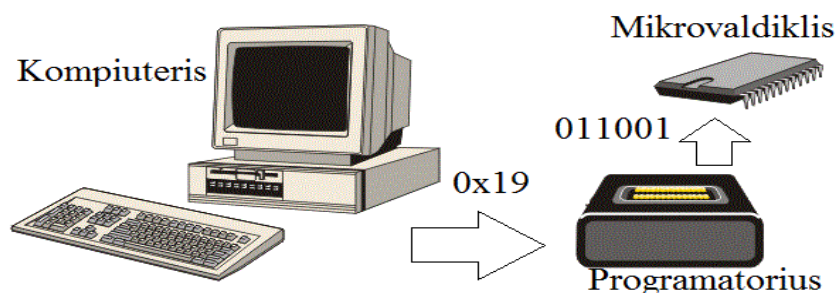
Variklio valdymo blokas yra atsakingas už tai kokių greičių yra sukamas variklis, bei gali valdyti jo sukimosi kryptį. Variklio greitis yra nustatomas į „Motor # PWM“ įėjimo kanalą paduodant atitinkamos trukmės impulsus, kurie atitinkamai atidarinėja tranzistorius ir yra praleidžiama atitinkamo dydžio srovė (40 pav). Todėl valdant pulsuojančiojo valdymo impulso trukmę galime valdyti ir variklio greitį. Krypties keitimui naudojame „H-tilto“ schemą. Schema turi du įėjimo kanalus („Motor1“ ir „Motor2“), valdymo signalą nukreipus į vieną iš jų variklis sukasi viena kryptimi, o nukreipus į kitą kanalą variklis pakeičia sukimosi kryptį. Varžos naudojamos tam, kad nuo per stipraus signalo neperkaistu tranzistoriai. Diodai D2, D3, D4 ir D5 apsaugo schemą nuo atgalinės srovės, kuri atsiranda dėl variklio induktyvumo keičiant variklio kryptį. Standartinė „H-tilto“ schema yra sudaryta iš 4 tranzistorių, čia naudojami du papildomi tranzistoriai (Q1, Q6), kadangi iš mikrovaldiklio paduodamas signalas yra per silpnas valdyti pagrindinius tranzistorius.



40 pav. Variklių valdymo blokas

3.4.4. Programavimo blokas

Sukurtą ir sukompiliuotą programą reikia įkelti į mikrovaldiklį, tam yra naudojamas „AVR“ mikrovaldikliams skirtas programatorius. Iš kelių galimų tipų buvo pasirinktas ir pagamintas laisvai platinamas „USBasp“ programatorius. Prie kompiuterio jungiamas per USB laidą, o prie mikrovaldiklio per ISP 10 kontaktų laidą. Pagrindinė programatoriaus detalė yra mikrovaldiklis „Atmega8“ atliekantis signalų siunčiamų iš kompiuterio konvertavimą (41 pav.) [27].

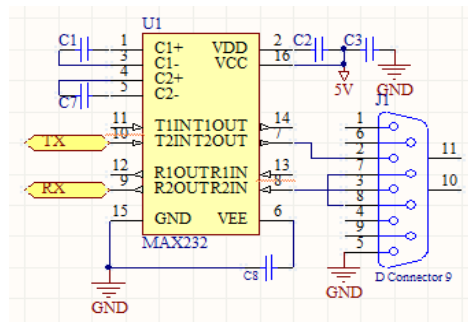


41 pav. *Sąsajos su kompiuteriu blokas*

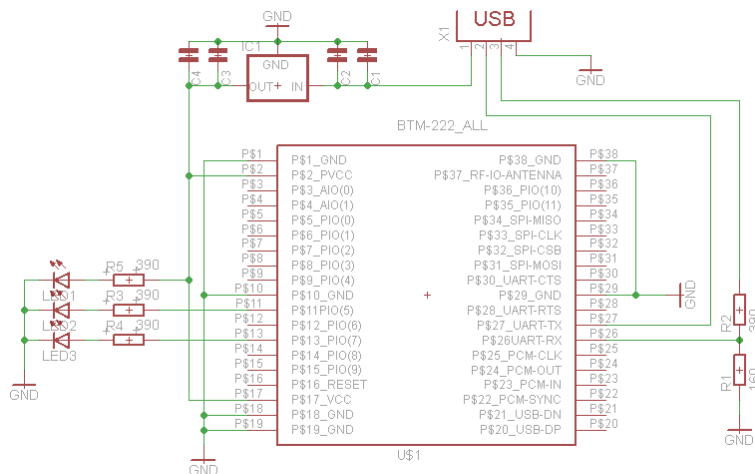
Prie programatoriaus mikrovaldiklio yra prijungta USB jungtis per suderinamas varžas (R1, R2, R3), lizdas ISP kabeliui, trumpikliai (J3 – programavimo dažniui pasirinkti; JP1 – perjungimas į adapterio programavimo režimą; JP2 – gali būti naudojamas programuojamo mikrovaldiklio maitinimui) ir du šviesos diodai su apsauginėmis varžomis (R4 ir R5), vienas maitinimo indikacijai (D1), o kitas programavimo aktyvumui stebėti (D2).

3.4.5. Sąsajos su kompiuteriu blokas

Šio bloko pagrindinė paskirtis – sujungti kompiuterį ir mikrovaldiklį tarpusavyje, kad būtų galima perduoti duomenis vaizdo nuskaitymo programai esančiai kompiuteryje, bei gauti iš jos duomenis. Valdiklis apdorojęs iš jutiklių gautus duomenis perduoda kompiuteriui, kurioje padėtyje yra griebtuvas. Taip vaizdo atpažinimo sistema gali pasirinkti vieną iš keturių regionų kuriame atlikti konteinerio ir konteinerio platformos padėties. Atitinkamai kompiuteris perduoda valdymo sistemai išmatuotą atstumą tarp konteinerio ir konteinerio platformos. Be sąsajos su kompiuteriu bloko sistemos veikimas būtų neįmanomas. Kompiuteris su mikrovaldikliu bendrauja RS232 protokolu, tačiau kompiuteris perduoda nuoseklius signalus, o mikrovaldiklis lygiagrečius TTL signalus. Signalų konvertavimui naudojama įterptinė mikroschema „MAX232“ (42 pav). Schemoje naudojami kondensatoriai C1 ir C7 naudojami mikroschemos konfigūravimui, o kondensatoriai C2 ir C3 iš šaltinio sklindančių impulsų surinkimui [18].



Kad duomenis gautume perduoti didesniu atstumu ir sistemos įdiegimo supaprastinimui, buvo suprojektuotas bevielės sąsajos blokas (43 pav.). Šis blokas veikia bluetooth „BTM222“ modulio pagrindu. Kadangi valdymo blokas dirba 5V režime, o bluetooth modulis veikia 3,5 režimu [26], schemoje naudojamas įtampos reguliatorius schemoje pažymėtas IC1 numeriu, įtampos išlyginimui naudojami kondensatoriai C1-C4.

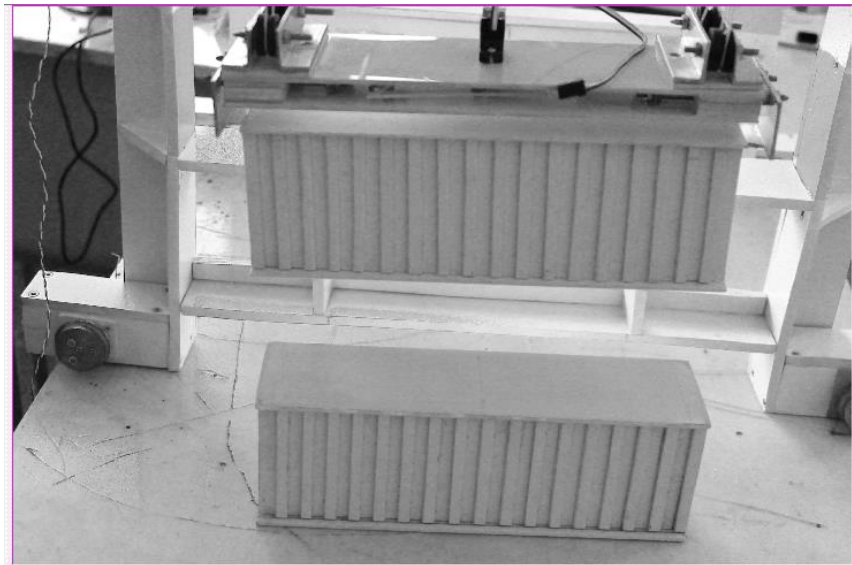


Kompiuterio pusėje atitinkamai yra naudojamas „Bluetooth“ adapteris, kompiuteryje sukuriantis virtualų COM uostą. Šio adapterio valdymui naudojama „BlueSoel“ programinė įranga. Sistemos saugumui susijungimo metu yra naudojamas slaptažodis.

4. EKSPERIMENTINĖ DALIS

4.1. Konteinerio aptikimas

Pirmasis procesas, kurį atlieka sukurtoji posistemė tai konteinerio atpažinimas. Kokybiškesniam vaizdui išgauti naudojame funkcinį bloką „delay“ su 20 milisekundžių užlaikymu. Naudojami algoritmai apdoroja nespaltvotą vaizdą, todėl „Vision Assistant“ funkcijų bloko pagalba vaizdas buvo konvertuojamas į nespaltvotą vaizdą (44 pav.).



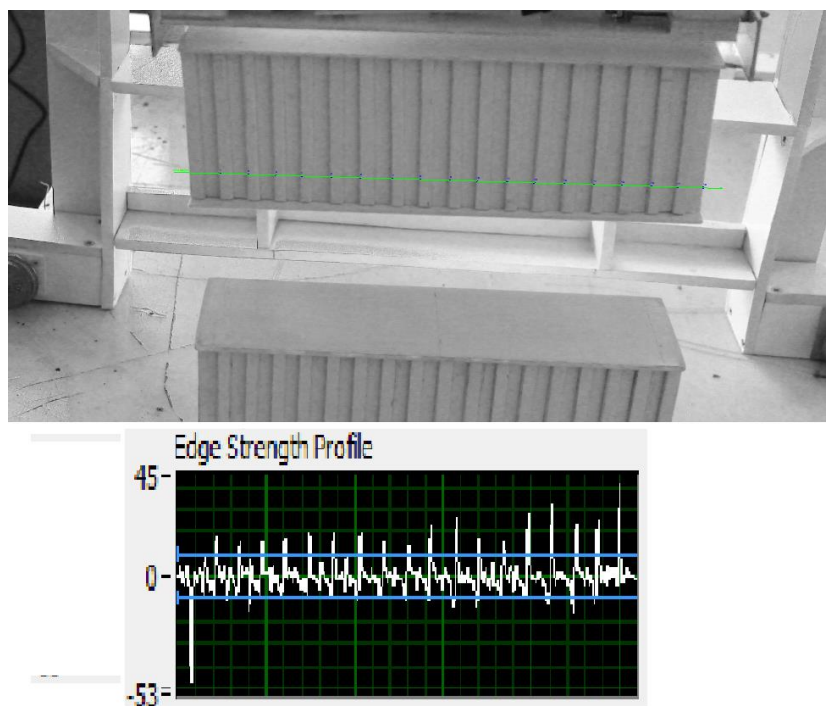
44 pav. Vaizdas konvertuotas į nespaltvotą vaizdą

Nespaltvotas vaizdas filtruojamas naudojantis Sobel briaunų atpažinimo operatoriumi. Filtracijos rezultatai pateikti 45 paveiksle. Balta spalva žymimos nekontūrinės zonos, O juoda spalva pavaizduotos briaunos ir konteinerio kontūrai. Kadangi pats Sobelio briaunų aptikimo operatorius negali nustatyti, kurios briaunos yra konteinerio, kurios ne atliekame geometrinių parametrų paiešką.



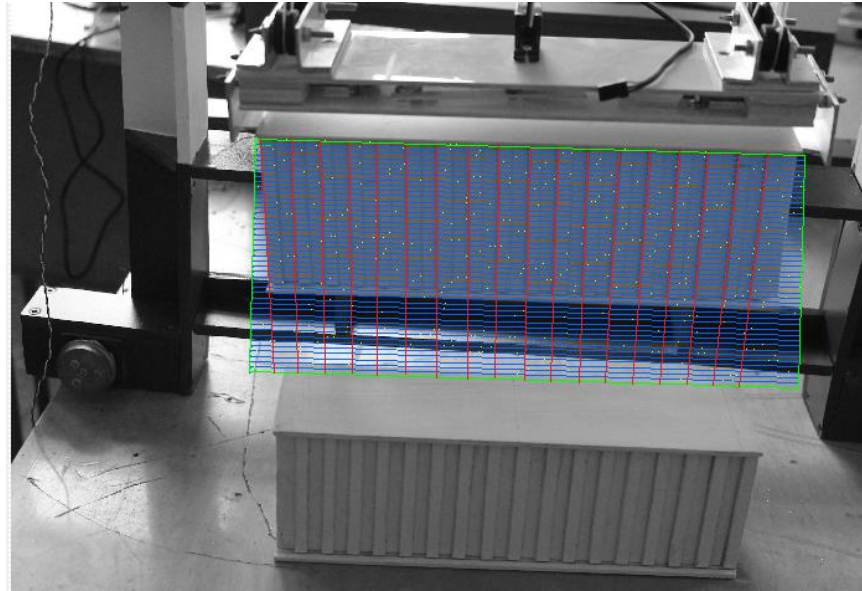
45 pav. *Vaizdas gautas po Sobel briaunų paieškos filtravimo*

Nespalvotame vaizde naudodamiesi „Find edges“ funkciniu bloku surandame zoną kurioje periodiškai kartojasi didesnio intensyvumo taškai. Šis blokas matuoja kelių iš eilės sekančių taškų intensyvumą ir išrenka pasirinktinai arba šviesesnes arba tamsesnes viršūnes. Tokiu būdu surandame konteinerio sienų iškilimus. 46 paveiksle pateikiamas tokios filtracijos rezultatas, kuriame matomas vaizdo viršūnių apšviestumo grafikas – tose vietose kuriose apšviestumas stipriai skiriasi nuo šalia esančių taškų grafike matomas pakilimas arba kritimas.



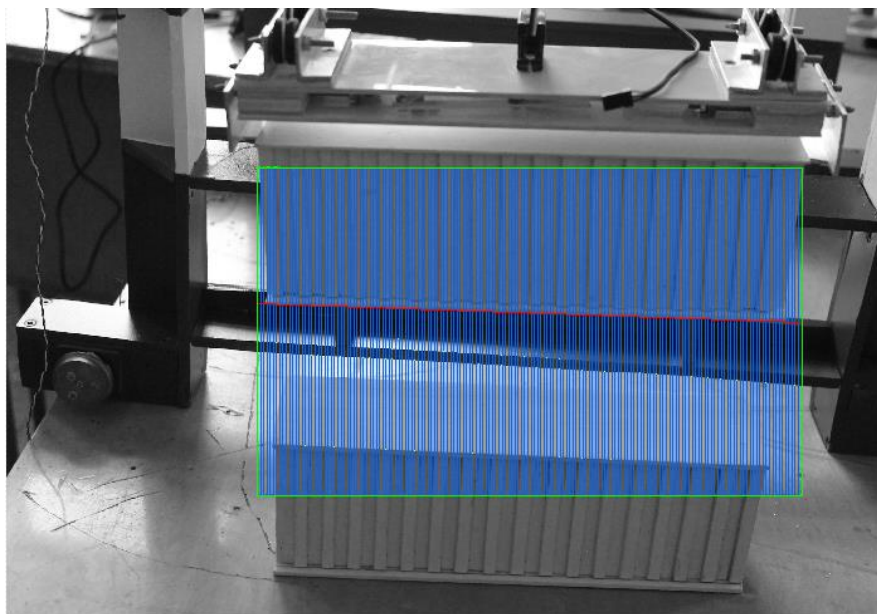
46 pav. *Konteinerio sienos iškilimų radimas*

Per vaizdą brėžiant daug tokių horizontalių linijų gaunama taškų aibė per kuriuos galime išbrėžti tiesias linijas nurodančias konteinerio sienos iškilimų kontūrus. Tokių tiesių brėžimui naudojame *Hough* transformaciją. Atitinkamai nustatome galimus linijų pakrypimo kampus, kadangi vaizdo kamera filmuoja ne viename aukštyje su konteineriu, o yra šiek tiek aukščiau. Dėl šios priežasties atstumas tarp sienos iškilimų konteinerio viršuje gaunamas didesnis nei konteinerio apačioje.(47 pav).



47 pav. *Hough transformacija konteinerio šono linijų suradimui*

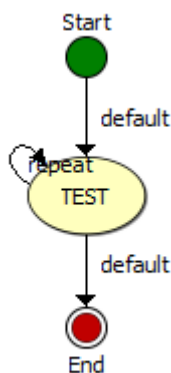
Analogiškus veiksmus atliekame ir konteinerio apatinės briaunos suradimui, tik skirtingai nei prieš tai atliktame žingsnyje naudojame vertikalų filtrą taškų suradimui ir horizontalių linijų brėžimą (*Hough* transformacija)(48 pav). Vertikalių tiesių susikirtimo taškai su horizontaliomis tiesėmis nustato konteinerio apatinės briaunos ilgį. Analogiški veiksmai atliekami apatinio konteinerio aptikimui.



48 pav. *Hough transformacija konteinerio pagrindo suradimui*

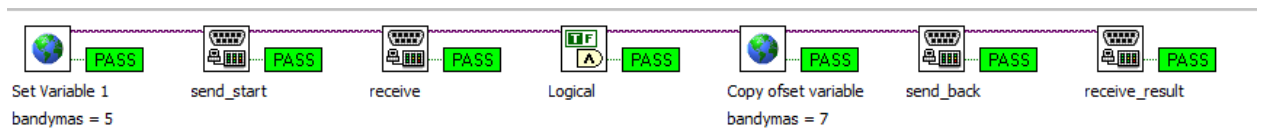
4.2. Bevielės Bluetooth ryšys

Kad suprojektuotą posistemę būtų lengviau įdiegti į konteinerių krovos kraną, naudojame bluetooth sąsają. Bevielės sąsajos testavimui buvo parašytos dvi programos. Pirmoji kompiuterinė programa veikianti NI Vision Builder aplinkoje. Šios programos algoritmas pateiktas 49 paveiksle.



49 pav. *Bevielės sąsajos testavimo NI Vision builder aplinkoje algoritmas*

Kadangi ši programa buvo projektuojama bevielės sąsajos testavimui, tai ji turi tik vieną patikrinimo skiltį „TEST“. Šios skilties programinis kodas pateiktas 50 paveiksle pagrindiniai naudojami funkciniai blokai yra „Serial I/O“ siunčiantis ir gaunantis duomenis į COM uostą, kintamojo sukūrimo bei keitimo blokas, bei loginės operacijos blokas, tikrinantis ar teisingi gauti duomenys.



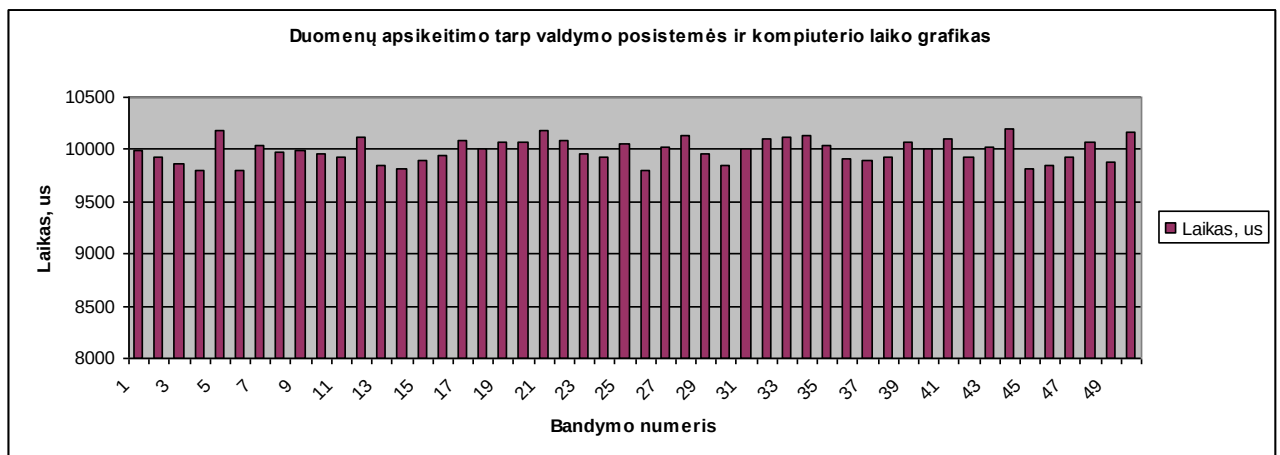
50 pav. Bevielės sąsajos testavimo NI Vision builded aplinkoje grafinis programos kodas

Valdymo bloko programinio kodo pagrindinio ciklo fragmentas pateiktas sekančiai:

```
while (1)
{
    while(test != '5') test = USART_Receive();
    TIMER_count=0
    USART_Transmit('6');
    while(test != 7) test = USART_Receive();
    USART_Transmit(TIMER_count);
}
```

Mikrovaldiklis laukia kol bus gaunamas starto ženklas '5' tada įjungia laikmatį, ir išsiunčia patvirtinimo užklausą kompiuteriui. Kompiuteris gavęs tokią užklausą siunčia patvirtinimo simbolį '7'. Valdymo blokas gavęs teisingą patvirtinimo ženklą stabdo laikmatį ir išmatuotą laiką perduoda kompiuteriui.

Duomenų apsikeitimo laiko matavimo eksperimentas buvo kartojamas 50 kartų. Ir gauti rezultatai apibendrinti 51 paveiksle.



51 pav. Duomenų apsikeitimo tarp valdymo posistemės ir kompiuterio laiko grafikas



52 pav. *Bluetooth sąsajos blokas*

Iš gautų rezultatų matome, kad duomenų apsikeitimo vidutinis laikas yra 10 milisekundžių, o tai tenkina iškeltus bevielės sąsajos reikalavimus.

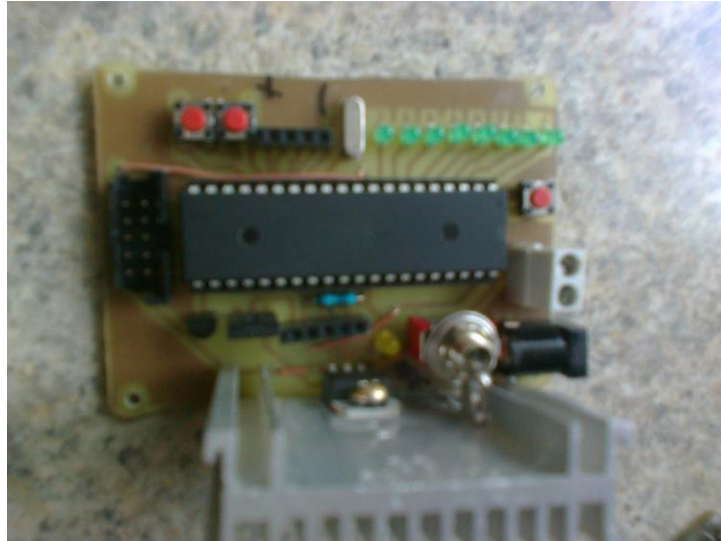
4.3. Adaptyvios valdymo posistemės testavimas

Neuroninio-neraiškaus tinklo testavimui buvo parašyta programa, kurios programinis kodas pateiktas prieduose. Toliau pateikiamas neuroninio tinklo apmokymo fragmentas, kurio metu yra atliekami skaičiavimai reikalingi norint koreguoti tinklo svorius:

```
for(int m = 0; m<9; m++){  
    double skait = 0;  
    for(int j = 0; j<9; j++){  
        //sumuojame 5 sluoksnio  
        elementu sandauga  
        if(j==m) continue;  
        skait += O[j]*(b[7+m]-b[7+j]); //sandauga is m elemento  
        centro atimant  
    }  
    //j elementus  
    f[m]=skait/vard; //m elemento išvestine  
    delta[m] = E0*f[m]; //m elemento delta  
    dWkm=ni*delta[m]*R[Ok[m]]; //paskaičiuojame svorių
```

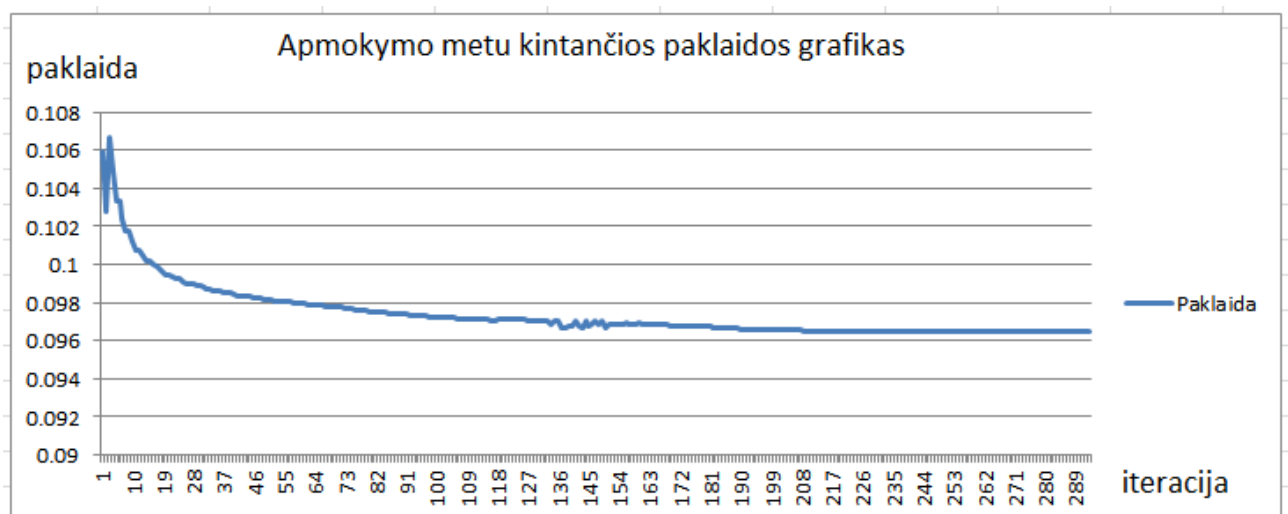
pokyčius

```
if ( (Wkm[Ok[m]] + dWkm) < 0.001 ) Wkm[Ok[m]] = 0.001;  
else Wkm[Ok[m]] += dWkm; //koreguojame svorius  
}
```



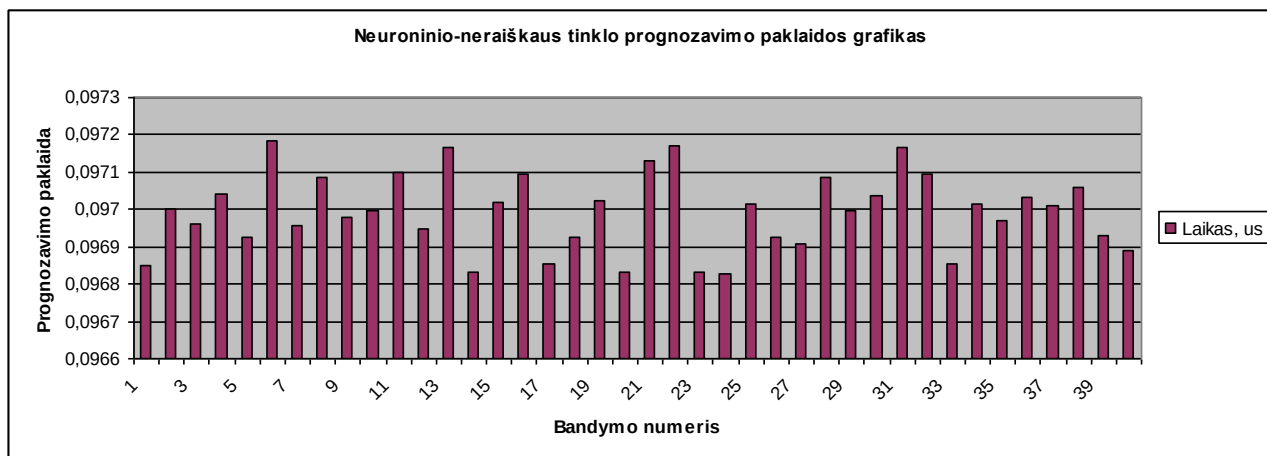
53 pav. Centrinis valdymo blokas

Kiekvienos iteracijos metu yra apskaičiuojama gauta paklaida – skirtumas tarp siektino išėjimo ir realaus tinklo išvesties rezultato. Norint išvengti melagingos paklaidos, dėl galimų teigiamų ir neigiamų reikšmių, yra skaičiuojama kvadratinė paklaida. Apmokymo tikslas yra pasiekti suminės kvadratinės paklaidos per visus apmokymo duomenis reikšmę mažesnę nei 0.01. 289 iteracijų apmokymo rezultatai pateikti 54 paveiksle. Iš gautų duomenų matome, kad neuroninis tinklas apmokytas iki siektino dydžio ir gauta 0.097 paklaida, kas tenkina užsibrėžtą tikslą.



54 pav. Neuroninio-neraiškaus tinklo apmokymo rezultatų grafikas

Keičiant apmokymo duomenų įmtį buvo nustatyta, kad neuroninio-neraiškaus tinklo prognozavimo paklaida kinta nuo 0,0968 iki 0,0972. Pasiektos prognozavimo paklaidos kitimo grafikas pavaizduotas 55 pav. Galutinė neuroninio tinklo prognozavimo paklaida $0,097 \pm 0,002$.



55 pav. Neuroninio-neraiškaus tinklo prognozavimo paklaidos

IŠVADOS

1. Atlikus nagrinėjamos srities literatūros analizę buvo nustatyta, kad dauguma mokslininkų atlieka mokslinius tyrimus, kurie yra susiję su konteinerių krovos papročiais susijusiais su konteinerių pagriebimu, griebtuvo su konteineriu svyravimų krovos metu apskaičiavimu, bei konteinerio griebtuvo padėties nustatymu. Nustatyta, kad konteinerių aptikimui geriausia naudoti *Sobelio* briaunų radimo operatoriaus bei *Hough* transformacijos vaizdo atpažinimo algoritmus, o valdymo posistemės variklių valdymui tinkamiausias adaptyvus neuroninio-neraiškaus tinklo algoritmas.
2. Remiantis minėtais algoritmais buvo suprojektuota adaptyvi konteinerių krovos valdymo posistemė, pagamintos reikalingos schemos ir sukurtas programinis kodas.
3. Vaizdo atpažinimo algoritmų testavimas parodė, kad taikomi algoritmai atpažįsta konteinerius ir išmatuoja atstumą tarp transportuojamo konteinerio ir konteinerio ant kurio norima padėti konteinerį. Atlikus bevielės sąsajos duomenų apsikeitimo tyrimą, eksperimentiškai nustatyta, kad sukurta valdymo posistemė matavimų duomenimis apsieičia per 10 milisekundžių, toks uždelsimas yra pakankamai mažas ir nedaro įtakos posistemės valdymui. Taip pat atlikus adaptyvios valdymo posistemės apmokymą pasiekta $0.097 \pm 0,002$ prognozavimo paklaida, kuri leidžia suprojektuotą neuronini-neraiškųjį tinklą naudoti konteinerių krovos papročio valdyje.

TOBULINIMO PERSPEKTYVOS

Ateities tyrimuose planuojama diegti garso jutiklį, bei naudoti skatinamojo mokymosi algoritmus, leidžiančius realiu laiku posistemei pačiai apsimokyti. Garso jutiklis pasitarnautų, nustatinėjant paskatą arba baudą priklausomai koks triukšmas sukeliamas konteinerio padėjimo metu. Tokią sistemą reikalinga testuoti realiame konteinerių terminale, kadangi tokių terminalų modeliuose, detalės gaminamos iš plastiko ir nesukelia analogiško triukšmo, koks yra sukeliamas tikruose konteinerių terminaluose.

LITERATŪRA

1. Hee-Joo Yoon, Young-Chull Hwang and Eui-Young Cha, 2010. Real-time container position estimation method using stereo vision for container auto-landing system, International Conference on Control, Automation and Systems (872 – 876).
2. Wei Wu, Zheng Liu,, Mo Chen, Xiaomin Yang, Xiaohai He, 2012. An automated vision system for container-code recognition Expert Systems with Applications, 2842–2855.
3. Deb, K., Kang, S.-J., & Jo, K.-H., 2009. Statistical characteristics in HSI color model and position histogram based vehicle license plate detection. Intelligent Service Robotics, 2(3), 173–186.
4. T. D.Duan, T. L. H.Du, T. V.Phuoc, & N. V.Hoang, 2005. Building an automatic vehicle license plate recognition system. In Proceedings of international conference on computer science (pp. 59–63).
5. N.Otsu, 1979. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics, 9(1), 62–66.
6. H. Kawai, Y. Choi, Y. B. Kim and Y. Kubota, 2008. Position Measurement of Container Spreader Using an Image Sensor System for Anti-sway Controllers. International Conference on Control, Automation and Systems (683 – 686).
7. Rafael C. Gonzalez, Richard E. Woods, 2008. Digital Image Processing, Pearson Prentice Hall.
8. Sobelio briaunų aptikimo operatoriaus veikimas ir pritaikymas [interaktyvus] [žiūrėta 2013-03-15] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>.
9. Hough transform [interaktyvus] [žiūrėta 2013-01-18] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>.
10. Do-Hyeon Kim, Eui-Young Cha, 2008. License Plate Extraction Using Gray Labeling and Fuzzy Membership Function, Korea Institute of Marine Information & Communication Sciences, Vol.12, No.8, pp.1495-1504.
11. Panašieji trikampiai [interaktyvus] [žiūrėta 2013-03-20] <http://752209539946970205.weebly.com/8/post/2012/09/panaieji-trikampiai.html>.
12. Li Wei, EungJoo Lee, 2009. Real-Time Container Shape and Range Recognition for Implementation of Container Auto-Landing System, Journal of Korea multimedia Society, Vol. 12, No. 6, pp. 794-803.
13. K.H. Kim, S.H. Won, J.K. Lim, T. Takahashi, 2004. An architectural design of control software for automated container terminals, Computers & Industrial Engineering, Vol.46, pp.741-754.

14. Hee-Joo Yoon, Eui-Young Cha, 2009. Weighted cost aggregation approach for depth extraction of stereo images,” Korea Institute of Marine Information & Communication Sciences, Vol.13, No.6.
15. S. Kunimitsu, 2001. Measurement of Relative Position of Container with Image Processing for Automatic Container Cranes, Trans. IEE Japan, Vol. 121-C, No. 5, pp. 882-891.
16. Ahmed Rubai, Abdul R. Ofoli, Legand Burge and Moses Garuba, 2005. Hardware Implementation of an Adaptive Network-Based Fuzzy Controller for DC-DC Converters, IEEE transactions on industry Applications, VOL. 41, NO. 6 pp. 1557 – 1565.
17. M.Sedighizadel and A.Razazadeh, 2008. Adaptive PID Controller based on Reinforcement Learning for Wind Turbine Control, World Academy of Science, Engineering and Technology, pp 257 – 262.
18. F.D.Kanellos, N.D.Hatziargyriou, 2002. A new control scheme for variable speed wind turbine using neural networks. IEEE Power Engineering Society Winter Meeting.
19. F.You-tong, F.Cheng-zhi, 2007. Single neuron network PI control of high reliability linear induction motor for Maglev. Journal of Zhejiang University SCIENCE A, pp. 408-411.
20. Y.Zhao-da, Z.Chong-guang, S.Shi-chuan, L.Zhen-tao, W.Xi-zhen, 2003. Application of neural network in the study of combustion rate of natural gas/diesel dual fuel engine. Journal of Zhejiang University SCIENCE A, pp. 170-174.
21. Chen G., Tat Pham T. Introduction to Fuzzy Sets, Fuzzy logic, and Fuzzy Control Systems (Hardcover). 200. P.328.
22. Medvedev V. Tiesioginio sklaidimo neuroninių tinklų taikymo daugiamačiams duomenims vizualizuoti tyrimai, Vilniaus Gedimino technikos universitetas matematikos ir informatikos institutas, daktaro disertacija technologijos mokslai, informatikos inžinerija (07t).
23. P.Rani, 2003. Human robot collaboration based on affect sensing. For the Degree of Master of Science.
24. G.DZEMYDA, O.KURASOVA, J.ŽILINSKAS, 2010. Daugiamačių duomenų vizualizavimo metodai, matematikos ir informatikos institutas. Vadovėlis informatikos krypties doktorantams ir magistrantams.
25. K.S.Calderon, W.W.Thompson, 2004. Biofeedback relaxation training: A rediscovered Mind-Body tool in Public Health // American Journal of Health Studies. vol. 19(4), p. 185–194.
26. BTM222 module datasheet [interaktyvus] [žiūrėta 2012-05-05] http://www.mikrocontroller.net/wikifiles/f/fc/BTM222_DataSheet.pdf.
27. USB Asp programmer [interaktyvus] [žiūrėta 2012-01-15] <http://www.fischl.de/>.
28. Konteineriai ir jų matmenys [interaktyvus] [žiūrėta 2012-05-05] <http://www.grandservice.lt/containers-dimensions>.

29. R. Silipo, 2003. Neural networks. In: M. Berthold, D. J. Hand (Eds.), Intelligent Data Analysis: an Introduction. Springer-Verlag, pp. 269–320.
30. Konteinerių priekabos [interaktyvus] [žiūrėta 2013-02-06]
<http://www.sinotrailers.com/container-chassis-trailers.htm>.
31. Konteinerių vagonas [interaktyvus] [žiūrėta 2013-02-06]
<http://www.ktg.bg/en/wagon1.html>.
32. F. P. Branca, S. A. Sciuto, A. Scorza, 2012. Comparative evaluation of ultrasound scanner accuracy in distance measurement, Review of Scientific Instruments.
33. M.Georgijevic, N.Zrinc, 2010. Container Terminals in River Ports, FME Transactions, pp 199-204.
34. National Instruments oficialus tinklapis [interaktyvus] [žiūrėta 2013-02-01]
<http://www.ni.com/vision/vbai.htm>.
35. Embarcadero oficialus tinklapis [interaktyvus] [žiūrėta 2013-01-04]
<http://www.embarcadero.com/>.
36. Atmel oficialus tinklapis [interaktyvus] [žiūrėta 2012-12-01] <http://www.atmel.com/>.
37. D.Dervinis 2012, Vaizdų apdorojimas.
38. Variety of different distance sensors comparison table [interaktyvus] [žiūrėta 2011-01-12]
http://zuff.info/RangeFindersComp_E.html.
39. Signalų kartotuvus, galvaninio atskyrimo schemas [interaktyvus] [žiūrėta 2011-03-15]
http://www.hidroteka.lt/lt/prekes_ir_paslaugos/Barjerai/Analog_duom_sius/MS31-LiU/
40. Kompiuterinė sąsaja RS232, MAX232, TTL signalų keitiklis [interaktyvus]
[<http://www.elektronika.lt/schemas/radiotechnika/24675/rs232-ttl-signalu-konverteris/>
41. C++ Programming tutorials & free online samples [interaktyvus] [žiūrėta 2011-04-01]
<http://www.syvum.com/squizzes/cpp/>
42. Andrea E. Rizzoli, Nicoletta Fornara, Luca Maria Gambardella, 2002, A simulation tool for combined rail/road transport in intermodal terminals. Mathematics and Computers in Simulation 59, p. 57-71.
43. Programming languages – C++ international standard IEC 148822 – 1998 [interaktyvus] [žiūrėta 2011-04-24] http://www-d0.fnal.gov/~dladams/cxx_standard.pdf
44. Variety of different microcontrollers. [Interaktyvus], [žiūrėta 2012-04-28]
http://www.tradeoy.com/products_405748.html
45. Nils Boysen, Malte Fliedner. 2011. Determining crane areas in intermodal transshipment yards: The yard partition problem. European Journal of Operational Research. 204. p. 336-342.
46. Barbara A. Reichmuth, Paul S. Stansbury, Randal Y. Taira, Thomas F. Sanquist, and Robert Collins, 2009. Radiation Scanning Methods for Ship-to-Rail Intermodal Cargo

- Containers. Technologies for homeland security. 9, p. 698–703.
47. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero. 2008. Nonlinear Predictive Control for the Management of Container Flows in Maritime Intermodal Terminals. 47th IEEE Conference on Decision and Control Cancun, p. 2800–2805.
 48. Alessandrit, S. Sacone^e, S.Siri, 2004. Management of intermodal container terminals using feedback control, Institute of Intelligent Systems for Automation, p. 882–887.
 49. Chuan Yang, Zhi Zhang, Qiang Zhao. 2008. Study on Intelligent Control of Two-Dimensional Precision Positioning System. Computer Science and Software Engineering. p. 835–838.
 50. Frank Meisel, Christian Bierwirth. 2011. A unified approach for the evaluation of quay crane scheduling models and algorithms. Computers & Operations Research. 38, p. 683–693.
 51. Iris F.A. Vis *, Ren e e de Koster. 2003. Transshipment of containers at a container terminal: An overview. European Journal of Operational Research. 147, p. 1-6
 52. Yijun Chen, Daofang Chang, Weijian MI, Houjun LU. 2010. Study on Container Terminal Virtual Reality System Based on Component Technology. 2nd International Asia Conference on Informatics in Control, Automation and Robotics. p. 299-302.
 53. Endre Pap, Milosav Georgijević, Vladimir Bojanić and Goran Bojanić. 2010. Pseudo-analysis application in complex mechanical systems modelling of container quay cranes. International Symposium on Intelligent Systems and Informatics. 8th, p. 493-496
 54. Atmel AVR Atmega32 microcontroler documentation [žiūrėta 2011-04-21]
http://www.atmel.com/dyn/products/product_docs.asp?category_id=163&family_id=607&subfamily_id=760&part_id=2014
 55. GP2Y0A02YK Long distance measuring sensor [interaktyvus] [žiūrėta 2011-04-24]
http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a02_e.pdf
 56. Balaševičius L, Ezerskis N. 2001 Programuojamieji valdikliai, praktiniai darbai, mokomoji knyga, Kaunas : Technologija, 139 p.

PRIEDAI

1. Bevielēs sāšajos programinis kods

```
#define F_CPU 16000000UL
#include<avr/io.h>
#include <avr/interrupt.h>
#include<util/delay.h>

// global variables

uint8_t TIMER_count = 0;
uint8_t sreg;

void TIMER_Init (void);
void USART_Init(int baudrate);
void USART_Transmit( unsigned char data );
unsigned char USART_Receive( void );

int main(void){
    TIMER_Init();
    USART_Init(103);
    sei();
    unsigned char test = 0;

    while(1)
    {
        while(test == '5') test = USART_Receive();
        TIMER_count=0
        USART_Transmit('6');
        while(test == 7) test = USART_Receive();
        USART_Transmit(TIMER_count);
    }
}

////////////////////////////////////
ISR (TIMER0_OVF_vect){
```

```

TIMER_count++;
if (TIMER_count >= 61) {
    PORTB = ~PORTB;
    TIMER_count = 0;
}
}
/////////////////////////////////////////////////////////////////
void TIMER_Init () {
    // Prescaler = FCPU/1024
    TCCR0|=(1<<CS02)|(1<<CS00);
    //Enable Overflow Interrupt Enable
    TIMSK|=(1<<TOIE0);
    //Initialize Counter
    TCNT0=0;
}

void USART_Init(int baudrate){
    UBRRH = (unsigned char) (baudrate>>8);
    // UBBR perstumiamas i ėemesni baita
    UBRRL = (unsigned char)baudrate;
    // Cia ikeliama UBBR reikėme
    UCSRA |= (1<<U2X);
    // Dvigubo UART greicio reėimas
    UCSRB = (1<<RXEN)|(1<<TXEN)|(1<<RXCIE)|(1<<TXCIE);
    // Ijungti: imtuvas, siustuvas, gautu,
    // priimtu duomenu pertrauktys i jungtos
    UCSRC = (1<<URSEL)|(1<<USBS)|(3<<UCSZ0);
    //Duomenys raėomi i UCSRC registra,
    //2 stop bitai, 8 duomenu bitai
}

void USART_Transmit( unsigned char data )
{
    while ( !( UCSRA & (1<<UDRE)) ) ;

//L

aukiama kol bus iėsiustas prieė tai siunciamas duomuo

```

```
UDR = data;
//Ikeliamas siunciamas duomuo
}
unsigned char USART_Receive( void )
{
while ( !(UCSRA & (1<<RXC)) ) ;
//Laukiama kol atsiunciamas naujas duomuo
return UDR;
//Gražinamas atsiustas duomuo
}
```

2. Neuroninio-neraiškaus tinklo realizavimo ir apmokymo programinis kodas

```
//-----  
-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Main.h"  
#include "About.h"  
//-----  
-----  
#pragma resource "*.dfm"  
TMainForm *MainForm;  
//-----  
-----  
//-----KINTAMIEJI-----  
-----  
float Xref = 0, // Uzduodamas siektinas dydis  
      X0 = 0, // realus dydis  
      X = 0, // iverstis skirtumas tarp reikiamo  
dydzio ir esamo is sensoriaus i valdikli  
      Y0 = 0, // iverstis praeitos iteracijos  
skirtumas skirtass paskaiciuoti pokyciui  
      Y = 0, // iverstis praeitos iteracijos  
skirtumas skirtass paskaiciuoti pokyciui  
      Xn = 0, // iverstis skirtumas tarp reikiamo  
dydzio ir esamo (normalizuotas dydis) - jau i neuronini tinkla  
      Yn = 0, // iverstis skirtumo tarp reikiamo  
dydzio ir esamo pokytis (lyginant su praeita iteracija)  
(normalizuotas dydis) - jau i neuronini tinkla  
      Zn = 0, // iverstis atstumas tarp  
konteineriu  
      On = 0, // isvestis is neuroninio tinklo
```



```

elemento kuris ieina i 4 sluoksni max
        double Xm[1156];                // apmokymo imties X
reiksmes
        double Ym[1156];                // apmokymo imties Y
reiksmes
        double Tm[1156];                // apmokymo imties
tikrasis outputas
        double E0;                      // reali paklaida T-O
realaus isejimo ir turimo isejimo skirtumas
        double E = 0.1;                 // siektinas tikslumas
        double ni = 0.02;               // mokymo zingsnis
        double f[5] = {0,0,0,0,0};     //isvestines
        double delta[5] = {0,0,0,0,0}; // deltos
        double dWkm = 0;
        double error = 0;
        float vvv[1156][1000];
        int vv = 0;
        FILE *F;                        // mokymo imties
failas
        //-----
-----
__fastcall TMainForm::TMainForm(TComponent *Owner)
    : TForm(Owner)
{
}
//-----
-----

void __fastcall TMainForm::CreateMDIChild(String Name)
{
    TMDIChild *Child;

    //--- create a new MDI child window ----
    Child = new TMDIChild(Application);

```

```

        Child->Caption = Name;
        if (FileExists (Name))
            Child->Memo1->Lines->LoadFromFile (Name);
    }
//-----
-----

void __fastcall TMainForm::FileNew1Execute(TObject *Sender)
{
    CreateMDIChild("NONAME" + IntToStr(MDIChildCount + 1));
}
//-----
-----

void __fastcall TMainForm::HelpAbout1Execute(TObject *Sender)
{
    AboutBox->ShowModal();
}
//-----
-----

void __fastcall TMainForm::FileExit1Execute(TObject *Sender)
{
    Close();
}
//-----
-----

void __fastcall TMainForm::Button1Click(TObject *Sender)
{
    Xref = StrToFloat(XrefEdit->Text);
    X0    = StrToFloat(X0Edit->Text);
    X     = Xref-X0;
}

```

```

        Y      = X-Y0;
        Y0     = X;
        Xn     = X/100;
        Yn     = Y/100;
        neuro();
    }
//-----
-----

void __fastcall TMainForm::Button2Click(TObject *Sender)
{
    if((F = fopen("sakusa.TXT", "r"))==NULL)
        ShowMessage("The file could not be opened");
    char skaic1[10],skaic2[10],skaic3[10];
    if(!F) { ShowMessage("Hit open..."); return;}
    for(int i = 0; i<1156; i++){
        fscanf(F, "%s", skaic1);
        fscanf(F, "%s", skaic2);
        fscanf(F, "%s", skaic3);
        Xm[i] = StrToFloat(skaic1);
        Ym[i] = StrToFloat(skaic2);
        Tm[i] = StrToFloat(skaic3);
    }
}
//-----
-----

void __fastcall TMainForm::Button4Click(TObject *Sender)
{
    ofstream output("duomenys.txt");
    for (int qq = 0; qq < 1000; qq++) {
        for (int mmm = 0; mmm < 1156; mmm++) {
            output << vvv[qq][mmm] <<"\t";

```

```

        }
        output << endl;
    }
}
//-----
-----

void __fastcall TMainForm::Button3Click(TObject *Sender)
{
    if((F = fopen("sakusa.TXT", "r"))==NULL)
        ShowMessage("The file could not be opened");
}
//-----
-----

void __fastcall TMainForm::Button5Click(TObject *Sender)
{
    int nr=0;
    nr = StrToInt(Edit11->Text);
    if((nr>=0) && (nr<=1156)){
        Edit8->Text  = FloatToStr(Xm[nr]);
        Edit9->Text  = FloatToStr(Ym[nr]);
        Edit10->Text = FloatToStr(Tm[nr]);
    }
    else ShowMessage("enter element number [0-10200]");
    nr=0;
}
//-----
-----

///svoriu isvedimui
void __fastcall TMainForm::Button6Click(TObject *Sender)
{
    Wkm0Edit->Text  = FloatToStr(Wkm[0]);
    Wkm1Edit->Text  = FloatToStr(Wkm[1]);
}

```

```

Wkm2Edit->Text  = FloatToStr(Wkm[2]);
Wkm3Edit->Text  = FloatToStr(Wkm[3]);
Wkm4Edit->Text  = FloatToStr(Wkm[4]);
Wkm5Edit->Text  = FloatToStr(Wkm[5]);
Wkm6Edit->Text  = FloatToStr(Wkm[6]);
Wkm7Edit->Text  = FloatToStr(Wkm[7]);
Wkm8Edit->Text  = FloatToStr(Wkm[8]);
Wkm9Edit->Text  = FloatToStr(Wkm[9]);
Wkm10Edit->Text = FloatToStr(Wkm[10]);
Wkm11Edit->Text = FloatToStr(Wkm[11]);

```

```

      XEdit->Text =FloatToStr( X      );
Y0Edit->Text =FloatToStr(      Y0      );
YEdit->Text =FloatToStr( Y      );
XnEdit->Text = FloatToStr(      Xn      );
YnEdit->Text = FloatToStr(      Yn      );

```

```

A1Edit->Text = FloatToStr(      A[0]);
A2Edit->Text = FloatToStr(      A[1]);
A3Edit->Text = FloatToStr(      A[2]);
A4Edit->Text = FloatToStr(      A[3]);

```

```

B1Edit->Text = FloatToStr(      B[0]);
B2Edit->Text = FloatToStr(      B[1]);
B3Edit->Text = FloatToStr(      B[2]);

```

```

R1Edit->Text  =      FloatToStr(      R[0]);
R2Edit->Text  =      FloatToStr(      R[1]);
R3Edit->Text  =      FloatToStr(      R[2]);
R4Edit->Text  =      FloatToStr(      R[3]);
R5Edit->Text  =      FloatToStr(      R[4]);
R6Edit->Text  =      FloatToStr(      R[5]);
R7Edit->Text  =      FloatToStr(      R[6]);
R8Edit->Text  =      FloatToStr(      R[7]);

```

```

R9Edit->Text    =    FloatToStr(    R[8]);
R10Edit->Text   =    FloatToStr(    R[9]);
R11Edit->Text   =    FloatToStr(    R[10]);
R12Edit->Text   =    FloatToStr(    R[11]);

O0Edit->Text    = FloatToStr (O[0]);
O1Edit->Text    = FloatToStr (O[1]);
O2Edit->Text    = FloatToStr (O[2]);
O3Edit->Text    = FloatToStr (O[3]);
O4Edit->Text    = FloatToStr (O[4]);

OnEdit->Text    =FloatToStr (On);
}
//-----
-----
//*****neuroninis
tinklas*****
void TMainForm::neuro(void)
{
    //1 sluoksnis
        //XnEdit->Text =    FloatToStr(    Xn    );
        //YnEdit->Text =    FloatToStr(    Yn    );
    // 2 sluoksnis
        A[0]=A[1]=A[2]=A[3]=0;
        B[0]=B[1]=B[2]=0;
        C[0]=C[1]=C[2]=0;
    // A 4 nariai
        for(int j = 0; j<4; j++){
            if(Xn <= b[0]){A[0]=1; break;}
            if(Xn >= b[3]){A[3]=1; break;}
            if(Xn>=a[j]&&Xn<=b[j]){A[j]=(Xn-a[j])/(b[j]-a[j]);}
            else if(Xn>=b[j]&&Xn<=c[j]){A[j]=(Xn-c[j])/(b[j]-c[j]);}
        }
    //B 3 nariai

```

```

    for(int i=0; i<3; i++){
        if(Yn <= b[4]){B[0]=1; break;}
        if(Yn >= b[6]){B[2]=1; break;}
        if(Yn>=a[i+4]&&Yn<=b[i+4]){B[i]=(Yn-a[i+4])/(b[i+4]-
a[i+4]);}
        else if(Yn>=b[i+4]&&Yn<=c[i+4]){B[i]=(Yn-c[i+4])/(b[i+4]-
c[i+4]);}
    }
//C 3 nariai
    for(int i=0; i<3; i++){
        if(Yn <= b[4]){C[0]=1; break;}
        if(Yn >= b[6]){C[2]=1; break;}
        if(Yn>=a[i+4]&&Yn<=b[i+4]){C[i]=(Yn-a[i+4])/(b[i+4]-
a[i+4]);}
        else if(Yn>=b[i+4]&&Yn<=c[i+4]){C[i]=(Yn-c[i+4])/(b[i+4]-
c[i+4]);}
    }
// 3 sluoksnis
    int e = 0;
    for(int i = 0; i<4; i++){
        for(int j = 0; j<3; j++){
            for(int k = 0; k<3; k++){
                R[e++]=A[i]*B[j]*C[k];
            }
        }
    }

//OUTPUT PASKAICIAVIMAS 4 sluoksnis o suma 4 sluoksnio bus 5

O[0] =O[1] =O[2] =O[3] =O[4] =0;
Oms[0]=Oms[1]=Oms[2]=Oms[3]=Oms[4]=0; //skaitikliai
Omv[0]=Omv[1]=Omv[2]=Omv[3]=Omv[4]=0; //vardikliai
Ok[0]=Ok[1]=Ok[2]=Ok[3]=Ok[4]=0;

```

```

//NL output paskaiciavimas R[0] O[0] abc[7]
O[0] = R[0]*Wkm[0];
Oms[0]=O[0]*b[7]*(c[7]-a[7]);
Omv[0]=O[0]*(c[7]-a[7]);
Ok[0] = 0;

//NS output paskaiciavimas R[1,3] O[1] abc[8]
if (R[1]*Wkm[1]>=R[3]*Wkm[3]) Ok[1]=1;
else {Ok[1]=3;}
O[1] = R[Ok[1]]*Wkm[Ok[1]];
Oms[1]=O[1]*b[8]*(c[8]-a[8]);
Omv[1]=O[1]*(c[8]-a[8]);

//ZE output paskaiciavimas R[2,4,6] O[2] abc[9]
if ((R[2]*Wkm[2]>=R[4]*Wkm[4]) && (R[2]*Wkm[2]>=R[6]*Wkm[6]))
Ok[2]=2;
else if (R[4]*Wkm[4]>=R[6]*Wkm[6]) Ok[2]=4;
else Ok[2]=6;
O[2] = R[Ok[2]]*Wkm[Ok[2]];
Oms[2]=O[2]*b[9]*(c[9]-a[9]);
Omv[2]=O[2]*(c[9]-a[9]);

//PS output paskaiciavimas R[5,7,9] O[3] abc[10]
if ((R[5]*Wkm[5]>=R[7]*Wkm[7]) && (R[5]*Wkm[5]>=R[9]*Wkm[9]))
Ok[3]=5;
else if (R[7]*Wkm[7]>=R[9]*Wkm[9]) Ok[3]=7;
else Ok[3]=9;
O[3] = R[Ok[3]]*Wkm[Ok[3]];
Oms[3]=O[3]*b[10]*(c[10]-a[10]);
Omv[3]=O[3]*(c[10]-a[10]);

//PL output paskaiciavimas R[8,10,11] O[4] abc[11]
if
((R[8]*Wkm[8]>=R[10]*Wkm[10]) && (R[8]*Wkm[8]>=R[11]*Wkm[11]))

```

```

Ok[4]=8;

else if (R[10]*Wkm[10]>=R[11]*Wkm[11]) Ok[4]=10;
else Ok[4]=11;

O[4] = R[Ok[4]]*Wkm[Ok[4]];
Oms[4]=O[4]*b[11]*(c[11]-a[11]);
Omv[4]=O[4]*(c[11]-a[11]);

//O0Edit->Text = FloatToStr (O[0]);
//O1Edit->Text = FloatToStr (O[1]);
//O2Edit->Text = FloatToStr (O[2]);
//O3Edit->Text = FloatToStr (O[3]);
//O4Edit->Text = FloatToStr (O[4]);

//OUTPUTAS
//if ((Omv[0]+Omv[1]+Omv[2]+Omv[3]+Omv[4])==0) {
//for(int oo = 0; oo<12; oo++){
//Memo2->Lines->Add(FloatToStr(Wkm[oo]));
//Memo2->Lines->Add(FloatToStr(R[oo]*Wkm[oo]));
//}
//Memo2->Lines->Add("-----");
//for(int oo = 0; oo<12; oo++){
//Memo2->Lines->Add(FloatToStr(Wkm[oo]));
//Memo2->Lines->Add(FloatToStr(R[oo]));
//}
//}

On=(Oms[0]+Oms[1]+Oms[2]+Oms[3]+Oms[4])/(Omv[0]+Omv[1]+Omv[2]+
Omv[3]+Omv[4]);

//OnEdit->Text =FloatToStr (On);

}

//-----
-----

// *****apmokymas*****

```

```

void __fastcall TMainForm::Button7Click(TObject *Sender)
{
    for(int q=0; q<1000; q++){
        error=0;
        for(int mm = 0; mm<1156; mm++){
            for(int k = 0; k<5; k++){
                Xn = Xm[mm];
                Yn = Ym[mm];
                //for (int iii = 0;iii<12; iii++) {
                //    if ((Wkm[iii]<= 0.001)&&(Wkm[iii]>=0))
Wkm[iii]=0.001;
                //    if ((Wkm[iii]>= -0.001)&&(Wkm[iii]>0)) Wkm[iii]=-
0.001;

                //}
                neuro();
                E0 = Tm[mm]-On;        // paskai2iuojame klaida
                vvv[q][mm]=E0;
                // isvestiniu ir delta reiksmiu skaiciavimas tuo paciu
delta = E0xF`
                //0 elemento isvestine

                double vard = 0;
                for (int i = 0; i < 5; i++) {                //susumuojame visu
4 sluoksnio element7
                    vard += O[i];                // sandaugas su centrais
                }
                vard*=vard;

                for(int m = 0; m<9; m++){
                    double skait = 0;
                    for(int j = 0; j<9; j++){                //suumuojame 5
sluoksnio elementu sandauga
                        if(j==m) continue;
                        skait += O[j]*(b[7+m]-b[7+j]);        //sandauga is

```

```

m elemento centro atimant
    }                                     //j elementus...
    f[m]=skait/vard;                     //m
elemento isvestine
    delta[m] = E0*f[m];                  //m elemento
delta
    dWkm=ni*delta[m]*R[Ok[m]];           //
paskaiciuojam svorio pokycius
    if ( (Wkm[Ok[m]]+dWkm)<0.001) Wkm[Ok[m]]=0.001;
    else Wkm[Ok[m]]+=dWkm;
    //koreguojame svorius

    }
    }
    //Mem1->Lines->Add(FloatToStr(E0));
    error+=E0*E0;

}
Mem1->Lines->Add(FloatToStr(error/1156));
//Mem1->Lines->Add(FloatToStr(error/1)+"\t"+q);
//Mem1->Lines-
>Add(FloatToStr(Wkm[0])+"\t"+FloatToStr(Wkm[1])+"\t"+FloatToStr(Wk
m[2])+"\t"+FloatToStr(Wkm[3])+"\t"+FloatToStr(Wkm[4])+"\t"+FloatTo
Str(Wkm[5])+"\t"+FloatToStr(Wkm[6])+"\t"+FloatToStr(Wkm[7])+"\t"+F
loatToStr(Wkm[8])+"\t"+FloatToStr(Wkm[9])+"\t"+FloatToStr(Wkm[10])
+"\t"+FloatToStr(Wkm[11]));
}

    //pilnam apmokymui iiskomentuoti sekancias dvi ir virsui dvi
    //}while (E>E0)
    //}

}
//-----

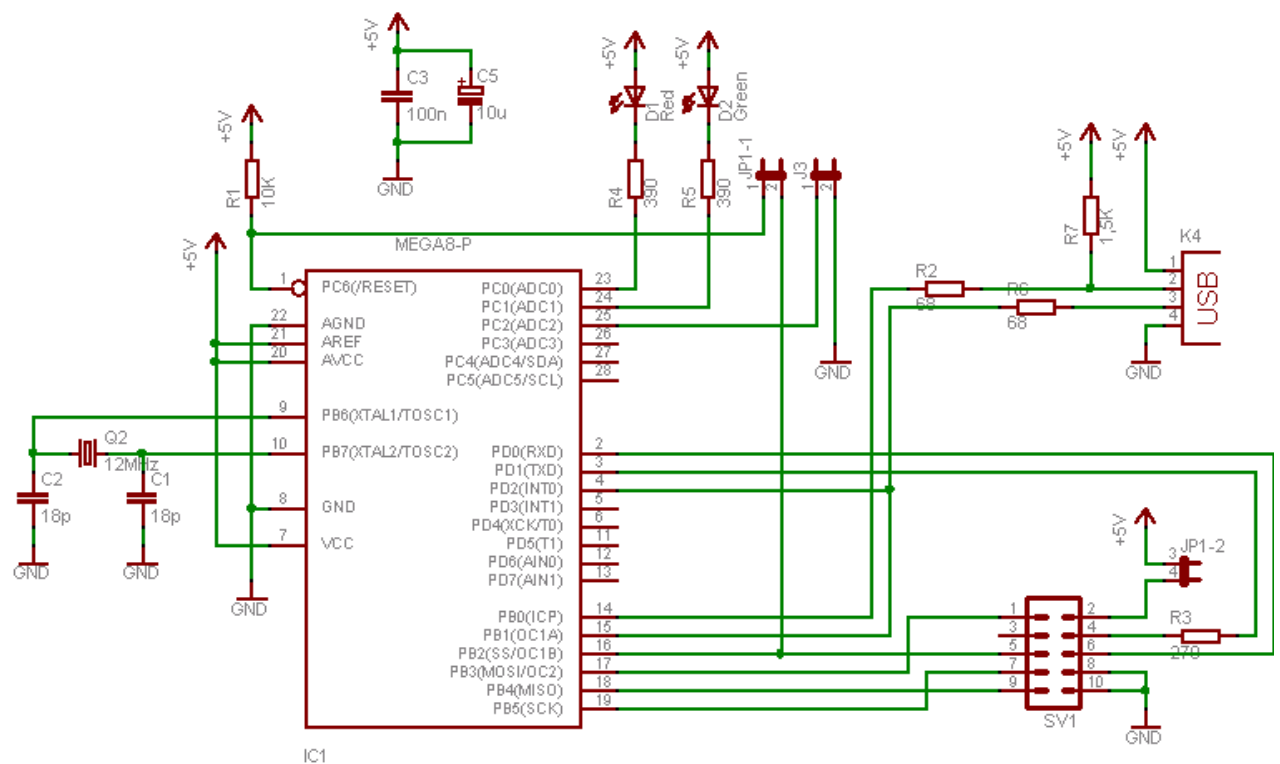
```

```
void __fastcall TMainForm::Button8Click(TObject *Sender)
{

    //for(int q=v; q<q+100; q++){
    //for(int mm=0; mm<1156; mm++){
        //      Memo2->Lines->Add(FloatToStr(vvv[vv][mm]));
    // }//}

    for(int mm=0; mm<1156; mm++){
        Memo2->Lines-
>Add(FloatToStr(mm)+"\t"+FloatToStr(vvv[vv][mm])+"\t"+FloatToStr(v
vv[vv+1][mm])+"\t"+FloatToStr(vvv[vv+2][mm])+"\t"+FloatToStr(vvv[v
v+3][mm])+"\t"+FloatToStr(vvv[vv+4][mm])+"\t"+FloatToStr(vvv[vv+5]
[mm])+"\t"+FloatToStr(vvv[vv+6][mm])+"\t"+FloatToStr(vvv[vv+7][mm]
)+"\t"+FloatToStr(vvv[vv+8][mm])+"\t"+FloatToStr(vvv[vv+9][mm]));
        }
        vv+=10;
    }
    //-----
    -----
```

3. Elektroninė programatoriaus „USB Asp“ schema



4. Mokslinės publikacijos

Impulso pločio moduliacijos kompiuterinis valdymas mikrovaldiklio pagrindu, skirtas nuolatinės srovės variklio greičio reguliavimui

Vytautas Beniušis¹, Ilja Gusiatin², Sergej Jakovlev³, Janis Donins⁴

^{1,2,3}Klaipėdos universitetas, Informatikos inžinerijos katedra, Bijūnų g. 17, LT-91225, Klaipėda, Lietuva
El. paštas: ¹v.beniušis@yahoo.com; ²igusiatin89@gmail.com; ³s.jakovlev.86@gmail.com

⁴Ventspils University College, The Faculty of Information Technologies, Inženieru str. 101a, LV-3600, Ventspils, Latvia
El. paštas: ⁴j.donins@yahoo.com

Santrauka. Analizuojant pramoninių procesų valdymo technines informacines sistemas buvo pastebėta, kad įmanoma išvengti specifinės techninės įrangos panaudojimo. Straipsnyje aprašomas nuolatinės srovės variklio greičio valdymo informacinės sistemos prototipas, paremtas mikrovaldiklio tiesiogiai išduodamo impulso pločio (IPM) reguliavimu, nenaudojant papildomų TL494 modulių, atsakingų už IPM formavimą. Suprojektuota vartotojo-kompiuterio sąsaja papildomai mažina panaudojamą techninę įrangą, kas leidžia efektyviau išnaudoti esamus pramoninių procesų resursus. Atlikta variklio apsisukimo greičio priklausomybės nuo paduodamo impulso pločio statistinė analizė parodė, kad egzistuoja labai stipri teigiama tiesinė koreliacinė priklausomybė tarp matuojamų duomenų.

Reikšminiai žodžiai: impulso pločio moduliacija, mikrovaldiklis, vartotojo-kompiuterio sąsaja, nuolatinės srovės variklis.

Įvadas

Pastaruoju metu automatinės duomenų surinkimo ir apdorojimo sistemos vis labiau yra pritaikomos procesų valdymo ir stebėjimo srityse (Sadjadi *et al.* 2011; Zhu *et al.* 2010). Tokio tipo sistemų realizavimui yra naudojami „Atmega“ tipo mikrovaldikliai, jie yra taikomi daugelyje sričių: prekyboje, pramonėje, agrokultūrose bei buityje (Fisher ir kt. 2010). Pramoninių procesų valdymo sistemų projektavimas yra ypač aktualus šiandieniniame pasaulyje.

Bayindir ir kt. (2007) savo darbe aprašo valdymo modulį, skirtą trinties suvirinimo aparatui. Jų darbe siūloma variklių greičio valdymo komandoms įvesti bei rezultatams atvaizduoti, naudoti atskirą techninę įrangą: skystųjų kristalų displejų ir klaviatūrą.

Sadjadi ir kt. (2011) sukūrė autonominę valymo sistemą, kur keturiems nuolatinės srovės varikliams valdyti yra naudojami atskiri mikrovaldikliai, keturis mikrovaldiklius apjungia vienas pagrindinis valdiklis.

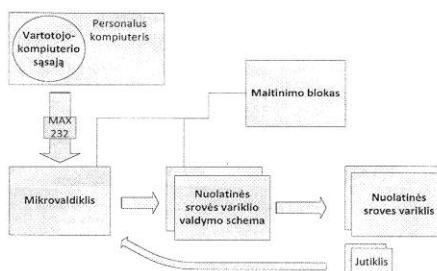
Zhu ir kt. (2010) suprojektavo įpurškimo siurblio valdiklį, kurio valdymas yra pagrįstas impulso pločio moduliacija. IPM kūrimui autorius panaudojo TL494 modulius, kuriuos valdo su mikrovaldikliu.

Šiame darbe yra analizuojamas kelių nuolatinės srovės variklių greičio su vienu mikrovaldikliu valdymas. Aprašomas modeliuojamas informacinės sistemos prototipas, padėsiantis sumažinti išlaidas nenaudojant papildomų modulių IPM formavimui bei techninės įrangos procesų stebėjimui ir duomenų įvedimui.

Papildomai yra atlikta variklio apsisukimo greičio nuo paduodamo impulso pločio priklausomybės statistinė analizė.

1. Sistemos architektūros aprašymas

Projektuojama sistema susideda iš keturių pagrindinių komponentų: maitinimo bloko, mikrovaldiklio, vartotojo-kompiuterio sąsajos ir dviejų lygiagrečių nuolatinės srovės variklių valdymo schemų, galinčių sukurti variklius į abi puses. Kuriamos sistemos blokinė diagrama parodyta 1 pav.



1 pav. Informacinės sistemos prototipo blokinė diagrama

Egzistuoja daug mikrovaldiklio programavimo kalbų, leidžiančių programuotojui sukurti programas su sudėtingomis funkcijomis, nesimokant atskirų mikrovaldiklių žemo lygio assemblerio programavimo kalbos, kur daugiausiai funkcionalumo suteikia pasirinkta laisvai prieinama C++ programavimo kalba.

1.1 Techninės įrangos panaudojimas

Sistamai iš išorinio šaltinio teikiama 24 V stabilizuota įtampa. Variklių valdymo schemoms reikalinga 24 V, o mikrovaldikliui 5 V įtampa, todėl yra naudojama papildoma įtampos keitimo schema (2 pav., a) keičianti ir stabilizuojanti įtampą iš 24 V į 5 V ir išduodanti 1 A srovę, kurios pakanka mikrovaldikliui ir MAX232 mikroschemai.

Mikrovaldiklis turi komunikuoti su kompiuteriu, kad gautų nurodymus ir perduotų esamą būseną, nuskaityti sensorių parodymus ir valdyti variklių greičius. Tam buvo panaudotas „Atmel Atmega32“ mikrovaldiklis (2 pav., b), kurio pagrindinės funkcijos yra:

- 1) keturi aštuonių bitų raiškos, 31,372 kHz greičio impulso pločio moduliavimo kanalai, variklių greičiui bei sukimosi kryptčiai nustatyti;
- 2) du išoriniai pertraukties kanalai, jutiklių nuskaitymui;
- 3) signalų priėmimo ir siuntimo kanalai, komunikacijai su kompiuteriu.

Variklio greičiui reguliuoti bei sukimosi kryptčiai keisti buvo sumodeliuota elektroninė schema (2 pav., c),

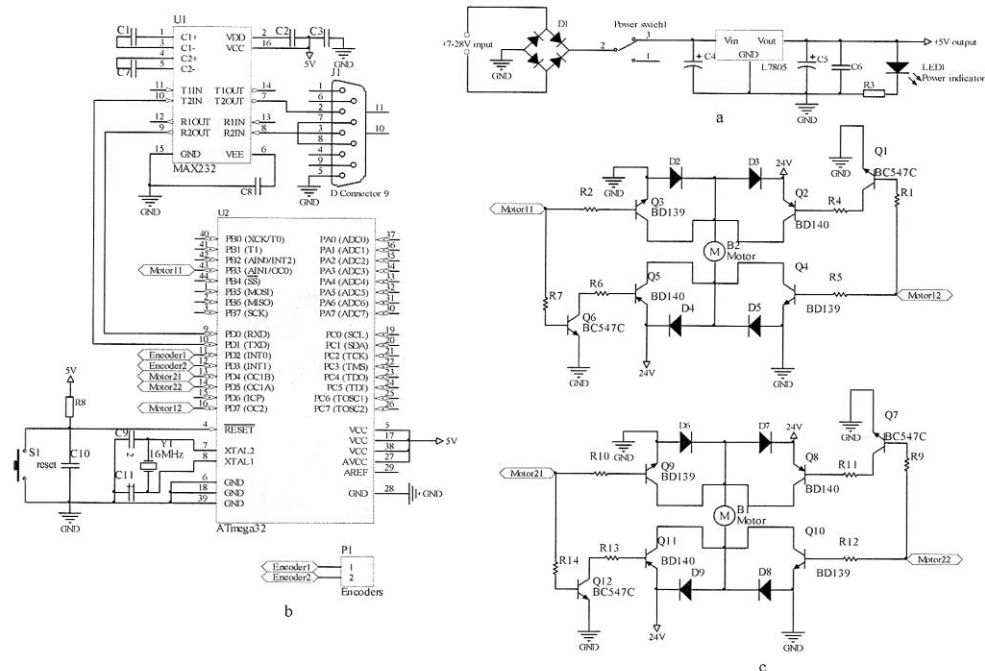
veikianti H-tilto principu, įgalinanti valdyti variklį dviem valdymo kanalais. Padavus 5 V įtampos signalą per atitinkamą kanalą, variklis sukasi viena kryptimi. Variklio greičio valdymui yra paduodama impulso pločio moduliacija, kuri reguliuoja schemeje esančių tranzistorių pilną atidarymą ar uždarymą.

Ryšiai užtikrinti tarp vartotojo ir sistemos yra panaudojamas paprastas nešiojamas (ar stacionarus) kompiuteris, turintis COM pajungimo uostą.

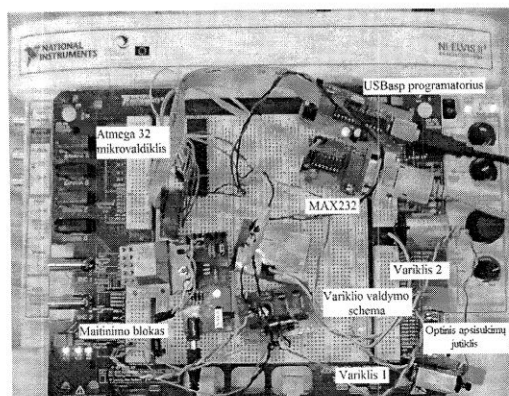
Tam yra sukurta vartotojo-kompiuterio sąsajos programa, bendraujanti su mikrovaldikliu RS-232 protokolu. Duomenys perduodami į MAX232 mikroschema, kur jie yra konvertuojami į mikrovaldikliams suprantamus lygiagrečius TTL signalus.

Pastovaus užduoto greičio užtikrinimui naudojami optiniai apskukų jutikliai, kurie skaičiuoja variklio apsisukimus ir siunčia signalus mikrovaldikliui.

Tokiu būdu realizuotas grįžtamasis ryšys leidžiantis mikrovaldikliui valdyti variklio greitį realiu laiku, reaguojant į variklio apkrovos pasikeitimus, ir perduoti esamų greičių parametrus kompiuteriui (3 pav.).



2 pav. Informacinės sistemos elektroninė schema: maitinimo blokas (a), mikrovaldiklis Atmega32, sujungtas su MAX232 mikroschema (b), nuolatinės srovės variklių valdymo schemas (c).



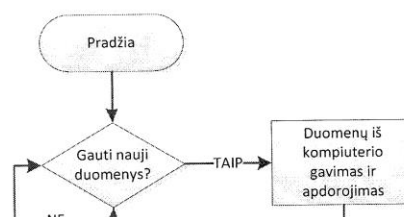
3 pav. Sukurtas sistemos prototipas

1.2 Programinės įrangos panaudojimas

Sukurto informacinės sistemos prototipo programinė dalis susideda iš kompiuterio, esančio vartotojo

„Connect“. Programa leidžia nustatyti dviejų variklių sukimosi kryptį bei greitį. Paspaudus mygtuką „Start“ prie reikiamo variklio, nustatytos impulso pločio ir krypties reikšmės yra išsiunčiamos į mikrovaldiklį. Esamas variklio greitis išvedamas apsisukimais per minutę. Mygtukas „Stop“ sustabdo variklį.

Sistemos mikrovaldiklio programos algoritmas yra pavaizduotas 5 pav. Sukurta valdymo programa yra sukompiliuojama į mašininį kodą ir įrašoma į mikrovaldiklį.



o programa atitinkamai reaguoja performuojant IPM.
2. Tyrimų rezultatai

Variklio greičio kitimo priklausomybės nuo užduotos impulso pločio reikšmės išgavimui (7 pav.), buvo sumaketuotas sistemos prototipas. Tyrimo metu gautiems duomenims buvo paskaičiuota koreliacinė priklausomybė.

Tyrimai buvo atlikti naudojant nuolatinės srovės ORION MMN-5C2RJ variklį (12 V, 0,4 W).

2.1. Valdymo impulso plotis

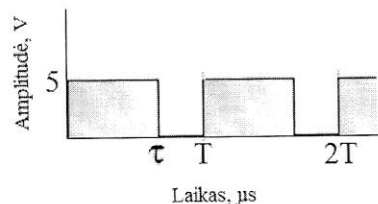
Impulso pločio reikšmė (IPR) yra abstrakti sąvoka, apibūdinanti mikrovaldiklio 5 V išduodamo signalo sklaidimo trukmę tam tikrų dažniu (6 pav.).

$$\tau = \frac{2 \cdot IPR \cdot}{F_{kvarc}} \cdot 10^6, \quad (1)$$

čia: IPR – impulso pločio reikšmė su (1:1:255) žingsniu; F_{kvarc} – naudojamo išorinio kvarcinio rezonatoriaus dažnis, Hz.

Variklio valdymo schemoje esantys tranzistoriai sustiprina mikrovaldiklio išduotą impulsą, nekeičiant jo trukmės. Variklis sukasi tik aktyvią veiklos ciklo dalį, o likusią dalį jis lėtėja. Kadangi impulso dažnis yra aukštas, variklis įgauna pastovią sukimosi būseną. Aktyviosios veiklos ciklo dalies kitimas įtakoja variklio sukimosi greitį.

Gavus komandą „sustoti“ variklis yra sustabdomas.



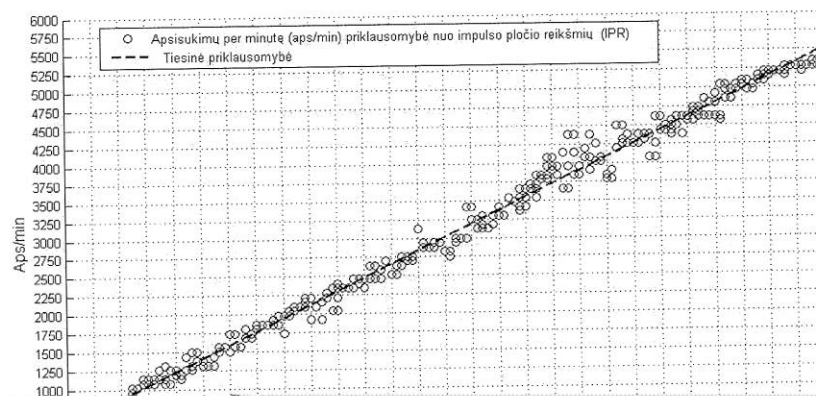
6 pav. Impulso pločio (trukmės) grafikas.

Impulso trukmė yra apskaičiuojama pagal tokia formulę (1):

2.2. Statistinė duomenų analizė

Eksperimento statistinių duomenų interpretavimui buvo panaudota „scatter“ atvaizdavimo komanda MATLAB programavimo aplinkoje. Atvaizduojant tyrimų rezultatai parodė, kad egzistuoja labai stipri teigiama tiesinė koreliacinė priklausomybė tarp impulso pločio paduodamų reikšmių ir variklio apsisukimų per minutę skaičiaus. Patikrinimui buvo apskaičiuotas koreliacijos koeficientas (2):

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \cdot \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (2)$$



čia: n – matavimų skaičius; (x_i, y_i) – impulso pločio ir apsisukimų per minutę reikšmės koordinatinių ašyse; $\bar{x}$ ir $\bar{y}$ – impulso pločio ir apsisukimų per minutę imčių reikšmių aritmetiniai vidurkiai. Prie $n = 262$, koreliacijos koeficientas $r_{xy} = 0,9947$ ir regresinė tiesinė lygties funkcija (trendo linija) yra formuluojama, kaip $y_i = 35,72 \cdot x_i - 196,52$, kur krypties koeficientas lygus 35,72, o laisvo narių koeficientas lygus -196,52. Tai

Praktinis sistemos prototipo panaudojimas yra įgyvendinamas Klaipėdos universiteto studentų projekte „Intelligent Train Control System“. Projekto oficialus tinklapis: <http://vagonas.info/>

Padėka

Autoriai dėkoja projektui “LLII-061 Development of Joint Research and Training Centre in High Technology Area”

ISSN 1822-296X



TRANSPORT MEANS

PROCEEDINGS OF THE
INTERNATIONAL CONFERENCE

Contents

Preface	
J. Furch, Prognostics and Health Management in Military	4
Ł.J. Orman, R. Chatys, Heat Transfer Augmentation Possibility for Vehicle Heat Exchangers	5
S. Šinkūnas, A. Kiela, A. Adomavičius, Shear Stress Variation in Liquid Film Flow inside a Vertical Tube	9
J.Lazauskas, A.Kazlauskienė, A. Aranskis, The Research of Transport Companies Performance Measurement System	13
J. Stodola, P. Stodola, Life-Time Weight Growth Impact on Military Vehicle Performance and Dependability	17
J. Glos, Modern Methods of Tribological Diagnostics	22
Š. Čornák, Assessment of Diesel Fuel Quality	26
J.Čerka, Marine Powering Prediction	30
B. Leitner, The Software Tool for Mechanical Structures Dynamic Systems Identification	34
B. Leitner, The Least Squares Method in a Parametric Identification of the Large Crane System	38
D. Vidriková, Z. Dvořák, V. Kaplan, The Current State of Protection of Critical Infrastructure Elements of Road Transport in the Conditions of the Slovak Republic	42
E. Sventeková, Z. Dvořák, Human Activity as a Risk in Railway Transport	46
Z. Vintr, M. Vintr, Preventive Maintenance Optimization of Pantographs of Railway Vehicles	50
V. Kaplan, M. Lojda, On Determination of the Loading Capacity on the Šílený Truss Bridge in Kojetín	54
A. Andziulis, G. Gaigals, T. Lenkauskas, E. Visakavičius, S. Jakovlev, T. Eglynas, V. Beniušis, Comparison of Two Image Processing Techniques for Objects Detection on the Sea Floor	58
I.Lešinskis, A.Pavlovičs, The Aspects of Implementation of Unmanned Aerial Vehicles for Ice Situation Awareness In Maritime Traffic	62
E. Speičytė, Impact of Public Logistic Centre Establishment on Freight Transport External Costs	65
T. Vintr, Approaches to Maintainability Allocation of Military Combat Vehicles	69
E.Damberg, J.Erbs, R.Jankevics, A.Strautins, P.Jefimenko, I.Rozenstrauha, J.Cimanskis, J.Zarins, The Freeport of Riga inventory of air emissions from ocean going vessels	73
L. Turkina, I. Panasiuk, Optimization of Production of Ship Hull Parts	77
P. Janu, J. Bajer, Communication Verification of On-board Aircraft Electronic System	81
I.Stelpa, Development Prospects of Northern Sea Route	84
D. Andrejeva, The Risk of Human Error Reduction by Implementing the Automatic Collision Avoidance System	88
L. Vindergauz, Baf in Short Sea Roro Shipping - Significance and Application: Dfids Seaways Case Study	92
	96

P. Maruschak, R. Bishchak, A. Pylypenko, D. Baran, Analysis of Multiple Cracking and Degradation of Physical and Mechanical Properties of Roller Surface of a Continuous Casting Machine	100
K. Kemzūraitė, G. Bureika, V. Žuraulis, Investigation of Vehicle Stability on Road Curves in Winter Conditions	104
L. I. Vasiljeva, Estimation of Elements' Residual Operational Resources of Ship Pipeline	108
Z. Krobot, T. Túró, Telemetry used for the maintenance of military vehicles	112
Š. Čorňák, V. Neumann, J. Veverka, Contribution to the Thermal Stress Solution of a Vehicle Brake System	116
R. Makaras, A. Kumetaitis, J. Sapragnas, Influence of Cylinders Cut-off on Economy of Internal Combustion Engine	120
Z. Bazaras, B. Timofeev, N. Vasilieva, A. Vilkauskas, Material State Evaluation Elements of Nuclear Power Plants Equipment with RBMK-1000 Reactors after Prolonged Operation Time	124
Z. Bazaras, B. Timofeev, N. Vasilieva, R. Kersys, The Base Tendencies of Global Electric Power Engineering	128
L. Šeibokaitė, A. Endriulaitienė, K. Žardeckaitė-Matulaitienė, R. Markšaitytė, Self-reported and experimental risky driving measurement: what to rely on in young drivers?	131
L. Miliauskaitė, A. Jarašūnienė, Concept of Railway in East West Green Transport Corridor	135
A. Jarašūnienė, L. Miliauskaitė, Application of Measures Solving Traffic Safety Issues on Lithuanian Roads	139
E. Baublys, N. Baublys, Improvement of Signalization System of Landing Gear of the Aircraft Yakovlev Yak-52	143
A. Griškevičiūtė-Gečienė, Modelling of the Assessment of the Investment Projects on Urban Transport Infrastructure and Communication Systems	148
V. Sadauskas, Drivers Age and Traffic Accidents in Lithuania	153
D. Kalisinskas, A. Keršys, NO _x Investigation for Internal Combustion Engine Operating on Hydrogen Fuel	156
Z. Hejmal, Acoustic Emission Signal Generated by Wooden Material under Bending Stress	159
A. Unbedahts, I. Silina, The Present Situation and Tendency of on Shore Power Supply (Ops) Implementation Process in the Ports of Baltic States.	163
Š. Liščák, K. Olejnik, Criticism of the European Regulations with an Aim to Increase Haulage Capability of the Road Train	166
G. Labanauskas, R. Palšaitis, Substantiation of Development Transport Terminals in Lithuania	170
V. Dyachkov, R. Mickevičienė, M. Gedvilas, S. Sulcius, Conceptual Design of ECO Research Floating Self-propelled Laboratory	174
R. Skvireckas, E. Lazauskas, G. Turauskas, Investigation of Nature of Failures of Freight Wagons, Arrived at the Kaunas Wagon Distribution Station	177
A. Andziulis, J. Harja, T. Eglynas, S. Jakovlev, V. Beniušis, T. Lenkauskas, Application of Speed Correction Algorithm for Trains Monitoring in an Intelligent Transport Control System	181
N. Dobržinskij, Use of Structural Schemes to Evaluate Reliability of the System Man-Internal Combustion Engine, Operating in Mountains of Afghanistan	185

Comparison of Two Image Processing Techniques for Objects Detection on the Sea Floor

A. Andziulis*, G. Gaigals**, T. Lenkauskas*, E. Visakavičius***, S. Jakovlev*, T. Eglynas, V. Beniušis

*Klaipėda University, Department of Mechanics Engineering, Bijūnų 17, 91225, Klaipėda, Lithuania,
E-mail: tomas.lenkauskas@gmail.com

**Ventspils International Radio Astronomy Centre, Akademijas laukums 1-1503, Rīga, LV-1050, Latvia

***Klaipėda University Coastal Research and Planning Institute, H. Manto 84, 92294, Klaipėda, Lithuania,
E-mail: erikas@corpi.ku.lt

Abstract

One of the main port maintenance problems is alluvial caused by water currents and due to the intense shipping gathered waste. Port Authority constantly updates the GIS data on the port and surrounding waters of the most important shipping lanes in order to ensure safe navigation. Therefore, it is essential to perform sea floor sonar scan surveys that are mandatory before any kind of underwater works. The major application is port expansion or renovation works, cable and pipe-laying. Sea floor sonar surveys are also important for detection of sunken ships and larger lost objects and defining seafloor surface structure. One of the main problems faced in such works is ultrasound sea floor sonar scan results analysis speed in real time (image processing). This case study deals with specific images processing techniques/methods while analyzing their performance in processing time.

KEY WORDS: seafloor research, sonar, image processing, shipping.

1. Introduction

In shallow waters of Baltic Sea, researchers use side scan sonars for Baltic Sea and Klaipėda port waters sea floor scanning. At this moment, there are no recognition tools that could help efficiently detect conspicuous areas of the sea floor in the research surveys. Here two methods were proposed for the application with side scan sonar mechatronics system that could solve the above mentioned problem [1].

Side scan sonar gives grayscale images, where pixels values range between 0 and 255 (here 0 is black and 255 are white colors). If the sonar acoustic signal is reflected very well, it will show as black spots on sonar image, e.g. metal ship hull would reflect signal very well and it will appear as a large dark color area on the sea floor. And if sonar gets absolutely no signal from that area, which shows us a white colored acoustic shadow (see Fig. 1). Two different methods were developed to detect large underwater objects:

- First method searches for reflective objects on the sea floor.
- Second method searches for large acoustic shadows. This method was designed to use less computing power, for faster image processing.

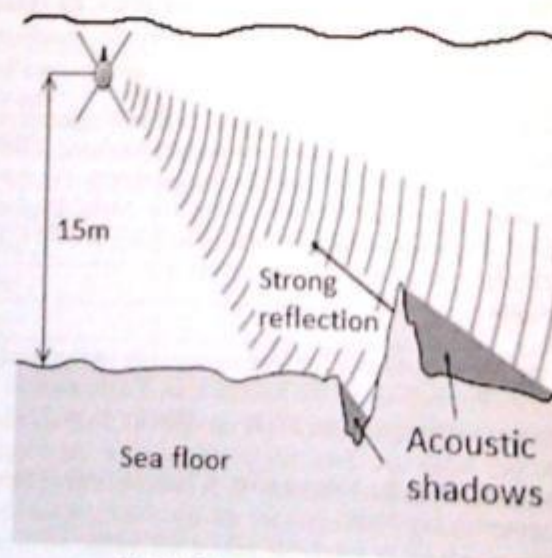


Fig. 1. Side scan sonar operation

2. Comparison of object detection methods in MatLab programming environment

To test the effectiveness of both image processing methods, a typical side scan sonar image was used. Sample image contains rocky sea bed and large object with a big acoustic shadow (see Fig. 2). Search object in rectangle 1 in Figure 2, emphasized image of object is in rectangle 2. Image size is 712x712 pixels and it covers area of about 220x220 meters. Also, it takes about 3 minutes to scan area of this size with typical side scan sonar.

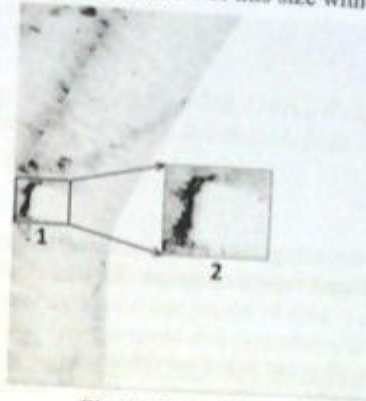


Fig. 2. Original image

First image processing method detects the object itself. Firstly, it converts the analyzed image to a numerical array (matrix form), which is the same size as image in pixels and values range from 0 to 255.

The method starts with image smoothing. The specific technique is to create small blocks of pixels in the initial array and compute the average values of pixels in that block. A circle was chosen as the best shape of the block with 4 pixels radius [1]. Next, the computed average values from specific areas are subtracted from original image matrix. Then, image is optimized to have 1% of totally black and 1% of totally white pixels. Other colors are also equalized. Then all pixels are converted to black or white [1]. So, all black pixels in the larger white areas are deleted, these pixels are caused by acoustic noise, they do not present any important shadow details. Further, all small white areas are also deleted, only areas that are bigger than 700 pixels are left, this number may vary, it depends on search object, sea floor, and image scale. These small white areas are shadows of rocks and sea floor inequalities or caused by sonar errors. Original size (1) and enlarged (2) result images are shown in fig 3, where is clearly emphasized search object. This image can be used as a target point for further object research.

Example of the code for MatLab software is provided below:

```
I = imread(file location);
backg = imopen(I, strel('disk',4));
I2 = I - backg;
I3 = imadjust(I2);
bw = im2bw(I3, 0.98);
g = imfill(bw, 'holes');
bw2 = bwareaopen(g, 700);
imshow(bw2);
```

Second image processing method is designed for low computing power controllers. It is designed to detect to detect larger acoustic shadows. Acoustic shadow appears as white pixels array in the image produced by the side scan sonar.

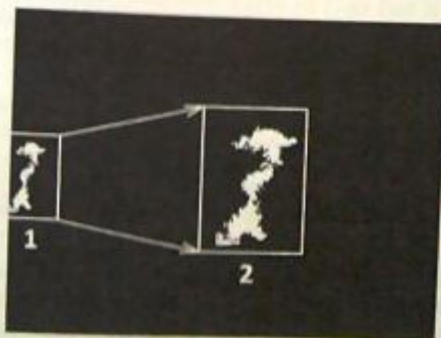


Fig 3. 1st method result

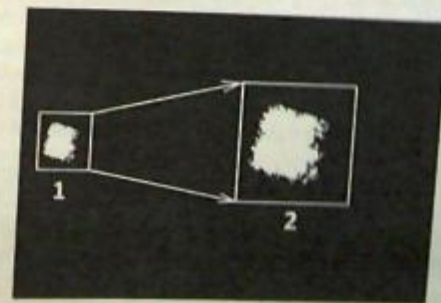


Fig 4. 2nd method result

White pixels on the ordinary sea floor image are very rare. The main point of this method is to detect "empty pixels" (large groups of empty pixels means large acoustic shadows). Same pixels conversion technique is used as in the

first method. The key difference between methods is that it does not subtract average pixels values from specific blocks. It also converts the image to a numerical array (matrix form) with only black and white colors [2]. Finally, it extracts areas of the picture that are bigger than 700 pixels, this number may vary depending on the sea floor texture and items that researcher want to detect. Original size (1) and enlarged (2) result images are shown in fig 4, where is clearly emphasized shadow of the search object. This image can be used as a target point for further object research.

Example of the code for MatLab software is provided below:

```
I = imread('file location');
bw = im2bw(I,0.985);
bw2 = bwareaopen(bw, 700);
g = imfill(bw2, 'holes');
imshow(g);
```

3. Computational results

Both image processing methods were compared using various computers (see table 1). The main reason for this testing was to check which method uses less computing power. Three different computers were used:

Table 1.

Test hardware specifications							
Computer type	CPU	CPU Cores	RAM	Processing time for 1 st method, s	Deviation	Processing time for 2 nd method, s	Deviation
Workstation 1	Pentium, 2.8GHz	1	768MB	0,38	0,01	0,26	0,01
Workstation 2	Core2, 2.5 GHz	4	4GB	0,1	0,02	0,07	0,01
Laptop	Turion, 1.8 GHz	2	3GB	0,24	0,01	0,12	0,01

Test was run on each computer 10 times. Average values were calculated and shown in table 1. Results are very consistent; difference between computational results is comparatively small. All tests ran on 32-bit Matlab 2010b software. Data processing results suggest that the 2nd method is up to 2 times faster to compute than first one, using all test hardware. If using the low processing power devices, we recommend using the 2nd method as it requires less computing power. Therefore, the main result suggest that indeed it is very important to analyze sonar images as fast as possible (more time efficient) when compared to real-time sea floor sonar scan which take up to 3 minutes.

4. Conclusions and future work

Both methods show good results when compared to more general methods used nowadays. Object detection is a complex method that requires more computing power. On the other hand, acoustic shadow detection is more time efficient. It is highly recommended to use second method for fast primary processing to detect large objects in high resolution images. Depending on the search object, both methods may be integrated and applied for specific tasks. For best results both methods are suggested for use as they look for areas where both, object and acoustic shadows appears.

Both methods are fast enough to be used in real life image processing application with modern computers. Real time scanning takes about 3 minutes, processing of the same area images takes much less time. Overall second method is more optimized for all kinds of terrains and objects. Parameters in first method has to be constantly adjusted for each situation, that makes it difficult to use.

Future work includes analysis of the shadow detection method application in low processing power controllers for real time image processing and to create specific complex algorithm that will use data from accelerometers and directional sensors for image smoothening.

Acknowledgement

The authors thank the Project „LLII-061 Development of joint research and training Centre in High Technology Area“(Latvia-Lithuania Cross Border Cooperation Programme Under European Territorial Cooperation Objective 2007-2013 Subsidy Contract No: LV_LT/1.1./LLII-061/2010/) as well as the Klaipėda University Coastal Research and Planning Institute for letting us use their research data funds.

References

1. Trucco A., Garofalo M. Processing and Analysis of Underwater Acoustic Images Generated by Mechanically Scanned Sonar Systems. IEEE Transactions on Instrument and Measurement, vol 58 (7), 2009. p. 2061-2071.
2. Gonzalez R. C., Woods R. E., Eddins S. L. Digital Image Processing using MATLAB. New York, 2006.
3. Penny Probert Smith, Konstantinos Zografos. Sonar for recognizing the texture of pathways, Robotics and Autonomous Systems, 2005, no. 51, p. 17-28.

Application of Speed Correction Algorithm for Trains Monitoring in an Intelligent Transport Control System

A. Andziulis*, J. Harja**, T. Eglynas*, S. Jakovlev*, V. Beniušis*, T. Lenkauskas*

*Klaipėda University, Bijunu str. 17, LT-91225, Klaipėda, Lithuania, E-mail: arunas.lik.ku@gmail.com, tmse@inbox.lt

**Ventspils Augstskola, Inženieru str. 101, LV-3600, Ventspils, Latvia, E-mail: janis.harja@venta.lv

Abstract

In this paper, we present an intelligent transport control system for real time monitoring of train movement. It uses agent technologies for correction of train speed (train scheduling) and therefore position with overall prediction in real time. Also currently received data and statistical information is analyzed. Computational results indicate the effectiveness of the proposed technique with statistical confidence limits for the compared analyzed data.

KEY WORDS: intelligent transport control system, train scheduling, real time monitoring, agent technology.

1. Introduction

In today's fast moving world, more and more attention is paid to security and quality. One of the most important aspects of security is constant monitoring (knowing the exact train position) of trains in real time. Automatic position detection system, will not only increase the security, but also will make transport process more agile.

Movement of trains depend heavily on such aspects as: technical equipment, human factor, external environment and etc. Such movement can be disruption during a simple communication loss problem, operator (later described as users) false decisions, or even weather changes. Therefore, the critical changes in the train schedule can result in critical situation on the road and etc. Therefore, a real time error correction system for the already scheduled movements is very important. Today, such error correction is done manually and almost always it is out of date when the changes are applied. Centralized movement control software tool is generally used, but all the corrections are made manually. But here the computational efficiency of all operators is comparatively low. Therefore, additional algorithms for such transport processes automation and other systems integration is vital.

During the last few years, many problems were faced when dealing with similar problems as discussed previously [1, 2, 6, 7, 9]. Most of them are related to scheduling problems in train transport. Mostly it is presented as a traditional optimization problem but with minor specific real life aspects which require modern optimization methods [3-5]. To design such real time monitoring and control intelligent control system, a feedback system is needed to be integrated to the currently used systems. Also, a model of the designed system needs to be proposed.

2. Data aggregation and analysis system

Fig. 1 shows a train control model proposed by Peng Wang [8]. Here the priority is made on scheduling and control of the network to increase the efficiency of the transport operations. That is why a timely conflict resolution and further trains rescheduling can decrease the loss of money and unforeseen tragical events. Such proposed model reflects the up-to-minute scheduling and control mechanism based on simulation system results [8].

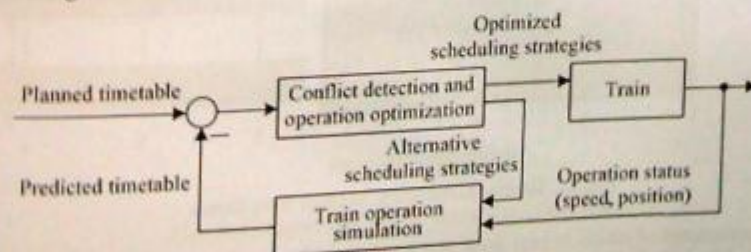
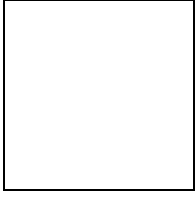


Fig. 1. The framework of real-time trains scheduling and control system based on model predictive control (RTPS) [8]

In order to make an affect on the scheduling procedures, a system needs to be proposed. Its integration into the existing trains control system will allow real time monitoring of trains traffic. Fig. 2 demonstrates the possible affect of the proposed system on real time data analysis of statistical data for scheduling procedures. Here when the train passes by the data collection centre, all the needed received information is analyzed and transferred to the train monitoring or conflict detection (emergency action) system.



deviations occur from the planned times (also see Table 1 for deviations and monitored times during experiment). The introduced algorithm programmed into intelligent system agent, collects data from registration points, compares them with the existing expected train times and alerts the users about the deviations from the planned times. Such deviation is made is commonly referred to as Standard in many aspects of data analysis.

As science, researchers commonly report the standard deviation of experimental data, and only effects that fall far outside the range of standard deviation are considered statistically significant – normal random error or variation in the measurements is in this way distinguished from causal variation. Therefore, the provided results suggest that almost constantly the mean values of the improved algorithm are in a higher confidence interval with 2 standard deviations compared to the programmed Siemens controller performance (see Fig. 4 and Table 1). The received deviation examination is statistically important.

Experimental and programed data comparison

Table 1.

Registration point	1	2	3	4	5	6	7	8
Improved algorithm performance average in time t_{av} [s]	5,4	5,7	9,4	3,6	9,8	7,9	6,0	6,0
Siemens controller constant performance in time t_{cs} [s]	5,1	5,4	10,1	2,6	8,6	7,4	5,3	5,9
Standard deviation for improved algorithm performance in time τ , [s]	0,36	0,48	1,13	0,44	1,00	0,32	0,39	0,32

Fig. 4 also displays comparison between mean values from 20 computations for each registration point and Siemens controller performance using real data from modelled intelligent train control system (see Fig. 5). Results indicate that the use of additional changes into the existing algorithms can in some cases quite significantly improve the existing control methods with increase in time efficiency. With 95% confidence one can say that in most cases: from 4th to 7th registration points, the proposed algorithm performed better, leaving the 2 standard deviation area. That means that the proposed algorithm is statistically significant in these cases. In other registration points, the algorithm performed similarly to the Siemens controller initial performance. That means that in these registration points, there is no difference between which algorithms to use. But one should have more analysis.

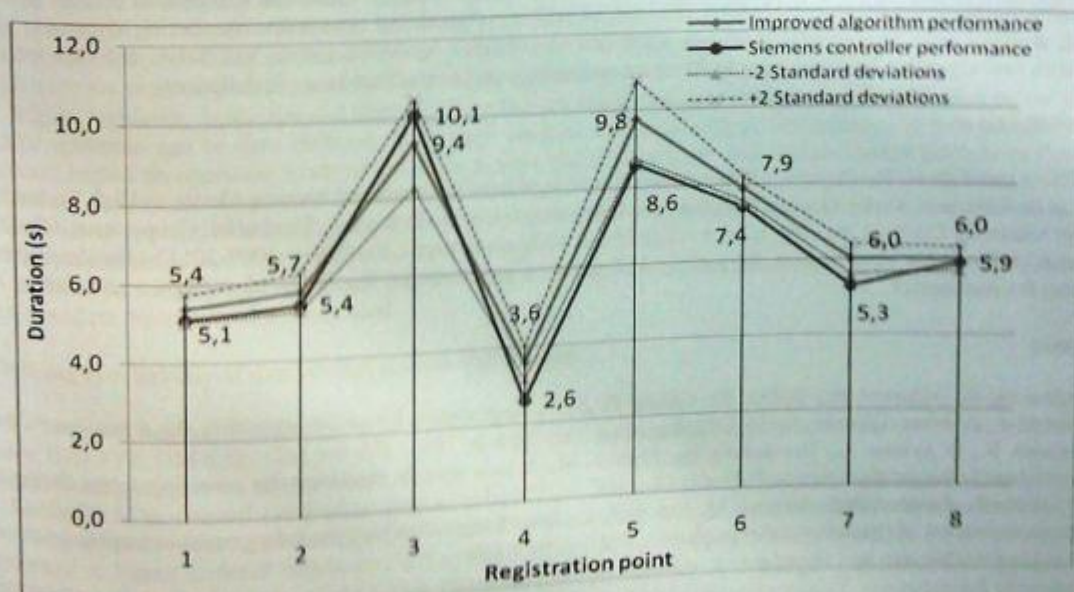


Fig. 4. Comparison of the computational data

The modelled train is controlled using SIEMENS S7-1200 programmed controller, which is set to pursue the programmed time. Similar system is used in real train systems and time plans are made manually by operators. Therefore, even a slight mismatch in planned time could result in serious problems or even alert situations.

Our examined controller had two roles: train operators and observers. When the train information reaches the controller, it automatically detects backwardness from the planned timetable, also preserving the data in a text file. Such data was used in the experiment which results are provided above. It should be mentioned that the use of such controller functionality is in real time. Therefore, the additional use of the provided agent system is also in real time. And that is why the use of such intelligent control system is worth using in real life applications with a vision of a future.

Computations were made in model seen on Fig. 5, where each train makes a full cycle in average 1 minute. During the cycle 8 hardware registration points are established which collects the needed data and makes correction to the train speed accordingly to the agents provided computed correction. The provided Fig. 5 demonstrates the use of 4 registration points from such point of view in 3-D.



Fig. 5. Intelligent train control system prototype (model).

4. Conclusions and future work

Such improvements integration into process control is an innovative decision which should be further validated at individual case studies in other real life applications where at each case more than 95% confidence interval needs to be observed. Agents are needed to be taught to correct the algorithm with each new iteration. Therefore, for statistical purposes larger amounts of data are needed to be processed to have clear vision of the system overall positive functionality. Overall, the development and application of such intelligent real time trains monitoring systems is a key objective in many countries worldwide. Future work includes further modelled system validation and received data mining with new algorithm improvements to the agent technology and overall hardware installations.

Acknowledgement

The authors thank the Project „LLII-061 Development of joint research and training Centre in High Technology Area“(Latvia-Lithuania Cross Border Cooperation Programme Under European Territorial Cooperation Objective 2007-2013 Subsidy Contract No: LV_LT/1.1/LLII-061/2010/), also Project “MOBAS 2009-2011” (The development of a virtual information environment for mobile and wireless services) for the financial support while writing and publishing the manuscript.

References

1. Caprara A., Monaci M., Toth P., Guida P. L. A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics* 154, 2006. p. 738 – 753.
2. Corman F., D'Ariano A., Pacciarelli D., Pranzo M. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B* 44, 2010. p. 175–192.
3. D'Ariano A., Pacciarelli D., Pranzo M. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research* 183, 2007. p. 643–657.
4. D'Ariano A., Pranzo M., Hansen I. A. Conflict Resolution and Train Speed Coordination for Solving Real-Time Timetable Perturbations. *Transactions on intelligent transportation systems*, vol. 8, 2007. p. 208 – 222.
5. D'Ariano A. Pacciarelli D. Pranzo M. Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research Part C* 16, 2008. p. 232–245.
6. Feng Li, Ziyao Gao *, Keping Li, Lixing Yang. Efficient scheduling of railway traffic based on global information of train. *Transportation Research Part B* 42, 2008. p. 1008–1030.
7. Smoczek J., Szpytko J. A mechatronics approach in intelligent control systems of the overhead traveling cranes prototyping. *Information technology and control* Vol.37, 2008. No. 2. p.154 – 158.
8. Peng Wang, Yonghua Zhou, Jiajie Chen, Yangpeng Wang, Pin Wu. Real-time Train Scheduling and Control Based on Model Predictive Control. *Second WRI Global Congress on Intelligent Systems*, 2010. p. 47 – 50.
9. Xuesong Zhou, Ming Zhong. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B* 41, 2007. p. 320–341.

Latvia has been a member of the European Union for 10 years. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country.

Latvia has been a member of the European Union for 10 years. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country.

Latvia has been a member of the European Union for 10 years. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country.

Latvia has been a member of the European Union for 10 years. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country.

Latvia has been a member of the European Union for 10 years. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country.

Latvia has been a member of the European Union for 10 years. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country.

Latvia has been a member of the European Union for 10 years. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country. The structure and provision of higher education in Latvia has changed significantly in the last 20 years in a context of the development of the country.

TVI

need for accumulation appliances increases in case third if we have a real consumption graph with the unumption factor β decreases to 0.75, the difference at the shape of the consumption curve affects the t; consequently, when changing the shape of the e accumulation equipment capacity.

A power solution, it will decrease the accumulation when $A = 1.0$ and $\beta = 1.0$. But in the case of lower A ant capacitances relating to wind+solar solution is The major impact is made by reduction of β , as a pipment in case of wind+solar solution may even be solution on conditions $A = 0.95$ and $\beta = 0.75$. The use consumption factors and availability close to 1.

vey of the options for the choice of a storage device e due to reliable wind speed and solar data available data period, the higher the reliability of calculations. eds and adding solar energy source the necessary ery high. In the given example, when the average s solution wind+solar is 0.34 kW, the value of the nditions is over 100 kWh. If we need to provide a nable reliability, we multiply the capacity of a unit ue of the storage capacity will change accordingly. y if the reliability of supply is reduced, and could vailability 0.95. The question arises if the consumer periods (annually 438 h). By availability level 0.99 it eergy supply interruption a year.

rgy producers, consumers and storage equipments to n the necessary value of the storage capacity or the

"Wind Turbines. Part 2: Design Requirements for

epa J. Wind energy application problems in inland Special issue, 2008, pp. 169-180.

mik (Handbook of Estonian solar radiation climate) n).

wind power system probabilistic modelling for long-vol. 80, 2006, pp. 578-588.

y S. Design of isolated renewable hybrid power 4-1136.

ing small wind generators according to probabilistic s; 39-th International Symposium „Actual Tasks on .011, Opatija, Croatia, pp. 235-243.

onian farmlands, Dr. Lingjuan Wang (ed.). North IGR E-Journal, vol 7, 2005, pp. 1-10.

in South-Eastern Estonia. Investigation and Usage of e proceedings / V. Titi (ed.). 2005, Tartu, Estonia, pp.

/ of CFB boiler burning oil shale. Oil Shale, vol. 26, k A. The estimation of needed capacity of a storage rs. Oil Shale, vol 26, no 3S, 2009, pp. 283-293.

COMPENSATED SINGLE-PHASE RECTIFIER

Ivars Rankis¹, Janis Donins¹, Vytautas Beniusis²

¹Riga Technical University, Latvia; ²Klaipeda University, Lithuania
rankis@eel.ru.lv, jdonins@gmail.com, vytautas@yahoo.com

Abstract. Paper describes methods of rectified DC pulsation reduction adding a compensation node to a single phase rectifier. Device is related to converters of electrical energy and it can be used in power electronics. Purpose of single phase compensated rectifier is to reduce load voltage pulsations. Several methods of compensation are observed, load characteristics are compared to a simple rectifier. Load voltage is compensated using compensation switch. It is operated so that rectifier output voltage pulsation is minimal. This current pulsation reduction effect is achieved so that the elevated voltage node is operating as an additional power source for load. Operation of compensation switch is controlled by comparing of two voltages, comparing reference voltage with voltage on load. While these voltages are equal or load voltage is greater than reference voltage, load voltage is not compensated. As soon as load voltage less than reference voltage, the compensation current is increased in proportion of difference of compared voltages. Load voltage compensation process goes on continuously.

Keywords: DC, compensation, capacitor, pulsation, load characteristic, reference voltage.

Introduction

In many cases there is a need to rectify single-phase AC. In addition, for example, in some technical applications (single-phase AC electrified transport and other) power has many hundreds of kilowatts. High quality rectified voltage of most of the rectifier circuits has a rectifier bridge circuit, but also in the rectified voltage there is a high variable voltage component that does not provide high-quality load voltage. For the voltage shape improvement there is the capacitive filter widely used [1,3], which is charging at the moment when AC voltage reaches its peak and discharges when the output voltage decreases shifting the stored energy to the load thus equalizing the load voltage. While the rectifier with the capacitance filter improves the rectified voltage form, it has the disadvantage that its AC current has a very bad shape with large magnitude because charging takes place in a very short time interval.

When connecting the capacitor in parallel to the bridge rectifier output, the load voltage shape significantly improves. However, the capacitor capacity is large, especially if the load current is large. The filter capacity can be approximately defined as

$$C = \frac{I_{load} (0.25T + \frac{1}{\omega} \arcsin(1 - \Delta U'))}{\Delta U}, \quad (1)$$

where ΔU is voltage reduction of the amplitude value, T is half period $0.5T = 0.5/f$ while, but $\omega = 2\pi f$ ($\Delta U'$ is ΔU and U_m ratio).

If, for example, the load current is 100 A, $\Delta U = 20$ V, $U_m = 312$ V, $f = 50$ Hz, then the necessary capacity of the capacitor is 44250 μF . Such a capacitor, while half-period is charging in a very short time with a current amplitude of more than 1.5 kA and the current THD = 1.66, i.e., results in a very bad shape of the current.

The form of network current can be significantly improved involving an inductance into the AC power side [2]. This inductance can be approximately calculated as

$$L = \frac{0.63 \cdot \Delta U \cdot 2(0.25T - \frac{1}{\omega} \arcsin(1 - \Delta U'))}{1.7I_{load}}, \quad (2)$$

and in case of this example, its inductance must be 29 mH having a RMS current of about 120 A and saturation current near 200 A. Although such coils and filter capacitor dimensions and weights are large, this significantly degrades the efficiency of the system.

These deficiencies can be prevented adding to a simple rectifier an elevated voltage compensation junction, circuits can be created in a number of ways, some of which are observed in this study, and the solution is called the single-phase compensated rectifier. This rectifier output curve compared to

the rectifier, which has only the capacitive filter, is less dropping within incident of load current rise and within a certain range of load current growth, the load voltage remains constant.

Development of single-phase compensated rectifier

The compensated rectifier is developed providing of an additional direct voltage source, a voltage just above the $U_m(1-\Delta U)$ value, and connecting it to the load through the transistor VT that is operating in emitter follower mode with the emitter base steering voltage $U_m(1-\Delta U)$ (Fig. 1). When the load voltage instantaneous value is slightly below the $U_m(1-\Delta U)$, the transistor opens and the compensation voltage will be connected to the load by maintaining the load voltage instantaneous value at the reference voltage $U_m(1-\Delta U)$ level. Using the compensated rectifier, part of the load power is provided from the unregulated bridge rectifier and part from the compensation unit.

The operation time of each of the sources is depending on the level of ΔU : if it is higher, the source of compensation operates longer:

$$t_{comp} = \frac{2}{\omega} \arcsin(1 - \Delta U) \quad (3)$$

but the operation time of the main rectifier is $t_{pr} = 0.5T - t_{comp}$. Curves of $t_{comp} = t_{comp} - 2/f(\Delta U)$ and $t_{pr} = f(\Delta U)$ at frequency $f=50$ Hz are shown in Fig. 2. a. As it can be seen both operation times are equal at $\Delta U=0.29$.

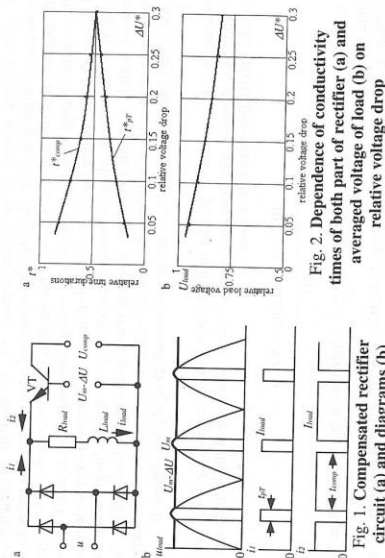


Fig. 1. Compensated rectifier circuit (a) and diagrams (b)

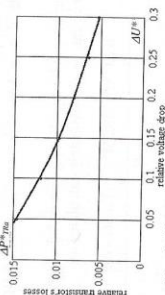


Fig. 3. Dependence of relative losses of transistor on relative voltage drop

Load voltage average value can be determined as

$$U_{avg} = \frac{U_{load}}{U_m} = 1 - \Delta U + 0.63 \cdot \Delta U \cdot t_{pr} \quad (4)$$

This curve, depending on the ΔU at $f=50$ Hz is depicted in Fig. 2. b. As it can be seen as ΔU goes higher, U_{avg} goes lesser and at $\Delta U=0.3$ it is 0.7956. At the same time it should be noted that the main rectifier bridge rectified DC voltage ratio to U_m is 0.6338, i.e., lesser than in the compensation system at $\Delta U=0.3$.

The transistor VT of the compensation circuit operates in the emitter follower mode, i.e., it has large power losses. When the transistor is conducting, the collector-emitter voltage is $U_{CE} = U_{comp} - U_m + \Delta U$, where U_{comp} is a compensation source DC voltage, which must be greater than $U_m(1-\Delta U)$, i.e., the reference voltage. Current, conducted by the transistor is I_{comp} . So the average transistor dissipated power is

$$\Delta P_{T_{VT}} = I_{comp} \cdot (U_{comp} - U_m + \Delta U) \cdot t_{comp} = \frac{(U_m - \Delta U + 0.63 \cdot \Delta U \cdot t_{pr})}{R_d} \cdot (U_{comp} - U_m + \Delta U) \cdot t_{comp} \quad (5)$$

In order to reduce power losses, it is rationally to adopt U_{comp} only slightly (by 2-5 V) greater than the reference voltage $U_m(1-\Delta U)$. Introducing the base capacity U_m^2/R_{load} , the transistor relative loss expression will be

$$\Delta P_{T_{VT}}^* = (1 - \Delta U + 0.63 \cdot \Delta U \cdot t_{pr}) \cdot (U_{comp} - 1 + \Delta U) \cdot t_{comp} \quad (6)$$

where it would be appropriate to accept $U_{comp} = 1.02(1-\Delta U)$. Basing on this assumption, at $f=50$ Hz calculated curve $\Delta P_{T_{VT}}^* = f(\Delta U)$, displayed in Figure 3. As it can be seen, at ΔU increase the relative loss decreases. At $\Delta U=0.3$ the relative loss is 0.0055, or at $U_m=312V$, $R_{load}=2\Omega$, it will be 268 W. Noticing that the power transistor must be rated for at least about 150A and the loading capacity will be almost 31 kW, the dissipated power will be small.

The main rectifier must be calculated on the average load current

$$I_{average} = \frac{U_m - \Delta U + 0.63 \cdot \Delta U \cdot t_{pr}}{R_d} \quad (7)$$

or

$$I_{average}^* = \frac{I_{average} \cdot R_d}{U_m} = 1 - \Delta U + 0.63 \cdot \Delta U \cdot t_{pr} \quad (8)$$

While the compensation source average current

$$I_{2average}^* = (1 - \Delta U + 0.63 \cdot \Delta U \cdot t_{pr}) \cdot t_{comp} \quad (9)$$

but this source power will be

$$P_{comp}^* = \frac{P_{comp} \cdot R_{load}}{U_m^2} = I_{2average}^* \cdot 1.02 \cdot (1 - \Delta U) \quad (9)$$

This way, at $\Delta U=0.3$, the compensation source should be calculated on 0.281 unit of the relative power, or, for example, at $U_m=312V$, $R_{load}=2\Omega$ compensation power capacity must be 13.7 kW.

Development of compensation source

Such source can be created as a bridge rectifier with the connected capacitor-reactor filter at its output; the compensation source will be able to feed from the same AC as the main rectifier (Fig. 4). The required coil inductance and capacitor capacitance have to provide that instantaneous meanings of the capacitor C_p voltage will be more as $U_m(1-\Delta U)$. Calculation of such filter approximately [2] can be realized at approach that the values of the capacitor voltage are decreasing linearly in the way of consuming from the junction compensation current (Fig. 5).

$$U_{cp} = U_{cm} + \Delta U \cdot t_{pr} \cdot \frac{I_{comp} \cdot t}{C} \quad (10)$$

where U_{cm} is the minimal value of voltage, ΔU_{cp} is the voltage difference between the maximum and minimum values, but the time t is between 0 and t_{comp} . In a similar way the capacitor voltage is rising in the way of operation of the main rectifier:

$$u_c = U_{cmin} + \frac{I_{comp} \cdot t}{C} \quad (11)$$

where I_{ca} is an averaged value of the capacitor current in the way of operation of the main rectifier when t is between 0 to t_{pr} .

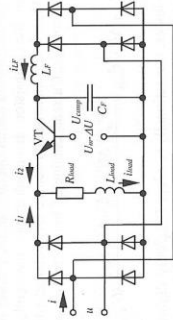


Fig. 4. Possible scheme of realization of compensation source

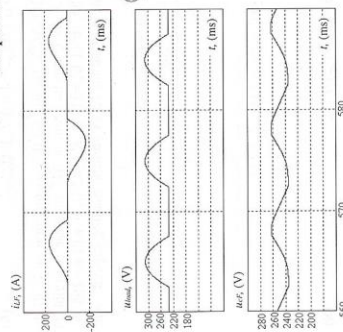


Fig. 5. Voltage and current diagrams of filter on Fig. 4

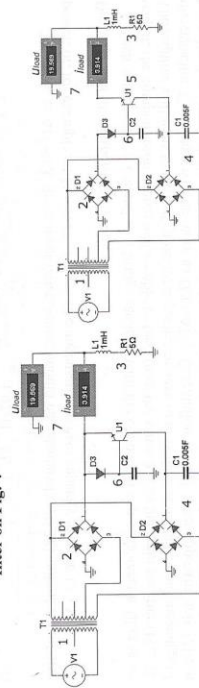


Fig. 7. Compensated single-phase rectifier with the additional elevated voltage rectifier and capacitance

Accepting necessary ΔU_{cr} from (10) the capacitance can be found:

$$C = \frac{I_{load} \cdot t_{comp}}{\Delta U_{cr}} \quad (12)$$

but from (11) - the necessary averaged current of capacitor charging is obtained

$$I_{ca} = \frac{\Delta U_{cr} \cdot C}{t_{pr}} \quad (13)$$

As it can be seen from Fig. 5 this current is slightly smaller as magnitude of the reactor current I_{cm} which is about 1.3 times bigger. Inductance of the coil can be found from a simplified equation for the capacitor charging circuit

$$L_r \frac{I_{ca}}{t_{pr}} = 0.63(U_{in} - U_{amb}) - 0.5\Delta U_{cr}$$

Here from

$$L_r = \frac{[0.63(U_{in} - U_{amb}) - 0.5\Delta U_{cr}] \cdot t_{pr}^2}{1.3I_{ca} \cdot t_{comp}} \quad (14)$$

It is assumed that current through the reactor is passing 1.2 times longer than main rectifier operation interval. RMS value of the reactor current can be found as:

$$I_{rd} = I_{ca} \sqrt{2.4 t_{pr} \cdot f} \quad (15)$$

If, for instance, $\Delta U_{cr} = 0.3$, $R_{load} = 2 \Omega$, $I_{load} = 125$ A, $f = 50$ Hz, $U_{amb} = 230$ V, $\Delta U_{cr} = 35$ V, then $L_r = 1.05$ mH, $C = 17100$ μ F, but RMS current of the reactor is about 90 A. If the compensation scheme is not used in the case of the rectifier with the filter must be $L = 1.45$ mH with RMS current 140 A. As the mass of the reactor is proportional to $L \cdot I_{rd}^2$ it should be possible to state that the reactor of the filter for the compensated scheme will be three times lighter. But the capacitance of the filter capacitor for the compensated scheme has to be $C = 17100$ μ F when in case of ordinary rectifier only 9656 μ F. Rising of the capacity is need because providing condition $U_{amb} > U_{rd}(1 - \Delta U_{cr})$.

It is possible to apply another scheme with the capacitor type voltage doubler (Fig. 6). With the numbers the following elements are indicated: 1 – transformer secondary winding with a centre tap; 2 – main rectifier bridge; 3 – active inductive load; 4 – diode-capacitor voltage doubler; 5 – compensation transistor; 6 – reference voltage unit, which is presented as diode and capacitor; 7 – *Multisim* virtual instruments to collect data for load characteristics of the system.

In Fig. 7 another version of the compensation voltage source is presented when it is possible to apply a transformer with elevated value of voltage of the secondary part of the winding. In addition, in this case, the transformer secondary winding has the mid-point output, but the compensation rectifier is connected to the transformer output voltage of about 40% higher than the output at which it is connected to the main rectifier. The action differs only in that the load is compensated by the lower voltage level and simpler compensation accumulator than in the case with the voltage doubler. The single-phase compensation rectifier shown in Fig. 8 is similar to the rectifier in the previous case, but the difference is that the power rectifier with the capacitor 6 performs only the reference voltage function. The outer curve of this circuit is shown as 3rd in Fig. 7 and 8.

Investigation of load characteristics

For the above mentioned schemes simulation programs have been used to obtain their load characteristics, i.e. dependence of the load voltage on load current. This was done using computer programs *PSIM* and *Multisim*. The characteristics are presented in Fig. 9 and 10. The load characteristics of the system with the voltage doubler are shown as 2nd in Fig. 7 and 8. The load characteristics for the circuit with additional elevated voltage are shown as 1st in Fig. 9 and 10. The load characteristics for the scheme 8 are presented as 3rd. The characteristics 4 are for the ordinary rectifier with the capacitor at output.

The comparison of the efficiency of the rectifier compensation option to the simple rectifier with the capacitance filter load characteristics of four rectifier circuits is shown in Fig. 9 and 10. The

characteristics are taken by changing the rectifier load gradually from 0 to large values and collecting data of average current and voltage measurements. All observed circuits had been measured at each load value but the load current and voltage average value reading is performed and the data are summarized in Fig. 9 and 10.

The compensated rectifier circuits were simulated according to the pictures illustrated in Fig. 6, 7, 8, all compensated rectifier circuits are without the main rectifier capacitor. All of the compensated rectifier circuit compensation accumulator capacitances are chosen equal to the simple rectifier capacitance, this way the load curves shown in Fig. 9 and 10 are comparable.

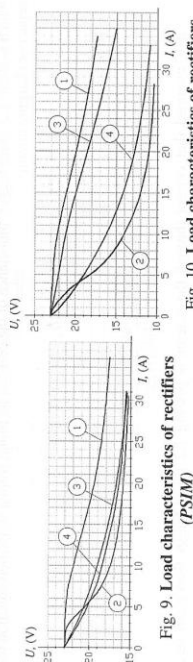


Fig. 9. Load characteristics of rectifiers (PSIM)

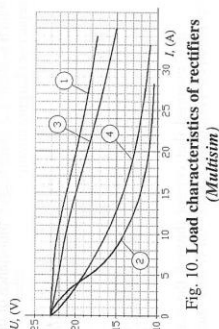


Fig. 10. Load characteristics of rectifiers (Multisim)

Estimating the load curves displayed in Fig. 9 and 10 it can be seen that the characteristics 1 and 2 have a critical value of the load to which they provide a very stable voltage – 1st curve to ~5 A, but 2nd curve ~2 A. It follows that the compensated rectifier with additional elevated voltage rectifier and capacitance shown in Fig. 7 gives the highest efficiency. Shown in Fig. 6 – the circuit single-phase rectifier with the voltage doubler – the efficiency is lower, the load curve provides a stable voltage to the load current value of 2 A, and more increasing current value of the load voltage drop becomes fastest of all the reference rectifier options, and the circuit negative characteristic is its complex construction.

The compensation rectifier circuit in Fig. 8, the load curve 3rd in Fig. 9 and 10 is similar to that of a simple rectifier with the capacitive filter circuit and the load curve 3rd in Fig. 9 and 10 the output average voltage value decreases in all areas where the output current average value is rising.

Conclusions

1. A simple rectifier with the filter capacitance load curve is decreasing at any load current rise in the range.
2. Any compensated rectifier circuit option load curves are less dropping comparing to the simple rectifier's with the capacitive filter load curve. The compensated rectifier with the additional elevated-voltage rectifier bridge and storing capacitance in a certain range of the load current rise- the time voltage value remains constant.
3. The most effective compensated rectifier option is the circuit with the elevated-voltage rectifier bridge and accumulator capacitance, its load curve shows the best characteristics of the voltage source among the observed circuits, as well as the technical construction is simple.

Acknowledgements

The study has been produced with the support of the project: Cross border JRTC, Interreg LAT-LIT, No: LV-LT/1.1/LLH-061/2010/.

References

1. Чижикова И. М., Руленко В. С., Сенько В. И., Основы преобразовательной техники, "Высшая школа", 1974, 426 стр. (In Russian).
2. Rodionova I., Rankis I. Operational Regimes of Input Filters// Electronics and Electrical Engineering.- Kaunas: Technologija, 2006 N 4(60), pp. 33-36.
3. Mohan N., Undeland T., Robbins W. Power Electronics: Converters, Application and Design. - NY: John Wiley and sons, 1989, 667 p.

HARDWARE AND SOFTWARE ENVIRONMENT FOR EVALUATION OF CONTROL ALGORITHMS AND STRATEGIES OF HYBRID POWER SYSTEMS

Vitalijs Osadskis, Aldis Pecka, Aldis Lojans

Latvia University of Agriculture

vt@tvetnet.lv

Abstract. This article presents an environment for design and evaluation of control algorithms and strategies for hybrid power systems, which use renewable energy sources. The environment relies on Hardware In the Loop (HIL) and Software In the Loop (SIL) approaches and allows to use together pure software Matlab Simulink models, physical models and real devices in non-real and real-time simulation modes using common hardware and software tools. The control system itself is implemented in Microsoft .NET environment using object-oriented techniques as a common application for both simulation modes. In order to show the capabilities and performance of the simulation environment renewable energy equipment block library in Simulink and microcontroller-based physical models of DC converters and load have been designed.

Keywords: modeling, hardware and software in the loop, control system, hybrid power system.

Introduction

Instability over a period of time is the main disadvantage of wind and sun, the most conventional renewable energy sources. It is hard to provide a consumer with electrical energy of good quality, especially in isolated small-scale power grids. Therefore so-called hybrid power systems or systems with distributed energy generation are used. Hybrid power systems (HPS) combine several mutually complementing renewable sources with energy storage and back-up powering devices in a single power grid. This allows compensating both short-term fluctuations and transients and long term unavailability of renewable energy sources. Such systems can be implemented with a connection to public power grid and can be used to supply power to the consumer during brownouts and blackouts in so-called islanding mode. Fully autonomous designs are also in use. HPS with distributed generation and energy storage along with cogeneration and multigeneration have begun to play a significant role in energetic in the past decade and it has a tendency to develop in future [1].

There are several possible ways of combining generators, accumulators and loads in a single system: central DC bus, central AC bus and composite architecture where several buses are connected using power interfaces. Use of unified power interfaces also increases modularity of hybrid power system. This results in simpler design, increased reliability, interchangeability of individual modules and reusability, which decrease the costs of implementation [2]. In a general case power interfaces handle the power flow from generators to loads and thus are considered as main final control elements or actuators in the control loop of hybrid power systems.

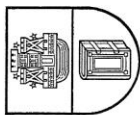
Because of nature or renewable energy a hybrid power system is distributed and stochastic and its configuration can vary significantly depending on the site. Designing control systems in such case is a time-consuming and expensive task; therefore various modeling techniques are extensively used. A variety of mathematical and simulation models are developed, which focus mainly on interaction of individual components of HPS in transients and static modes [3]. These models help to deal with problems related to stability, reliability and behavior in typical solutions, like the wind turbine-diesel generator, battery or wind-solar-battery etc, which are extensively studied by scientists and engineers, e.g. [4]. Theoretical and practical studies are being carried out to generalize control systems of HPS to allow achieve modularity also in a lower level and supervisory control, and data acquisition. For example, Awad et al. [5] discuss several concepts of microgrid (a type of HPS, which covers larger geographical areas) with different topologies and control strategies. Hajizadeh and Golkar study power flow in a battery/fuel cell system and focus on power interfaces [6].

Researchers also work on optimization models of HPS, for example [7], and optimal configuration searching, planning and strategy making software for use with renewable energy based distributed power systems. HOMER (Energy Modeling Software for Hybrid Renewable Energy Systems) and HARPsim (Hybrid Arctic Remote Power Simulator) [7] are to be mentioned.

In contrast to previously mentioned work where processes in HPS are studied from physical, energetic, tactical and strategic control perspective this article addresses problems related to design-



KLAIPĖDOS UNIVERSITETO
Sveikatos mokslų fakultetas



KLAIPĖDOS
UNIVERSITETAS

VIRTUALŲS INSTRUMENTAI BIOMEDICINOJE 2012

Tarptautinės mokslinės – praktinės konferencijos
pranešimų medžiaga

VIRTUAL INSTRUMENTS IN BIOMEDICINE 2012

Proceedings of International Scientific - Practical Conference



Klaipėda, 2012

Arvydas Martinkėnas, Renata Paškevičiūtė, Aurelija Podlipskyte, Giedrius Varoneckas. Prediction of maintenance of sinus rhythm after the conversion in patients with AF	89
Giedrius Varoneckas. Širdies ritmo variabilumo miego metu analizė kaip priemonė kardiovaskulinės funkcijos adaptacijos ir nuovargio įvertinimui	93
Giedrius Varoneckas. A tool for assessment of fatigue and adaptability of cardiovascular function using heart rate variability analysis.....	97
Igor Basov, Ilona Rachnova, Albinas Stankus. Širdies ritmo netiesinės dinamikos rodiklių palyginamoji analizė.....	101
Arvydas Martinkėnas, Albinas Stankus. Širdies ritmo variabilumo analizė POINCARÉ diagramų deriniais didinant užlaikymo žingsnius	106
Albinas Stankus, Giedrius Varoneckas, Lina Mažrimaitė, Olga Kyselova. Paros širdies ritmo pasiskirstymo dekompozicija laike.....	111
Dovilė Metrikytė, Vytautas Beniušis, Albinas Stankus. Elektrokardiogramos signalo perdavimas bevielio tinklo priemonėmis	117
Žydrūnas Lukošius, Dainius Aponkus, Albinas Stankus, Vaiva Strukčinskaitė, Timofėjus Jurevičius. Pulsinės bangos greičio dinamikos tyrimas kojose, keičiant jų padėtį.....	122
Lina Mažrimaitė, Albinas Stankus, Mantas Paulikas. Atskirų signalų išskyrimas iš daugiakanalio miego tyrimo įrašų EDF ir TDMs formatais	127
Julius Berneckis, Toma Dagenytė, Albinas Stankus, Olga Kyselova. Paros širdies ritmo aukšto dažnio filtravimas.....	131
Aurelija Livenis, Lina Mažrimaitė, Simona Malakauskaitė, Dovydas Imbrasas, Albinas Stankus. Žmogaus skalpo įsitempimo dinamika kaip aritmetinio testo markeris	135
Augustinas Dirmantas. Labview valdomų AD5933 ir mikrovaldiklio sistemos sukūrimas kūno kompleksinės varžos matavimui	141
Romualdas Šimatolskis, Albinas Stankus, Tomas Girkontas. Regėjimo efektyvumo kėlimo būdas ir tyrimas.....	145
Носовец О.К., Кисельова О.Г., Настенко М.А. Построение процентильных диаграмм «чсс – тисю» в гендер-однодринных группах для оценки состояния микроциркуляторной сети	149
Е.А. Настенко, Е.К. Носовец, М.В. Даниленко. Исследование возрастных изменений в системе кровообращения за процентильными диаграммами «возраст-давление» в гендер-однодринных группах	155
Матвийчук А.О., Киселева О.Г., Настенко Е.А., Чеховой Н.В. Математическая модель левого желудочка сердца с аневризмой	160
Чеховой Н.В., Киселева О.Г., Матвийчук А.О. Особенности генерации псевдослучайных последовательностей для имитации измерительных сигналов	164

BIOMEDICAL SIGNALS AND SENSORS

Eugenijus Kaniusas

Institute of Electrodynamics, Microwave and Circuit Engineering
Vienna University of Technology, Vienna,
Austria kaniusas@tuwien.ac.at

Sensing technologies in physiology gain a lot of importance for the assessment of the human functional state. The registered biomedical signals - referred to as biosignals here - are important not only for timeless classical applications concerning medical diagnosis and subsequent therapy, but also for future applications such as daily driver monitoring or nocturnal sleep monitoring. For instance, the diagnosis of sleep apnea, i.e., abnormal cessation of respiration during sleep, requires the monitoring of a complete set of sleep and respiratory variables with at least eight different sensors distributed over the entire body.

In order to adequately apply biomedical sensors and reasonably interpret the corresponding biosignals, a proper understanding of the physiologic phenomena involved, their influence on the registered biosignals, and the technology behind the sensors is critical [1]. Moreover, a nearly unlimited diversity of biosignals - as illustrated in Fig. 1 - emphasizes the need for a strategic approach in the genesis of biosignals, including a profound understanding of fundamentally different mechanisms in a biosignal's generation.

From a strategic point of view, biosignal generation involves the biosignal formation path from the biosignal source at the physiological level, to biosignal propagation in the body, to biosignal transmission in the sensor up to its conversion to a, usually electric, signal. To give an example, heart sounds, an acoustic biosignal, are created by the closure of heart valves, which constitutes the biosignal source. Sound attenuation in the thoracic tissue represents the propagation mechanism. Amplification and filtering of the heart sounds in the chestpiece (of the stethoscope) reflect biosignal transmission effects in the sensor, with biosignal conversion being performed by a microphone at the output of the chestpiece.

5. Stankus A. Kokybiniai širdies ritmo požymiai, įtakojantys jo modeliavimą. // *Biomedicininė inžinerija. Tarptautinės konferencijos pranešimų medžiaga* // *Kauno technologijos universitetas*. Kaunas. 2005, pp. 37-40.
6. Brožaitienė J., Stankus A., Bovina E. Relationships of Heart rate and Oxygen Consumption // *Proceedings of International Biomedical Engineering Conference*. – Kaunas. – 2005. – P. 107-110.
7. Mielke, M, Housh, TJ, Hendrix, CR, Camic, CL, Zuniga, JM, Schmidt, RJ, Johnson, GO. Oxygen uptake, heart rate, and ratings of perceived exertion at the PWCv02. *J Strength Cond Res* 2009 Jul;23:1292-1299.

Summary

TIME DISTRIBUTION ANALYSIS OF CIRCADIAN HEART RATE MODIFICATIONS

Albinas Stankus^{1,2}, Giedrius Varoneckas¹, Lina Mažrimaitė³, Olga Kyselova⁴

¹ Klaipėda University, ² Mechatronics Science Institute, ³ LSMU, ⁴ KU Medical technology department, ⁵ National Technical University of Ukraine

A new method for analysis of the functional state (active or passive) using RR intervals collected during 24-hour holter monitoring is presented.

*Tyrimą finansuoja Lietuvos mokslo taryba (sutarties Nr. TAP-LU-11-005). Tyrimas buvo įgyvendintas bendradarbiaujant su Klaipėdos universiteto Mechatronikos mokslo institutu, LSMU Elgesio medicinos institutu ir Ukrainos nacionaliniu techniniu universitetu.

ELEKTROKARDIOGRAMOS SIGNALO PERDAVIMAS BEVIELIO TINKLO PRIEMONĖMIS

¹Dovilė Metrikytė, ¹Vytautas Beniušis, ²Albinas Stankus
¹Klaipėdos universitetas, ¹Informatikos inžinerijos katedra,
²Mechatronikos mokslų institutas

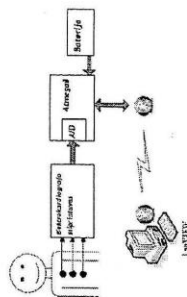
Ivadas

Sparčiai augant sveikatos problemoms auga ir medicinos įstaigų bei namų poreikis sveikatos priežiūros prietaisų. Dauguma sveikatos priežiūros prietaisų labai brangūs, reikalauja didelių sąnaudų, aparatu- ra gremėzdžiška, nepatogaus naudojimo ir ne kiekvienam prieinamos dėl aukštos kainos. Bluetooth - nauja technologija, naudojanti radijo ryšį trumpais atstumais siekiant pakeisti stacionarių ir nešiojamų elektroninių prietaisų sujungimus kabeliais į sujungimus radijo ryšiu. Ši technologija leidžia padaryti prietaisus patogiais, miniatiūriais. Nešiojamų sistemų dizainas, nuotolinė širdies veiklos stebėseną yra įvardijama kaip viena iš svarbiausių stebėsenos sistemų telemedicino- je [1, 2]. Darbe visas dėmesys skiriamas tik širdies ritmui ir jo perda- vimui. Aprašomas modeliujamas sistemos prototipas, padėsiantis sumažinti išlaidas nenaudojant kelių perdavimo kanalų duomenų įve- dimui ir stebėjimui. Suprojektuota virtuali sistema leidžia iš širdies elektrokardiogramos QRS komplekso aptikti RR intervalą.

Tikslas. Sukurti EKG duomenų bevielio tinklo perdavimo sistemą bei R dantelio identifikavimo programą, panaudojant mikrovaldiklį ir „LabView“ virtualius instrumentus.

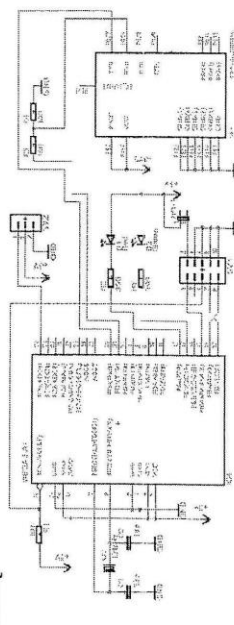
Metodika ir rezultatai

Projektuojama sistema susideda iš šešių pagrindinių komponentų: elektrodų, elektrokardiografo stiprintuvo, maitinimo bloko, mikroval- dikiio (Atmega8), bluetooth duomenų perdavimo sąsajos modelio, kompiuterio su programine LabVIEW įranga. Duomenų perdavimo sistemos principinė schema parodyta 1 paveiksle.



1 pav. EKG duomenų perdavimo sistemos principinė schema.

duomenų jungtis SV2 veda į EKG schemą. Signalas yra perduodamas į mikrovaldiklio analoginio/skaitmeninio keitiklio pirmąjį nuskaitymo kanalą.



2 pav. EKG matavimo sistemos principinė schema.

Išmatuoti duomenys yra perduodami kompiuteriui pasinaudojus beveik Bluetooth jungtimi. BTM222 modulis naudojamas duomenų konvertavimui ir perdavimui, o mini USB-Bluetooth adapteris naudojamas duomenų priėmimui kompiuterio pusėje. Mikrovaldiklis dirba 16 MHz takiniu dažniu, kad pasiektų tokį dažnį yra naudojamas išorinis kvarcinis rezonatorius.

Kadangi mikrovaldiklis naudoja 5V įtampos šaltinį, o bluetooth modulis 3.3V, todėl yra naudojami du įtampos reguliatoriai. Pirmasis LM7805 iš baterijos gaunamą įtampą mažina iki 5V, o antrasis LM7833 šią įtampą sumažina iki 3.3V. Prie abiejų įtampų reguliatorių yra įtampos filtravimo kondensatoriai, reikalingi sumažinti galimus įtampos svyravimus bei trikdžius. Dėl skirtingų

3 pav. Sistemos maitinimo blokas.

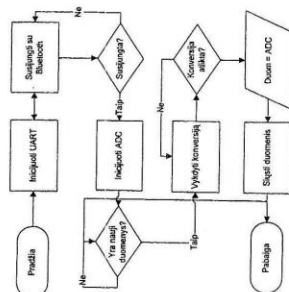
darbinių įtampų mikrovaldiklis su bluetooth moduli negali tiesiogiai bendrauti, nes mikrovaldiklis „mato“ perduodamus duomenis kurių įtampa yra nuo 0.5V iki 5V o bluetooth modulis mato tik iki 3.6V. Dėl šios priežasties yra reikalingas įtampos daliklis mažinantis mikrovaldiklio perduodamus 5V signalus į 3.3V signalus.

Kuriamos sistemos algoritmas pateiktas 4 paveiksle. Programos pradžioje sistema atlieka RS-232 sąsajos protokolo (UART) inicijavimą. Tada mikrovaldiklis susijungia su BTM222 moduli, po šio susijungimo yra sujungiami mikrovaldiklio bluetooth modulis su kompiuterio USB – Bluetooth adapteriu.

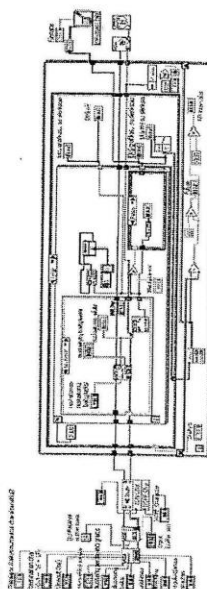
Paruošus sistemą duomenų perdavimui yra inicijuojamas analoginis – skaitmeninis signalų keitiklis (ADC). Kadangi duomenys yra perduodami 8 bitais, o ir matavimų tikslumui pakanka 8 bitų raiškos, todėl ADC keitiklis yra sukonfigūruojamas matuoti 8 bitų tikslumu.

Gavus naujus duomenis iš EKG schemos yra jungiamas ADC keitiklis, gauti duomenys yra nuskaitymi ir įrašomi į Duom kintamąjį. Iškart po matavimų duomenys yra persiunčiami kompiuteriui.

LabVIEW programa apima keletą etapų ir atlieka skirtingus veiksmus, ji suskirstyta į dvi dalis – signalo įvedimą ir RR intervalo skaičiavimą. Programos blokinė schema pateikta 5 paveiksle. Pirmiausia konfigūruojama nuoseklyji jungtis, kurioje pasirenkama VISA ištečių pavadinimas, informacijos perdavimo greitis, duomenų baitai, bitų užlaikymas, duomenų



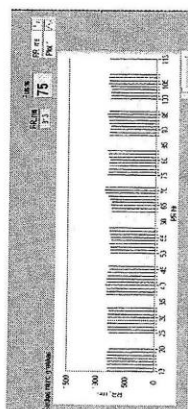
4 Pav. Sistemos algoritmas.



5 pav. Programos blokinė schema.

Norint surasti RR intervalą būtina surasti QRS kompleksą viršinės R dantelį. Norint vykdyti aukštesnių taškų detekciją pirmiausia elektrokardiogramos signalui reikia nurodyti slenkstį. Pikai, kurie yra žemesni už nurodytą slenkstį, grafiškai neatvaizduojami. Norint apskaičiuoti RR intervalą, būtina atlikti veiksmus su laiku, kuris yra saugomas iki tol, kol ateis kitas signalas. Abu šie signalai atimami vienas iš kito, tokiu būdu gaunamas RR intervalas.

6 paveiksle pateiktas RR intervalo skaitinių verčių atvaizdavimas.



6 pav. RR intervalo skaitinės vertės.

RR tai laikas tarp dūžių, kuris atspindi organizmo energetinius poreikius.

Išvados

Pateiktas EKG duomenų perdavimo modelis, kuris duomenis registruoja ir perduoda realiu laiku bevielio tinklo technologijomis. Gautas signalas toliau apdorojamas LabVIEW programa, kuri leidžia priimti duomenis realiu laiku, nuskaityti, stebėti, apskaičiuoti RR - laiką tarp dūžių ir atvaizduoti grafiškai.

Literatūra

1. D. Shih, H. Chiang, B. Lin. *An embedded mobile ecg reasoning system for elderly patients*. IEEE Transactions On Information Technology In Biomedicine, vol. 14, no. 3, pp. 854-865, 2010
2. D. Bansal, M. Khan. *A computer based wireless system for online acquisition monitoring and digital processing of ECG waveforms*. Computers in Biology and Medicine Vol.39, no. 4, pp.361-367, 2009

Summary

ELECTROCARDIOGRAM SIGNAL TRANSFIERES TO WIRELESS NETWORK DEVICES

¹Dovilė Metrikytė, ¹Vytautas Beniušis, ²Albinas Stankus
Department of Informatics Engineering, Mechatronics Science
Institute Klaipėda University

Due to extreme increase in medical problems, there is an immense need of medical care devices in medical institutions and home environment. Regarding this issue, a mean of transmission that can register records and transmit them in a real time mode by wireless technologies is the best possible model provided. A received digital signal is processed by LabVIEW VISA programme, which allows receiving the information, to read it and to calculate RR interval that is a time period between the heartbeats, and that reflects the energetic needs of the organism, and depicts it graphically.

5. Kompaktinis diskas

Kompaktinis diskas, kuriame pateikiami:

Darbo katalogas: „Vytautas_Beniušis/..“.

Darbo aprašymas: „../Vytautas_Beniušis _baigiamasis_darbas.docx“.

Darbo aprašymas: „../Vytautas_Beniušis _baigiamasis_darbas.doc“.

Darbo prezentacija: „../Vytautas_Beniušis _prezentacija.pptx“.

Darbo prezentacija: „../Vytautas_Beniušis _prezentacija.ppt“.