

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ. И. СӘТБАЕВ АТЫНДАҒЫ ҚАЗАҚ ҰЛТТЫҚ ТЕХНИКАЛЫҚ ЗЕРТТЕУ
УНИВЕРСИТЕТІ



SATBAYEV
UNIVERSITY

Автоматика және ақпараттық технологиялар институты

«Робототехника және автоматиканың техникалық құралдары» кафедрасы

Тәжібек Еркебұлан Ауданбекұлы

«Миниатюралық сонарды қолдана отырып, кедергілер мен жартастарды
анықтайтын роботты әзірлеу»

ДИПЛОМДЫҚ ЖОБА

5B071600 – Аспап жасау мамандығы

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ. И. СӘТБАЕВ АТЫНДАҒЫ ҚАЗАҚ ҰЛТТЫҚ ТЕХНИКАЛЫҚ ЗЕРТТЕУ
УНИВЕРСИТЕТІ



SATBAYEV
UNIVERSITY

Автоматика және ақпараттық технологиялар институты

«Робототехника және автоматиканың техникалық құралдары» кафедрасы

ҚОРҒАУҒА РҰҚСАТ
РТжАТК кафедра меңгерушісі
техника ғылымдарының кандидаты
Қ.А. Ожикенов
« 24 » мамыр 2022 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: «Миниатюралық сонарды колдана отырып, кедергілер мен
жартастарды анықтайтын роботты әзірлеу»

5B071600 – Аспап жасау мамандығы бойынша

Орындады



Рецензент
тех.ғылым кандидаты,
қауымдастырылған профессор м.а.

(Ғылыми атағы, дәрежесі)

Жаменкеев Е.К

қолы

аты-жөні

« 25 » мамыр 2022 ж.

Тәжібек Еркебулан

Ғылыми жетекшісі
тех.ғылым магистрі,
сениор-лектор

Бигалиева Ж.С.

« 26 » мамыр 2022 ж.

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ. И. СӘТБАЕВ АТЫНДАҒЫ ҚАЗАҚ ҰЛТТЫҚ ТЕХНИКАЛЫҚ ЗЕРТТЕУ
УНИВЕРСИТЕТІ



SATBAYEV
UNIVERSITY

Автоматика және ақпараттық технологиялар институты

«Робототехника және автоматиканың техникалық құралдары» кафедрасы

5B071600 – Аспап жасау



БЕКІТЕМІН

РТЖАТҚ кафедра меңгерушісі
техника ғылымдарының кандидаты
Қ.А. Ожикенов
« 24 » мамыр 2022 ж.

ТАПСЫРМА

дипломдық жұмысты орындауға

Білім алушыға Тәжібек Еркебұлан Ауданбекұлы

Тақырыбы: Миниатюралық сонарды колдана отырып, кедергілер мен жартастарды анықтайтын роботты әзірлеу

Университет ректорының бұйрығымен бекітілген №489-П/Ө 24.12.2021 ж.

Аяқталған жұмысты тапсыру мерзімі «25» мамыр 2022 ж.

Дипломдық жұмыстың бастапқы мәліметтері: кедергілерді анықтау және айналып өтетін роботтың жүйесін қарастыру. Сол жүйенің құрамын талдау. Ультрадыбыстық және инфрақызыл датчиктер.

Дипломдық жобада әзірленуге жататын мәселелер тізімі:

- Роботтың кедергілерді айналу алгоритмдерін зерттеу, түсіну
 - Кедергілерді анықтау, анықтаушы құрастырушылар.
 - кедергіні анықтайтын робот әзірлеу, жүру алгоритмін ұсыну.
- Графикалық материалдың тізбегі (міндетті сызбаларды дәл көрсете отырып):

9 слайд

Ұсынылатын негізгі әдебиеттер: 17 әдебиеттер тізімі


Дипломдық жобаны дайындау

КЕСТЕСІ

Бөлімдер атауы, әзірленетін сұрақтар тізбесі	Ғылыми жетекшіге ұсыну мерзімдері	Ескертпелер
Теориялық бөлім	10.02-11.03.2022ж	Орындалды
Бағдарламалық бөлім	14.03-18.04.2022ж	Орындалды
Зерттеу бөлімі	19.04-03.05.2022ж	Орындалды
Қорытынды бөлім	04.04-16.05.2022ж	Орындалды

Аяқталған дипломдық жобаға және оған қытысты бөлімдерінің кеңесшілері мен қалып бақылаушының

ҚОЛТАҢБАЛАРЫ

Бөлімдердің атауы	Ғылыми жетекшілер, кеңесшілер, (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қол қойылған күні	Қол
Қалып бақылаушы	Көшербай М.А. тех.ғылым магистрі	15.05.22	

Ғылыми жетекшісі



Бигалиева Ж.С.

Тапсырманы орындауға алған білім алушы



Тәжібек Е.А.

Күні

«__» мамыр 2022 ж.

АНДАТПА

Дипломның тақырыбы миниатюралық сонарды қолдана отырып кедергілер мен жартастарды анықтайтын роботты әзірлеу. Роботтардық әрекеті және алгоритмдері талданды. Кедергілерді анықтау және айналып өтетін роботтың жүйесін қарастырылды. Жүйе екі (инфрақызыл және ультрадыбыстық) датчиктен, Arduino микроконтроллерінен және тұрақты ток мотор қозғалтқыштан тұрады. Ультрадыбыстық және инфрақызыл датчиктер интерфейсті микроконтроллерге сигналдар жіберу арқылы робот жолындағы кедергілерді анықтау үшін жүзеге асырылады. Кез келген кедергіге жарық пен дыбыс толқындарын жіберу үшін роботтың алдыңғы жағына орналастырылған инфрақызыл және ультрадыбыстық датчик сонар модульдері арқылы кедергілерді анықтау тізбегі жүзеге асырылды.

АННОТАЦИЯ

Тема диплома - разработка робота, обнаруживающего препятствия с помощью миниатюрного сонара. Анализируются поведение и алгоритмы роботов. Рассмотрена роботизированная система обнаружения и обхода препятствий. Система состоит из двух датчиков (инфракрасного и ультразвукового), микроконтроллера Arduino и двигателя постоянного тока. Ультразвуковые и инфракрасные датчики используются для обнаружения препятствий на пути робота путем отправки сигналов на интерфейсный микроконтроллер. Для передачи световых и звуковых волн на любое препятствие была реализована схема обнаружения помех с использованием гидроакустических модулей инфракрасных и ультразвуковых датчиков, расположенных на передней части робота.

ANNOTATION

The theme of the diploma is the development of a robot that detects obstacles using a miniature sonar. The behavior and algorithms of robots are analyzed. A robotic system for detecting and avoiding obstacles is considered. The system consists of two sensors (infrared and ultrasonic), an Arduino microcontroller and a DC motor. Ultrasonic and infrared sensors are used to detect obstacles in the path of the robot by sending signals to the interface microcontroller. To transmit light and sound waves to any obstacle, an interference detection scheme was implemented using hydroacoustic modules of infrared and ultrasonic sensors located on the front of the robot.

МАЗМҰНЫ

Кіріспе	9
1 Бөлім	11
1.1 Кедергілерді өту алгоритмдері	11
1.1.1 Bug алгоритмдері	11
1.1.2 Dist-Bug алгоритмі	12
1.1.3 ИВА алгоритмі	12
1.1.4 Потенциалды өрістер әдісі (Artificial potential field)	13
1.1.5 Векторлық өріс гистограммасы әдісі (Vector field histogram)	14
1.2 Кедергілерді анықтайтын датчиктер	16
2 Құрастыру бөлімі	20
2.1 Кедергіні анықтау	20
2.2 Ақпаратты-өлшеу жүйесі	21
2.3 Құрастырушы элементтер	22
Қортынды	31
Қолданған әдебиеттер тізімі	32
Қосымша А	34
Қосымша В	48

КІРІСПЕ

Роботтар соқтығысудан аулақ бола отырып, роботты белгісіз ортада басқаруға арналған. Кедергілерден аулақ жүретін робот өз жолындағы кедергілерді танып, олардан аулақ болып, жұмысын жалғастырады. Қабырғаларды қадағалау, жиекті анықтау, сызықты қадағалау және т.б. сияқты өте танымал роботты навигация әдістері бар. Кедергілерді болдырмаудың неғұрлым жалпы және жиі қолданылатын әдісі жиекті анықтауға негізделген. Жиекті анықтауға негізделген кедергілерді болдырмау әдісінің кемшілігі - дәлірек өлшеуді қамтамасыз ету үшін робот кедергінің алдында тоқтауы керек. Барлық мобильді роботтар кедергіні анықтайтын және соқтығысуды болдырмау үшін роботты тоқтататын қарапайым алгоритмдерден бастап, роботқа кедергілерді болдырмауға мүмкіндік беретін күрделі алгоритмдерге дейін белгілі бір дәрежеде соқтығысудан аулақ болады.

Олар анықталғаннан кейін кедергілерді болдырмау алгоритмі роботты кедергіні айналып өтіп, бастапқы мақсатқа қарай қозғалысын жалғастыруы керек. Басқару алгоритмі навигация кезінде роботтың кедергі алдында тоқтамауын қамтамасыз етеді. Ультрадыбыстық датчик алда тұрған кез келген кедергіні анықтау үшін қолданылады және микроконтроллерге пәрмен жібереді. Сондықтан, роботтар навигация кезінде кейбір проблемаларды жеңе алады және соқтығысуды болдырмай жұмыс барысында бірқалыпты қозғала алады. Егер инфрақызыл датчикті пайдалансақ, инфрақызыл датчиктер инфрақызыл сәулелену арқылы объектіге дейінгі қашықтықты анықтайды. Сәуле нысанды анықтаған кезде жарық сәулесі шағылысудан кейін бұрышпен қабылдағышқа оралады.

Кедергілерді анықтау үшін қолданылатын әртүрлі датчиктер бар. Өте танымал датчиктердің бірі инфрақызыл датчиктер (ИК), ультрадыбыстық датчиктер, компьютерлік көру бөлігі ретінде пайдаланылуы мүмкін камералар, сонарлар. Ол мыңдаған нүктеден жүздеген нүктеге дейінгі қашықтықты өлшей алады. Роботты жобалау кезінде кедергілерді анықтау және болдырмау үшін ультрадыбыстық датчиктерді қолданамыз. Ультрадыбыстық датчиктер жиілік сигналдарын үздіксіз шығарады, кедергі анықталған кезде бұл сигналдар кері шағылысады, олар кейін сенсорға кіріс ретінде қарастырылады.

Ультрадыбыстық датчиктер өлшенген қашықтықты сантиметрмен немесе дюйммен шығара алады. Датчиктің сантиметрмен өлшеу диапазоны 0-ден 255 см-ге дейін, дюймдерде - 0-ден 100 дюймге дейін. Сенсор 3 см-ден (1,5 дюйм) аз қашықтықтағы нысандарды анықтай алмайды. Сондай-ақ, ол жұмсақ, тіндік және аз көлемді заттарға дейінгі қашықтықты өлшеу үшін жеткілікті тұрақты емес. Бұл режимде түрлендіргіш ультрадыбыстық импульстарды шығармайды, бірақ басқа ультрадыбыстық түрлендіргіштен импульстарды анықтауға қабілетті.

Тақырыптың өзектілігі. Бүгінгі таңда роботтар өнімділігі мен сенімділігінің жоғары деңгейіне байланысты көптеген салаларда қолданылады, бұл адамдарға көп көмектеседі. Соқтығысты болдырмау робототехникасы

кедергілерді анықтау және соқтығысуды болдырмау үшін қолданылады. Кедергілерді болдырмау роботының дизайны олардың міндеттеріне сәйкес бірнеше датчиктерді біріктіруді талап етеді.

Жұмыстың мақсаты: кедергілерді анықтау және айналып өтетін роботтың жүйесін қарастыру. Сол жүйенің құрамын талдау. Ультрадыбыстық және инфрақызыл датчиктер интерфейсті микроконтроллерге сигналдар жіберу арқылы робот жолындағы кедергілерді анықтау жүйесін келтіру қажет. Кез келген кедергіге жарық пен дыбыс толқындарын жіберу үшін роботтың алдыңғы жағына орналастырылған инфрақызыл және ультрадыбыстық датчик сонар модульдері арқылы кедергілерді анықтау тізбегін жасау керек.

Уақыт өте келе роботтар олардың қолданылуына байланысты әртүрлі пішіндерге ие болды және олардың өлшемдері үлкен 51 футтан микроскопиялық деңгейге дейін өзгерді. Роботтардың технологиялық дамуы кезінде олардың жұмыс істеуі үшін бір аспект маңызды болып қала берді, атап айтқанда ұтқырлық. Қазіргі уақытта «кедергілерді болдырмау» термині қолданылады.

Робототехника роботтың айналадағы объектілермен соқтығыспай, белгісіз орта арқылы қозғалу қабілетін білдіреді. Роботтардағы кедергілерді болдырмау әртүрлі орталарда маневр жасауда үлкен икемділікті қамтамасыз етеді және әлдеқайда тиімдірек болады, өйткені ол адамның тұрақты бақылауын қажет етпейді.

1 Негізгі бөлім

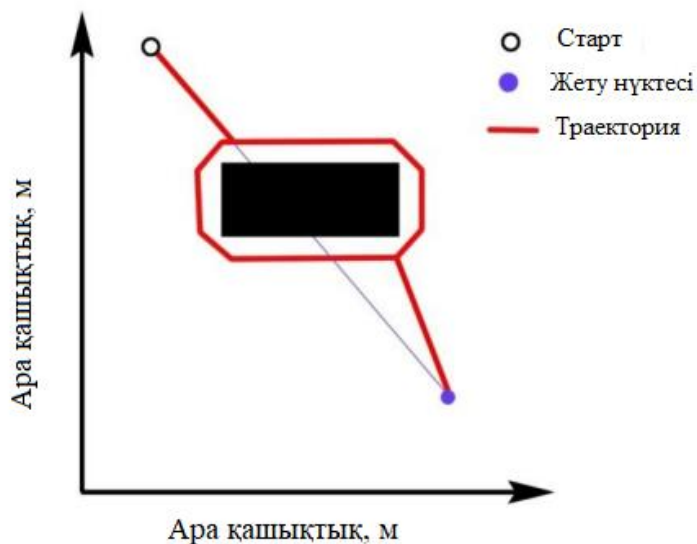
1.1 Кедергілерді өту алгоритмдері

Робототехниканың ең өзекті міндеттерінің бірі - автономды мобильді роботтарды жасау, оның негізгі міндеті - кедергілерге соқтығыспай, түпкілікті мақсатқа жету. Бұл жағдайда жүйенің тиімді жұмысының критерийлері жолдың ұзындығы және маршрутты аяқтауға кететін уақыт болып табылады.

Бүгінгі күнге дейін көптеген кедергілерді болдырмау алгоритмдері ұсынылды, әр түрлі сыртқы жағдайларда бір-бірінен жұмыс тиімділігімен ерекшеленеді. Осы жұмыста бұл мәселені шешудің ең танымал тәсілдерінің қатары қарастырылады. Жалпы олардың жұмыс істеу принциптері, негізгі артықшылықтары мен кемшіліктері қарастырылады [4].

1.1.1 Bug алгоритмдері

Bug алгоритмдері кедергілерді болдырмаудың ең қарапайым әдістері болып табылады. Бұл топтың бірінші өкілі - Bug1 алгоритмі, оның барысында робот кедергі сызығымен жүреді және мақсатты нүктеге дейін қалған қашықтықты үздіксіз бақылайды. Осылайша, робот кедергінің айналасында толық айналым жасап, нысанаға дейінгі қашықтық ең төменгі мәнге жеткен нүктеге оралады. Bug1 алгоритмінің негізгі кемшілігі кедергінің айналасындағы нүктелерді толық санау қажеттілігі екені анық [14].



Сурет 1.1 – Bug1 алгоритмін өңдеу суреті

Bug1 алгоритмі сипатталған тәсілдің дамуы Bug2 алгоритмі болып табылады. Оның түбегейлі айырмашылығы - кедергі анықталған кезде робот

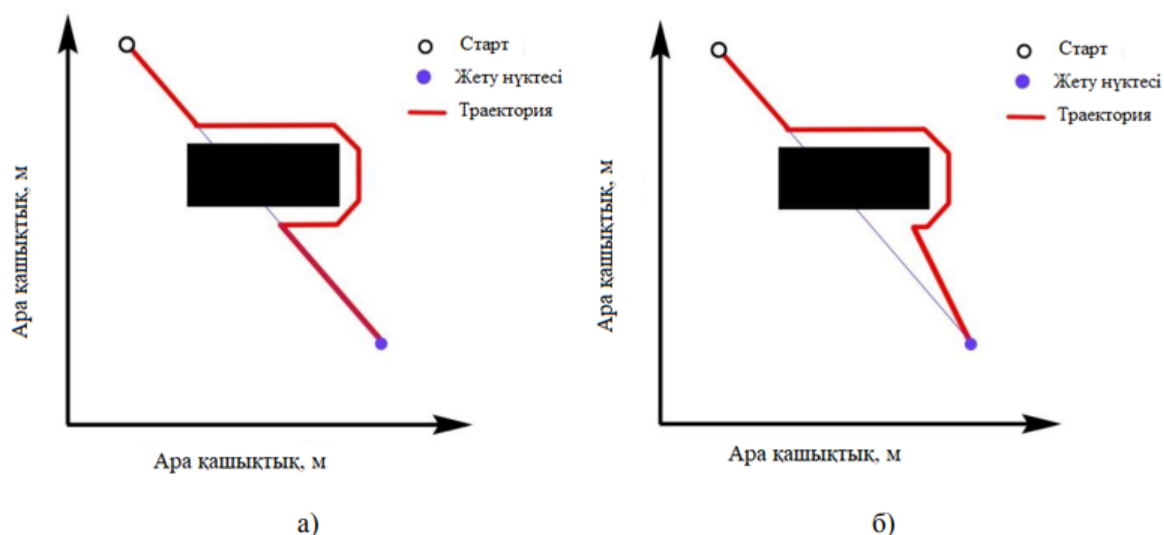
мақсатты нүктеге бағытталған векторды есте сақтайды. Bug1 алгоритміндегі сияқты, робот кедергінің бойымен қозғалады, бірақ векторды кесіп өткен кезде ол тікелей мақсатты нүктеге қарай жүріп, траекторияны өзгертеді.

1.1.2 Dist-Bug алгоритмі

Dist-Bug алгоритмі Bug алгоритмдерінің тағы бір дамуы болып табылады. Әдістің мәні роботтың орны мен нысана нүктесі арасындағы сәйкессіздікті үнемі өлшеу болып табылады. Сәйкессіздік мәнін азайту немесе сақтау кезінде робот кедергінің контуры бойынша қозғалуды жалғастырады. Әйтпесе, бағыт нысанаға өзгертіледі.

Bug2 және Dist-Bug алгоритмдерінің иллюстрациялары тиісінше а және б суреттерінде көрсетілген.

Суреттерде сипатталған жағдайларда Dist-Bug алгоритмі ең тиімді болып табылатыны анық.

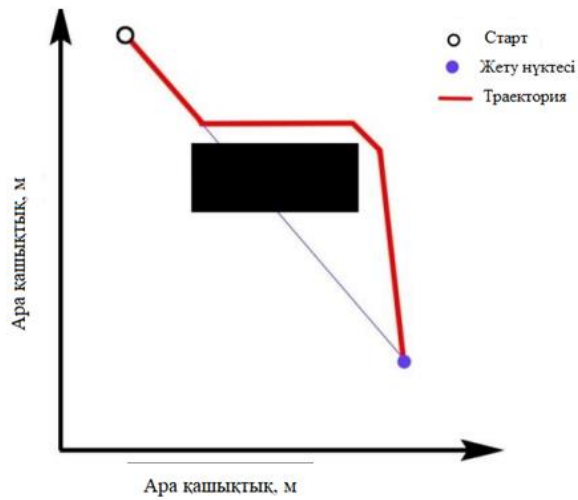


Сурет 1.2 – Bug2 (а) және Dist-Bug (б) алгоритмдерінің иллюстрациясы [13]

1.1.3 IVA алгоритмі

Bug тобының тағы бір алгоритмі - IVA алгоритмі. Ол алдыңғы нұсқалардың кейбір кемшіліктерін түзетуге арналған. Алгоритмді өңдеудің басында робот өтпейтін кедергінің контуры бойынша қозғалады (бұрын сипатталған Bug1, Bug2 және Dist-Bug алгоритмдеріндегідей). Бірақ DistBug-тен айырмашылығы, мақсатты қозғалыс тек соңғы нүктеге еркін жолдың болуымен

реттеледі. Мұндай модификация роботтың траекториясын жеңілдетеді және маршрутты аяқтауға кететін уақытты айтарлықтай қысқартады.

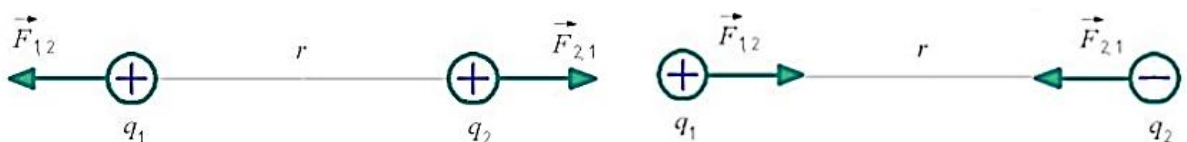


Сурет 1.3 – IVA алгоритмінің иллюстрациясы

Bug-алгоритмдерінің артықшылықтарына іске асыру және конфигурациялаудың қарапайымдылығы жатады. Дегенмен, соңғысын пайдалану тиімсіз болатын жағдайлар бар. Атап айтқанда, кейбір жағдайларда (ұзартылған кедергілер және т.б.) робот нысана траекториясының ұзындығын айтарлықтай арттыра алады [13].

1.1.4 Потенциалды өрістер әдісі (Artificial potential field)

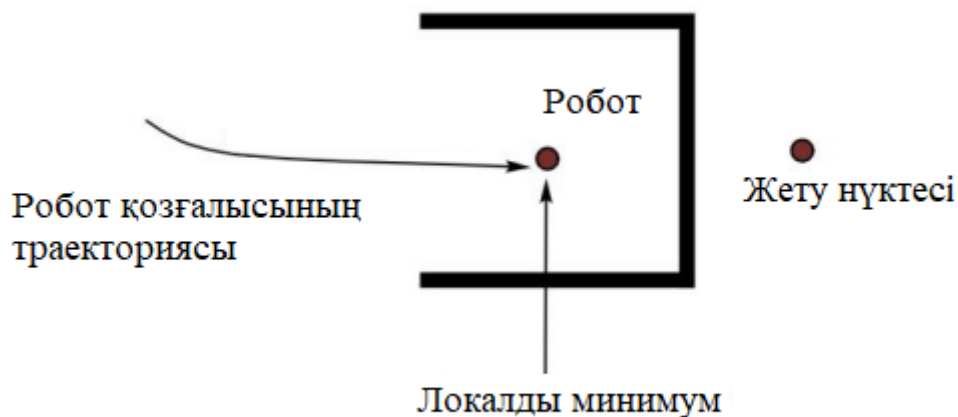
Әдіс кейбір зарядталған денелер ретінде объектілердің (робот пен кедергілер) абстрактілі көрінісін болжайды. Ұқсас зарядталған денелер бір-біріне жақындаған кезде қозғалатын объектінің (роботтың) жылдамдық векторының бағытын реттейтін өзара тебілудің молекулалық күштеріне ұқсас күштер пайда болады.



Сурет 1.4 – Потенциалды өрістер әдісін өңдеу кезінде жүйе объектілерінің өзара әрекеттесуінің иллюстрациясы

Нәтижесінде роботтардың есептелген жылдамдығы мақсатты қозғалыстың векторлық қосылуының және жүйе объектілерінің өзара әрекеттесу күштерінің нәтижесі болып табылады.

Бұл әдісті іске асыру және конфигурациялау оңай. Дегенмен, бұл әдіс үшін шешілмейтін жағдайлар бар, оларды локалды минималды жағдайлар ретінде сипаттауға болады. Олар U-тәрізді немесе симметриялы кедергілерді айналып өткенде, барлық объектілерден түсетін күштердің векторы нөлге тең болғанда пайда болуы мүмкін.

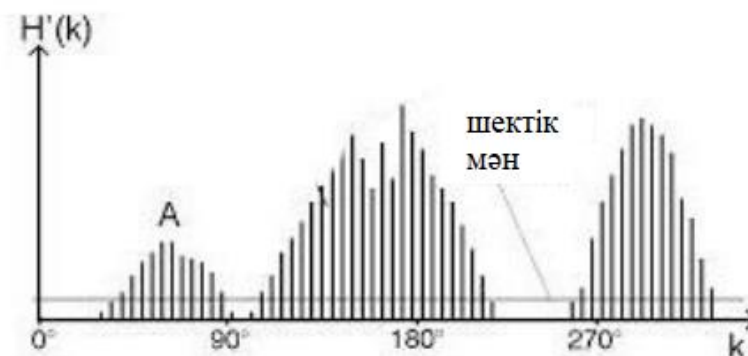


Сурет 1.5 – Шешілмейтін жағдайлардың мысалдары [9]

1.1.5 Векторлық өріс гистограммасы әдісі (Vector field histogram)

Осы әдісінің айрықша ерекшелігі бұрын анықталған жақын объектілер туралы ақпаратты сақтауға және есепке алуға мүмкіндік беретін аймақтың жергілікті картасын пайдалану болып табылады. Кедергілерді болдырмау үшін векторлық өріс гистограммасы әдісі полярлық гистограмманы жасайды.

X - осі кедергі мен робот қозғалысының бағыты арасындағы бұрышты, ал Y - осі кедергінің болуының P ықтималдығын көрсетеді.

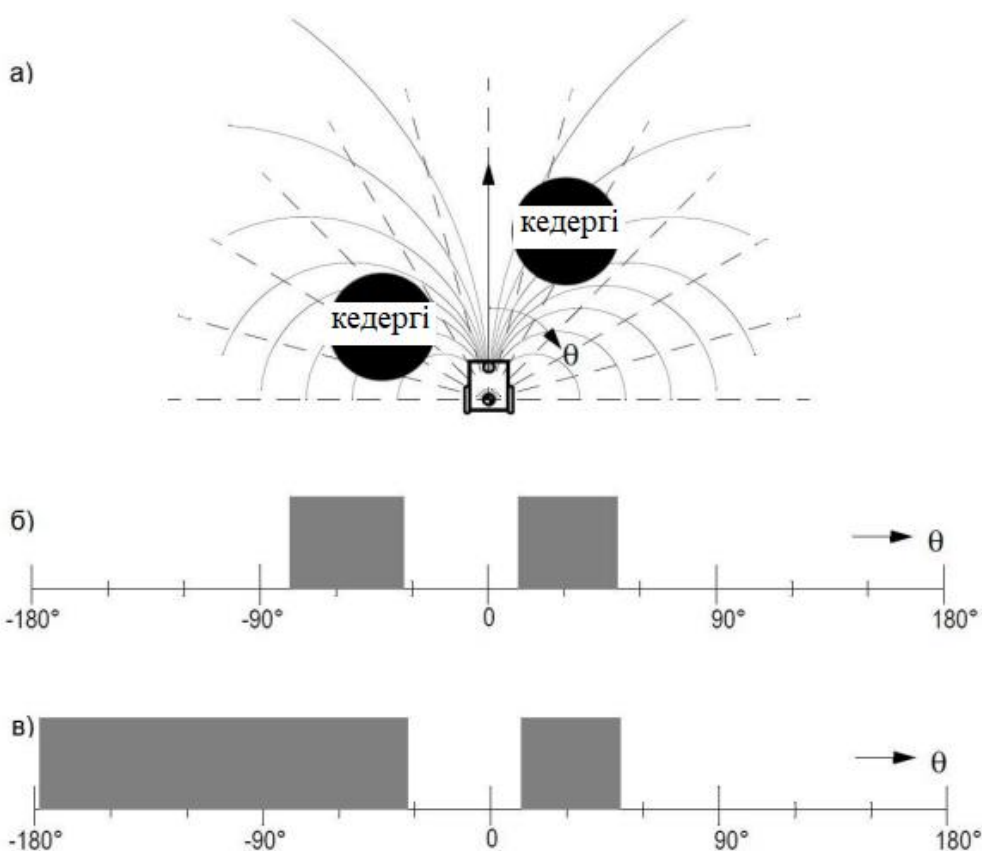


Сурет 1.6 – Өрістік гистограмма

Роботтың қозғалысы осы гистограмманы талдау негізінде бірегей түрде анықталады: бірінші қадамда кедергісіз бағыттар (саңылаулар) анықталады, содан кейін әрбір саңылауға салмақ функциясы қолданылады және салмағы ең азы таңдалады.

Әдістің артықшылықтары: жергілікті минимумы бар жағдайлардың болмауы (өтпейтін жағдайлар) және шектеулі дәлдіктегі датчиктермен (мысалы, ультрадыбыстық қашықтық өлшегіштермен) жұмыс істеу мүмкіндігі [3].

Алгоритмнің кейінгі нұсқаларында роботтың кинематикасын ескере отырып, ықтимал траектория бойынша қозғалыстың жеңілдетілген үлгісі ескеріледі. Ол үшін роботтың қозғалысы оның мобильді платформасы мүмкіндік беретін барлық ықтимал траекториялар бойынша үлгіленген. Бар кедергілер олар арқылы өтетін барлық траекторияларды бөгейді. Нәтижесінде полярлық гистограмманың маскасы (сурет 1.7), онда барлық жарамсыз траекториялар кедергілерді кеңейту арқылы кесіледі. Бұл мобильді роботтың кинематикалық шектеулерін ескере отырып, ашудың көлемін ғана емес, сонымен қатар одан өту мүмкіндігін ескере отырып, кедергілерді айналып өту жолын таңдауға мүмкіндік береді.



- б) полярлық гистограмма;
- в) полярлы гистограмма маскасы

Сурет 1.7 – Роботтың мүмкін траекторияларын есептеу

Айқын артықшылықтарға қарамастан (конвергенция, тиімділік) әдісті жүзеге асыру өте қиын, сонымен қатар борттық компьютер ресурстарының үлкен көлемін (есептеу қуаты, жады) қажет етеді. Сонымен қатар, U-тәрізді және симметриялы кедергілер болған кезде өтпейтін жағдайлар бар [7].

Кедергілерді болдырмау алгоритмін таңдағанда, қолданылатын мобильді роботтың техникалық жабдықталуы мен кинематикалық мүмкіндіктерін ескерген жөн. Bug алгоритмдері роботтың сезімтал жүйесіне және борттық компьютердің өнімділігіне аз талап етеді, бірақ басқа әдістерге қарағанда тиімділігі төмен. Потенциалды өріс әдісі күрделірек зондтау жүйелерін қажет етеді және симметриялы және U-тәрізді кедергілерден өтуде қиындық тудырады. Векторлық өріс гистограммасы әдісі ықтималдық тәсілінің қолданылуына байланысты дәл емес ақпаратпен жұмыс істеу үшін өте қолайлы, бірақ ол күрделі зондтау жүйелерін және борттық компьютердің жеткілікті өнімділігін талап етеді. Векторлық өріс гистограммасы әдісі әдісіне өзгертулер роботтың кинематикалық шектеулерін де ескеруі мүмкін.

Осы алгоритмдерді талдау арқылы бізге тиімдісі ИВА алгоритмі. Біздің жобаланатын роботымыз осы алгоритммен жүреді деп жобаланады.

1.2 Кедергілерді анықтайтын датчиктер

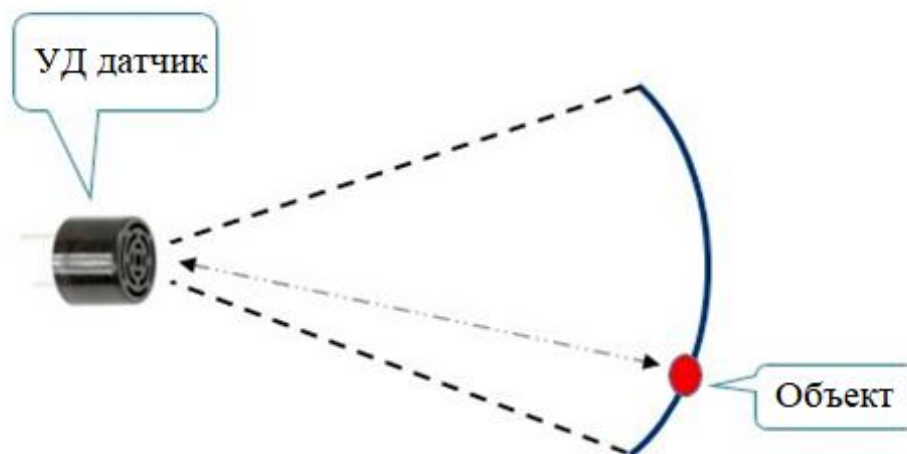
Роботтарды жобалауда жиі кездесетін мәселелердің бірі - кедергілерді болдырмаудың тиімді алгоритмін жасау мәселесі. Робот қозғалыс кезінде тұрып қалмауы және тапсырманы орындамайынша тоқтап тұруы керек.

Қазіргі уақытта лазерлік технология кедергілерді анықтаудың негізгі құралы болып табылады, себебі ол жоғары ажыратымдылық пен ұзақ қашықтықты қамтамасыз етеді. Осы сияқты кедергілерді анықтаудың құралы ультрадыбыстық датчик технологиясы, бірақ оның бұрыштық ажыратымдылығы нашар. Бұл жұмыста біз ультрадыбыстық датчиктің алынған деректерді пайдаланамыз, себебі ол тиімді нұсқа.

Бұл жүйе қоршаған орта температурасындағы ауадағы дыбыс жылдамдығын ескере отырып, көру өрісіндегі ең жақын кедергіге дейінгі қашықтықты +/-30 градусқа қайтаратын 40 кГцты таратқыш/қабылдағыш комбинациясын пайдаланады. Таратқыш/қабылдағыштың екі жағында орналасқан екі гидроакустикалық қабылдағыш көлденең айналмалы білікке орнатылады. Бұл біліктің айналуы таратқыштың/қабылдағыштың көру аймағын қамтитын тұрақты аралықта жеке гидроакустикалық бақылауларды жүргізуге мүмкіндік береді. Сынама алу жылдамдығын азайту үшін әрбір сонар арнасы үшін аналогты-сандық түрлендіру алдында конверт детекторы пайдаланылады. Әрбір арна үшін скалярлық Калман сүзгісі нақты уақыттағы сүзуді қамтамасыз ете отырып, сигнал шуының әсерін азайтады [5].

Бұл роботтарды белгісіз ортада шарлау контроллерге жылдамдық, позиция, кедергілерден қашықтық және бағдар туралы деректер беру үшін көптеген түрлі датчиктерді пайдалануды талап етеді. Оптикалық сенсорлар

жарыққа сезімтал, ал камераларды пайдалану жиі есептеуді қажет етеді, бұл дронның борттық процессорына көмектесу үшін жерүсті станцияларының пайдаланылуына әкеледі. Балама нұсқасы - он метрге дейінгі жақын қашықтықта анықтауға жарамды және секундына бірнеше диапазонды өлшеуді қамтамасыз ететін ультрадыбыстық датчиктерді пайдалану. Бұл датчиктердің артықшылығы - олар арзан, қуатты аз тұтыну және басқа датчиктер жүре алмайтын қоршаған орта жағдайында жұмыс істей алады, мысалы, түтін ортасы. Ажыратымдылық 60 градусқа дейін төмен болуы мүмкін.



Сурет 1.8 – Датчиктің көру аймағы

Бұл жұмыстың негізгі үлесі - бірқатар сонар қабылдағыштарынан алынған бастапқы деректерді және кедергілерді анықтау кезінде бұрыштық рұқсатты жақсарту үшін оны қалай өңдеуге болатынын зерттеу.

Жердегі көліктерге бағытталған автономды жүйелердің бастапқы дамуы және кедергілерді анықтау және болдырмау үшін ультрадыбыстық сенсорларды пайдалану болып табылады. 300 градустық көру өрісі бар жалғыз айналмалы ультрадыбыстық түрлендіргішті басқару үшін кадамдық қозғалтқышты пайдаланатын жердегі робот ұсынылды. Деректерді біріктіру алдын ала қысқарту және көшіру (баламалы қайтару жолдары) сияқты ішкі қателердің әсерін азайту үшін Байес комбинациясын пайдалану арқылы орындалады.

Әрбір бұрышты өлшеу робот қозғалмайтын кезде өңделеді, нәтижесінде өлшеулер арасында үлкен кідіріс пайда болады, бұл әдіс робот қолданбалары үшін жарамсыз етеді. 2013 жылы олар роботқа орнатылған ультрадыбыстық сенсор арқылы қоршаған ортаны картаға түсіруді жүзеге асырды. Толтыру торын пайдалана отырып, әрбір тор ұяшығына сәйкес аумақты толтыру ықтималдығы тағайындалады. Торды жаңарту қай ұяшықтар жаңартылатынын және алдыңғы көрсеткіштерді ескере отырып, әрбір ұяшықты толтыру ықтималдығын анықтайтын сәуленің еніне және сенсор ауқымына байланысты. Берілген ұяшық үшін сенімділік сенсор қайтарған қашықтықта сәуленің енін құрайтын

ұяшықтардың санына байланысты, яғни ұяшыққа тағайындалған мән сәуленің еніндегі әлеуетті тор ұяшықтарының кері мәні болып табылады [1].

Төменгі суретте көрсетілгендей, робот қоршаған ортада қозғалған кезде бірнеше доғаның қиылысуын қарастырады.



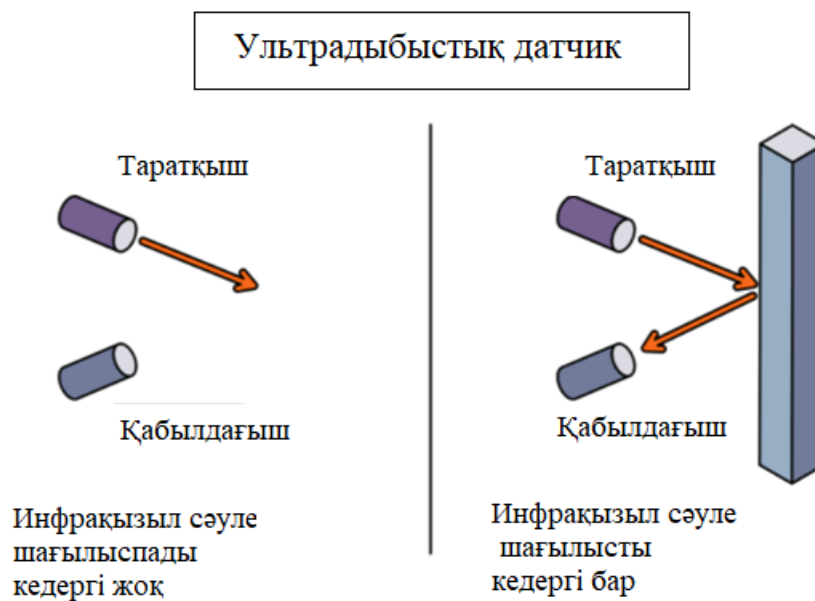
Сурет 1.9 – Роботтардың әсер ету радиусы доғаларының қиылысулары

Сенсорды имитациялайтын көптеген доғалар қиылысатын жерлерде кедергіден шағылысу нүктесінің болмау мүмкіндігі аз. Қоршаған ортадағы және сенсордағы жүйелік шудың арқасында бұл қиылыстар нүктелер тобын құрайды, содан кейін олар доға бойымен объектінің шамамен орналасуын қамтамасыз ете алады. Робот үшін Arc-Transversal Median алгоритмін қолданудың әлсіз жері дәл одометрияны және кедергіні өлшеу кезінде көлікті тоқтату мүмкіндігін қамтамасыз ету қажеттілігі болып табылады, өйткені жүріп өткен қашықтық доғаның көлденең бұрышпен өтетін орнын анықтайды (30°). Сенсорлық жүйенің өнімділігін арттырудың бір жолы - әртүрлі сенсорлардың деректерін біріктіру, бұл сенсорды біріктіру деп аталады. Caasenbrood (2006) Калман сүзгісінің вариациясын пайдалана отырып, UAV-лардың орнын бағалау үшін IMU өлшемдерімен бірге ультрадыбыстық сенсорларды пайдалануды зерттеді. Датчиктің көру аймағындағы кедергілерді анықтауға немесе бұрыштық ажыратымдылыққа ерекше көңіл бөлінбеді [8].

Кедергілерді анықтау және болдырмау мобильді роботтарды дамытудағы орталық мәселе ретінде қарастырылуы мүмкін. Бұл технология роботтарға өздеріне зиян келтірместен бейтаныс ортада қозғала алатын сезімдер береді.

Осы дипломдық жобада өз жолындағы кедергілерді анықтай алатын және кедергі туралы ақпарат беретін робот әзірленді. Бұл Arduino микроконтроллерінде жұмыс істейтін және кедергілерді анықтау үшін үш ультрадыбыстық қашықтық сенсорын қолданатын роботты көлік. Микроконтроллер платформасы ретінде Arduino тақтасы таңдалды, ал бағдарламалау үшін Arduino Software аналогы пайдаланылды.

Ультрадыбыстық қашықтық сенсорының интеграциясы айналадағы кедергілерді анықтаудың жоғары дәлдігін қамтамасыз етеді. Робот толық және жартылай автономды 2 режимді жұмыс жасай алатын болғандықтан, ол белгісіз орталарға ешқандай соқтығыспай кедергі қай бағытта және арақашықтығын нақты есептеп сәтті маневр жасады. Бұл жобада қолданылатын аппараттық құрал кеңінен қол жетімді және арзан, бұл роботты қайта өндіруге қабілетті етеді.



Сурет 1.10 – Ультрадыбыстық датчиктің кедергіні көруі [1]

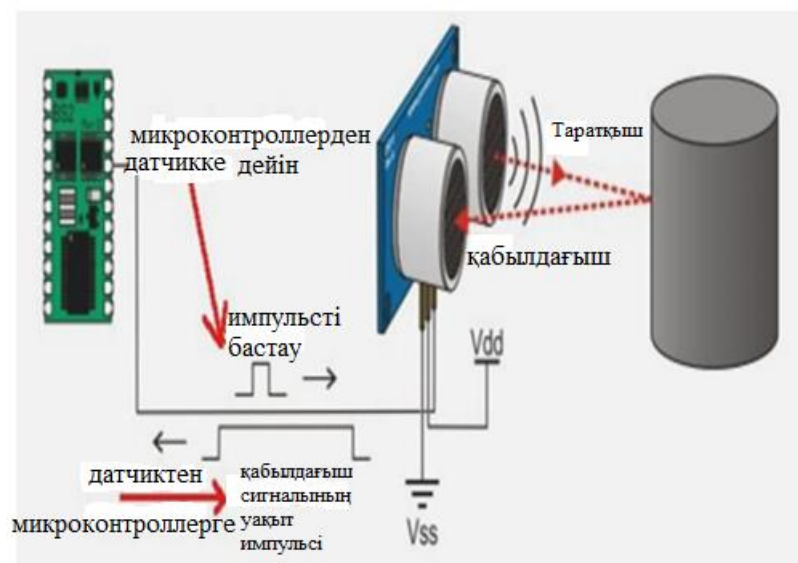
Ультрадыбыстық қашықтық датчик көмегімен жолындағы кедергілерді анықтау арқылы соқтығыспай қозғала алатын кедергілерден аулақ жүретін робот. Осы технологиямен басқарылатын роботтар геодезиялық, ұшқышсыз көліктер, автономды тазалау, автоматтандырылған шөп шабу машинасы және өнеркәсіпте басқару роботы сияқты көптеген қолданбаларда қолданылуы мүмкін. Осы жоба аясында жасалған робот келесі тапсырмаларды орындайды деп күтілуде:

- Робот алдын ала белгіленген шекті қашықтық негізінде өз жолындағы кедергілерді анықтай алуы керек.
- Кедергіні анықтағаннан кейін робот автономды шешімдер қабылдау арқылы кедергі қай бағытта қай арақашықта нақты есептеп мәлімет беріп отыру қажет.
- Жұмыс кезінде сыртқы бақылауды қажет етпейді.
- Ол белгісіз орталарда тиімді жұмыс істей алады.

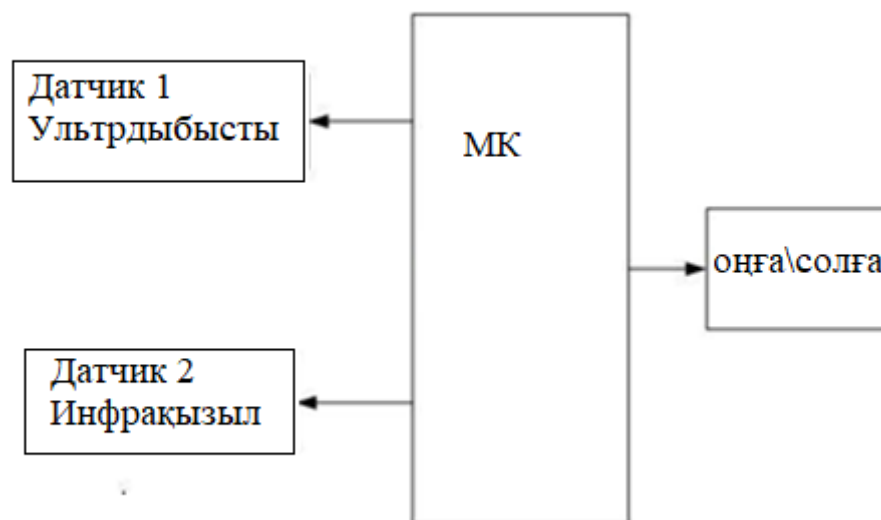
2 Құрастыру бөлімі

2.1 Кедергіні анықтау

Дүние жүзінде маңызды және өзекті тапсырмаларды орындау үшін автономды жүйелерді пайдалану қарқынды дамып келеді. Дегенмен, оның әртүрлі салаларда қолданылуын ерекше атап өтуге болмайды. Осы бөлімде кедергілерді анықтау және айналып өтетін роботтың жүйесін қарастырамыз. Жүйе екі (инфрақызыл және ультрадыбыстық) датчиктен, Arduino микроконтроллерінен және тұрақты ток мотор қозғалтқыштан тұрады. Ультрадыбыстық және инфрақызыл датчиктер интерфейссті микроконтроллерге сигналдар жіберу арқылы робот жолындағы кедергілерді анықтау үшін жүзеге асырылады. Микроконтроллер анықталған кедергіні болдырмау үшін қозғалтқышты іске қосу арқылы роботты баламалы бағытта қозғалуға бағыттайды. Кез келген кедергіге жарық пен дыбыс толқындарын жіберу үшін роботтың алдыңғы жағына орналастырылған инфрақызыл және ультрадыбыстық датчик сонар модульдері арқылы кедергілерді анықтау тізбегі жүзеге асырылды.



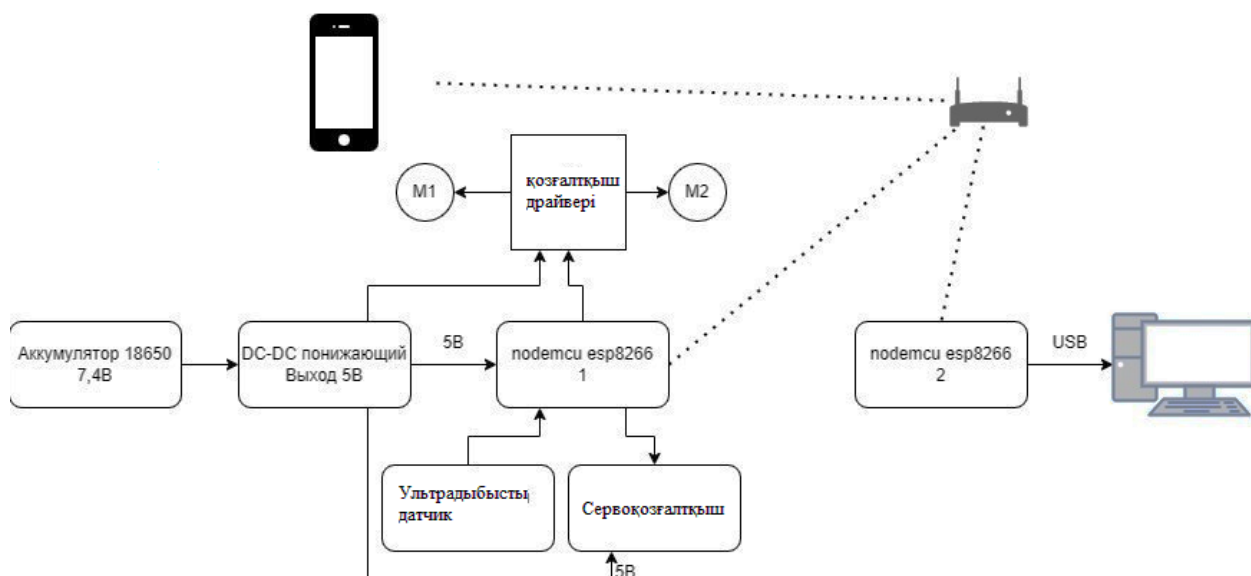
Сурет 2.1 – Кедергіні анықтау [13]



Сурет 2.2 – Жүйенің блок-схемасы [2]

2.2 Ақпаратты-өлшеу жүйесі

Ақпарат-өлшеу жүйесін жүзеге асыру үшін робот пен қоршаған ортамен байланысты ұйымдастыру керек, яғни физикалық диапазонда роботтың сезімталдылығын келтіру керек. Осындай кезде тактильді датчиктер, дальномерлер (лазерлі, ультрадыбысты, инфрақызыл) бұрылу бұрыш датчигі және тағы басқа сенсорлар. Төмендігі кестеде датчиктердің түрлері, олардың артықшылықтары мен кемшіліктері ұсынылған.



Сурет 2.3 – Құрастырылған роботтың қосылу сұлбасы

Жолды жоспарлау мобильді роботтардың навигациясы саласындағы ең маңызды міндет болып табылады.

Бұл міндет негізінен үш аспектіні қамтиды. Біріншіден, жоспарланған жол берілген бастапқы нүктеден берілген соңғы нүктеге дейін жүруі керек. Екіншіден, бұл жол роботтың мүмкін болатын кедергілерді болдырмай қозғалысын қамтамасыз етуі керек. Үшіншіден, жол алғашқы екі талапты қанағаттандыратын барлық мүмкін жолдардың ішінде белгілі бір мағынада оңтайлы болуы керек.

Жолды жоспарлау әдістерін әртүрлі критерийлер бойынша жіктеуге болады. Интеллектуалды технологияларды қолдану жағдайында оларды дәстүрлі әдістер мен эвристикалық әдістерге бөлуге болады. Қоршаған ортаның табиғаты бойынша жоспарлау әдістерін статикалық ортада және динамикалық ортада жоспарлау әдістеріне бөлуге болады (бірақ статикалық орта тәжірибеде сирек кездесетінін атап өту керек). Әдістерді қоршаған орта туралы ақпараттың толықтығына қарай да бөлуге болады: толық ақпараты бар әдістер (бұл жағдайда ғаламдық жолды жоспарлау) және толық емес ақпараты бар әдістер (әдетте роботқа жақын маңдағы жағдайды білу туралы дерек айту, бұл жағдайда біз жергілікті жолды жоспарлау).

2.3 Құрастырушы элементтер

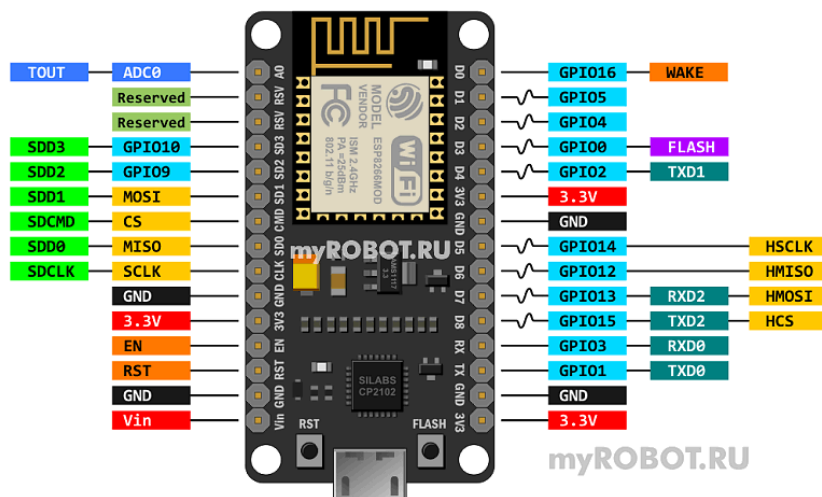
1) nodemcu esp8266

NodeMCU — ESP8266 модуліне негізделген платформа. Плата Wi-Fi арқылы жергілікті желіге немесе Интернетке сигнал беру арқылы қашықтықтағы әртүрлі тізбектерді ыңғайлы басқаруға арналған. Бұл платаны пайдалану

мүмкіндіктері көп. Мысалы, Node MCU негізінде елефон арқылы жарық немесе желдетуді басқаруды орнату, датчик көрсеткіштерін тіркеу және т.б. арқылы «ақылды үй» жасауға болады.

NodeMCU платасының өлшемі 6*3 см. Плата жеткілікті ықшам түрде, бұл оны көбірек жобаларда пайдалануға мүмкіндік береді.

Платаның алдыңғы жағында микро USB қосқышы бар, оның көмегімен эскиздер контроллерге құйылады немесе қуат көзінен немесе компьютерден беріледі.



Сурет 2.4 – nodemcu esp8266 платасы

Қосқыштың жанында екі түйме бар: «Жарқыл» және «Қалпына келтіру». «Жарқыл» түймесі жөндеу үшін, ал «Қалпына келтіру» түймесі тақтаны қалпына келтіру үшін қолданылады.

Платадағы кеңістіктің көп бөлігін ESP8266 чипі алады, оған 80 МГц тактілік жиілігі бар микропроцессор орнатылған (оны 160 МГц-ке дейін үдетуге болады). Тақтада 4 мегабайт Flash жады бар.

Платаны қуаттандыру үшін кернеуді 5-тен 12 В-қа дейін беруге болады, бірақ ол 10 В-тан ұсынылады. Оны Micro USB-ден де, Vin істікшесінен де (5 В-тан) қуаттауға болады. Сондай-ақ модульдерді ыңғайлы электрмен жабдықтау үшін қосымша кеңейту платалары бар.

- күрделі пішінді объектілерден сигналдың дұрыс көрсетілмеуі;
- түрлендіргіштегі күшті ауа ағыны сияқты жоғары дыбыс қысымдарына қатты сезімталдық.

Тестілеуден кейін қосымша өлшеу сигналын, атап айтқанда ИҚ сәулеленуін қамтитын қашықтық сенсорының жеке модулін әзірлеу туралы шешім қабылданды. ИҚ өлшеу сигналының жұмыс істеу принципі ультрадыбыстық сигналға ұқсас, тек дыбыстың орнына - жарық. Дегенмен, өлшеудің бұл түрі де кемшіліктерсіз емес. Бұл қара күнгірт нысандардың сигналды толық дерлік сіңіруі; шыныдан жасалған заттарды қате анықтау; тікелей күн сәулесіне күшті сезімталдық; азырақ, ультрадыбыстықпен салыстырғанда, максималды өлшенген қашықтық.

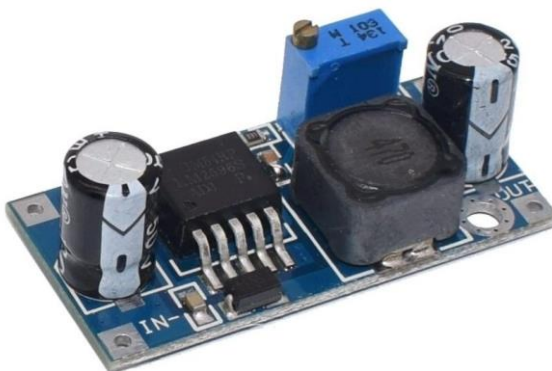
Өлшемдердің екі түрінің бірлескен жұмысы бір-бірінің кемшіліктерін сәтті өтейді. Нәтижесінде роботтың дұрыс жұмыс істеуі үшін барлық талаптарға жауап беретін қашықтық сенсорының модулі әзірленді.

Сондай-ақ әзірленген қашықтық сенсоры модулінің ерекшеліктерінің бірі объектіні анықтаудың жарық көрсеткіші болып табылады. Тақтада RGB жарық диоды орнатылған, оның жарқырау түсі бойынша сенсордың объектіге дейінгі қашықтықты бағдарламамен белгіленген мәннен азырақ алғанын түсінуге болады. Бұл нақты уақытта роботтың қалай және нені «көретінін» түсінуге мүмкіндік береді.

3) Кернеуді төмендететін түрлендіргіш 5V DC-DC GSMIN LM2596

Кернеу түрлендіргіші - бір кернеуді екіншісіне түрлендіретін пайдалы құрылғы. Құрылғы әмбебап болып табылады және оны, мысалы, автомобильде панельдегі кернеуді 12-24 В-тан 5 В-қа дейін төмендету үшін пайдалануға болады.

LM2596 микросхемасы ауыспалы төмендететін кернеу реттегішінен басқа ештеңе емес. Ол бекітілген кернеуде (3,3 В, 5 В, 12 В) және реттелетін кернеуде (ADJ) қол жетімді.



Сурет 2.7– Кернеуді төмендететін түрлендіргіш

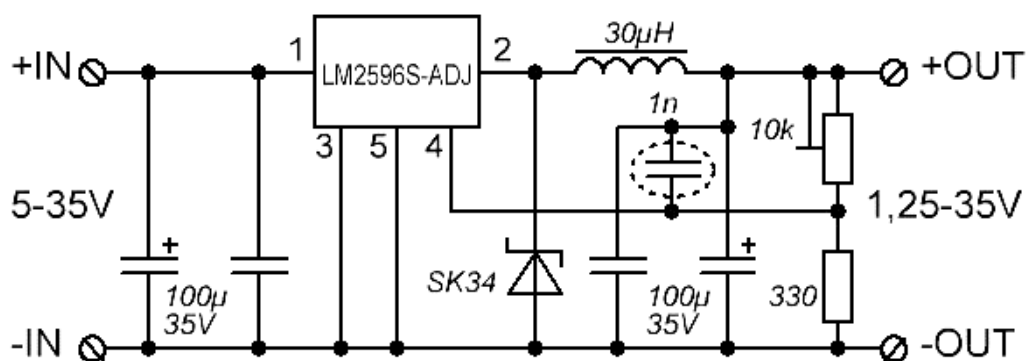
Кіріс кернеуі 5-35 В

3-40В шегінде шығыс кернеуін реттеу мүмкіндігі

Максималды жүктеме тогы - 2,5А

Түрлендіру жиілігі - 150 кГц

ПӘК - төмен кернеуде 70-75%, жоғары кернеуде 95% дейін.



Сурет 2.8– LM2596 микросхемасы ауыспалы төмендететін кернеу реттегіші [8]

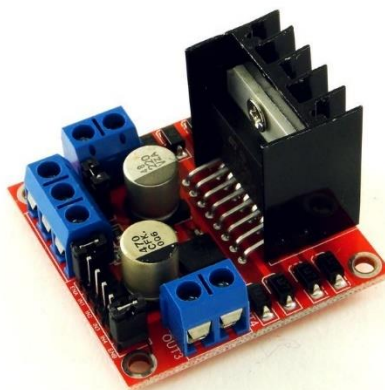
Шығу конденсаторы полимер болып табылады, ол шығыс кернеуінің толқынын азайтады.

1 кГц жиіліктегі кіріс конденсаторының өлшенген ESR 0,15 Ом, қандай да бір себептермен шығыс полимер конденсаторы 0,25 Ом. 150 кГц жиілікте ол өзін жақсырақ көрсетуі керек.

Schottky диоды SK34 3 А / 40V үшін, бірақ өлшемдері бойынша ол SK24-ге көбірек ұқсайды.

Баспа схемасы өте жинақы және жартылай өткізгіштер нашар салқындатылған, сондықтан ұзақ мерзімді максималды шығыс тогы 1,5А-дан аспауы керек, ал плата өте төмен тиімділікке байланысты өте қызады. Егер ток асып кетсе, микросхема қызып кетеді және оның кіріктірілген термиялық қорғанысы іске қосылады - шығыс кернеуі мезгіл-мезгіл секундқа жоғалады, содан кейін қалпына келтіріледі.

4) Тұрақты ток қозғалтқышының драйвері L298N, 1,5А, 2 арна

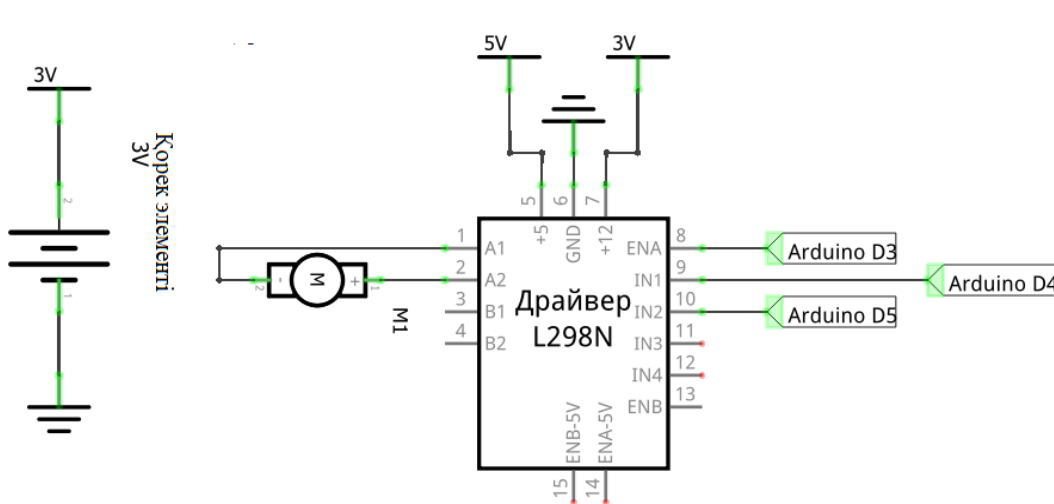


Сурет 2.9– L298N тұрақты ток қозғалтқышының драйвері

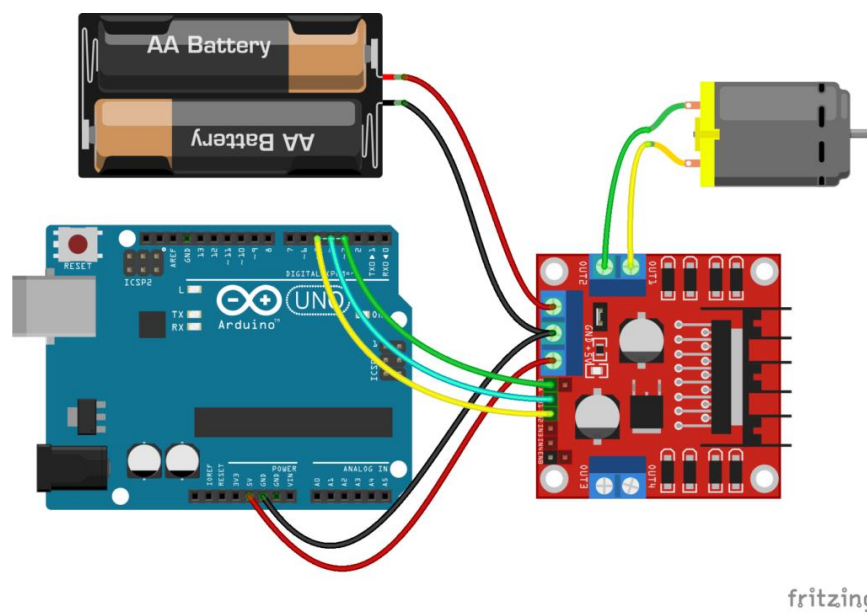
L298N модулінің сипаттамасы:

- қозғалтқыштың кернеуі: 35 В дейін;
- жұмыс тогы (арна бойынша): 2 А;
- мерзімді ток (80% - қосулы, 20% - өшіру): 2,5 А;
- қысқа мерзімді ток: 3 А;
- салмағы: 33 г.

L298N драйвері жоғары токпен жұмыс істейді. L298N көмегімен п20 сияқты әлсіз қозғалтқыштарды да, JGA25 немесе тіпті JGA37 сияқты қуатты қозғалтқыштарды 20-25 кг/см-ге дейінгі айналу моментімен басқаруға болады.



Сурет 2.10 – Принципиалды сұлбасы

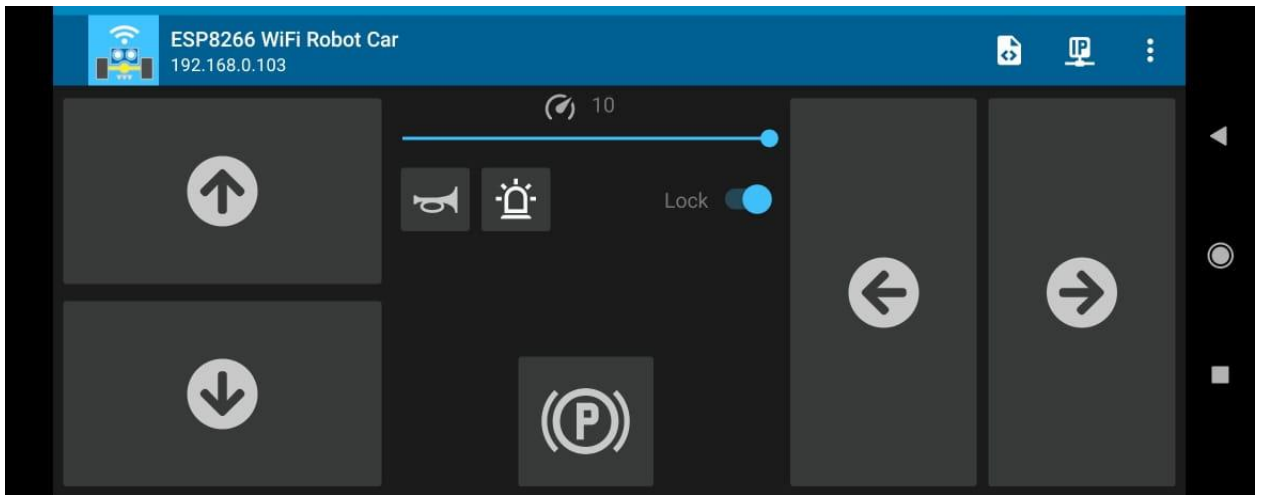


Сурет 2.11– Қосылу сұлбасы

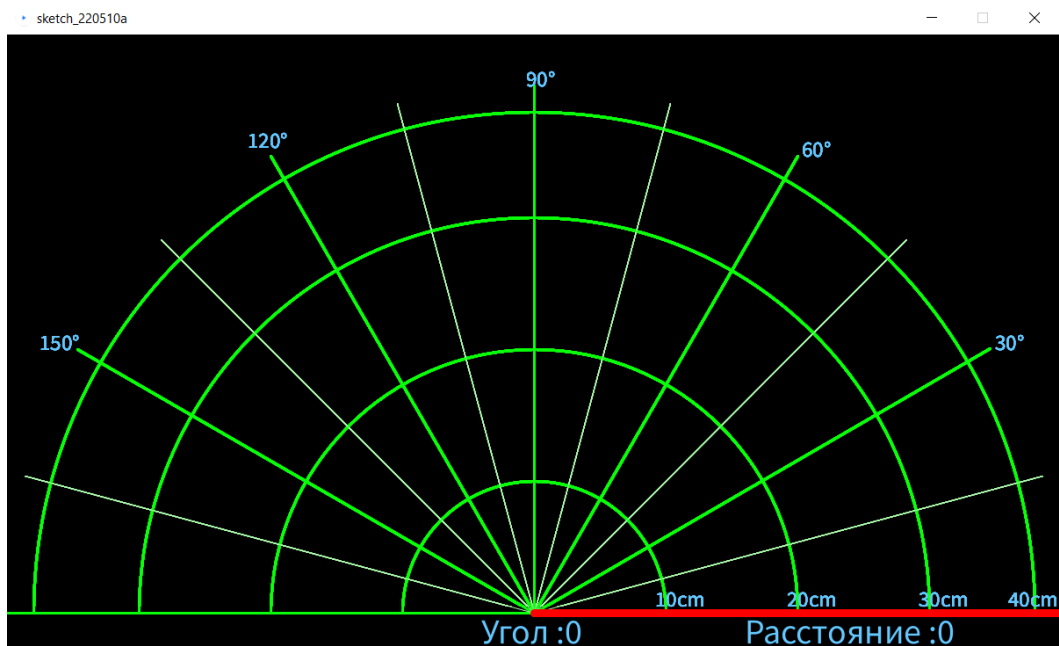
5) Аккумулятор 18650 7.4v 2600mAh



Сурет 2.12– Аккумулятор 18650 7.4v 2600mAh



Сурет 2.13 – Смартфонда роботты басқарушы бағдарлама (бұл ашық бағдарлама)



Сурет 2.14 – Компьютертерде экранға кедергіні анықтап ақпарат беру барысы. (бағдарламалық коды “Қосымша В” көрсетілген)



Сурет 2.15 – Құрастырылған мобильді робот (бағдарламалық коды “Қосымша А” көрсетілген)

ҚОРЫТЫНДЫ

Кедергілерді анықтау – бұл автономды роботтың басты талабы. Роботқа орнатылған датчиктер арқылы робот қоршаған ортадан ақпаратты алады. Кедергілерді анықтау үшін қолданылатын кейбір датчикте, мысалы, соққы датчиктері, инфрақызыл датчиктер, ультрадыбыстық датчиктер және т.б. Ультрадыбыстық датчиктер кедергілерді анықтау үшін ең қолайлы, ол арзан және ұзақ диапазонға ие [10].

Робот қажетті жол бойымен қозғалған кезде, ультрадыбыстық датчик сенсор басынан ультрадыбыстық толқындарды үздіксіз жібереді. Алда кедергіге тап болған кезде, ультрадыбыстық толқындар объектіден шағылысып, бұл ақпарат микроконтроллерге беріледі. Микроконтроллер ультрадыбыстық сигналдар негізінде қозғалтқыштарды сол, оң, артқы, алдыңғы жағын басқарады [2].

Сонарлардың кең таралуы олардың төмен құны, салмағы мен қуат тұтынуы, сигналдарды өңдеудің қарапайымдылығымен түсіндіріледі.

Сонымен қатар, электромагниттік сәулеленумен салыстырғанда дыбыстың төмен жылдамдығы (қалыпты температурада және тығыздықта ауада — 343 м/сек) сонарлардың жылдамдығын төмендетеді. Олардың тағы бір кемшілігі-тіркелген шағылысқан сигналдың шағылысатын ортаның қасиеттеріне қатты тәуелділігі. Мысалы, көлбеу бетінен бұл сигнал сонарға мүлдем енбейді.

Ауа ортасында 30-100 кГц жиіліктер қолданылады. Осы тербелістердің эмитенттері электростатикалық, пьезоэлектрлік, сондай-ақ ультрадыбыспен электр сигналының магнитострикциялық түрлендіргіштерін пайдаланады. Қабылдағыштар бірдей жұмыс принципінің қайтымды электроакустикалық түрлендіргіштер принципіне сәйкес ұйымдастырылған. Қабылдағыштар мен таратқыштарда ең көп таралған тікелей және кері пьезо эффектілері болды.

Қорыта келгенде, негізгі бөлімдерде роботтың алдында кездесетін кедергілерді анықтау үшін арналған сонар талданды, оның құрамы, сипаттамалары берілді, жұмыс принципі анықталынды. Алгоритмдерге талдау жасалынды. Практикалық бөлімде дипломның мақсаты бойынша кедергіні анықтайтын робот жасалынды, жүру алгоритмі ұсынылды. Оның басқарушы бағдарламасы, құрастырушы элементтері және қосылу сұлбалары келтірілді. Қосымшада оларға арналған бағдарлама кодтары ұсынылды.

ПАЙДАЛАНҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Борисов Ю. Инфракрасные излучения .М.: Энергия,1976.
- 2 Боригов В.Н Микроконтроллеры в измерительных устройствах
- 3 Бобровский, С.Н. Навигация мобильных роботов [Текст] / С.Н. Гончаров// Журн. PC Week. - 2004. - №9. - С. 60-63
- 4 Управление роботами. Состояние и перспективы [Текст] : материалы XX общ. собрания академии навигации и управления движением, 26 октября 2005 г. С.-Петербург / редкол : П.К.Плотников (отв. ред.). - С.-Петербург: Электроприбор, 2008. - 20 с.
- 5 Палагин В.А. Техническое задание на перспективную разработку мобильного робота для использования в чрезвычайных ситуациях [Текст] / Разработка СКБ «Робототехника и мехаторника» ХНУРЕ - Харьков, 2008. - 18 с.
- 6 Инструментарии роботостроения [Электронный ресурс] / Колорадо, М. Тим Джонс. - Режим доступа: [www/ URL:http://developerworks.ru/library/Irobotools/#author.html/](http://www.developerworks.ru/library/Irobotools/#author.html/) - 05.09.2008 г. - Загл. с экрана.
- 7 Баранов, Д.Н. Разработка интеллектуальной системы управления мобильными роботами на основе следящей системы технического зрения и нечеткой логики [Текст] : автореф. дис. кандидата техн. наук: 12.06.08 / Д.Н. Баранов; [Ун-т «СТАНКИН»]. - М., 2008. - 222с.
- 8 Навигация зрячего робота [Электронный ресурс] Режим доступа: [www/ URL: http://cofelis.ru/?page_id=46&page=3.html/](http://www.cofelis.ru/?page_id=46&page=3.html/) - 17.09.2008 г. - Загл. с экрана.
- 9 Навигация мобильных роботов [Электронный ресурс] / Лондон, Имперский Колледж. - Режим доступа: [www/ URL: http://computer-vision.usoz.ru/MobRoboNavigati/mobile_robot_navigation.html/](http://www.computer-vision.usoz.ru/MobRoboNavigati/mobile_robot_navigation.html/) - 10.06.1997 г. – Загл. с экрана.
- 10 Мартыненко, Ю. Г. Управление движением мобильных колёсных роботов [Текст] / Ю.Г. Мартыненко - МГУ им. М.В. Ломоносова, 2005. - 29-80с.
- 11 Однородные управляющие структуры адаптивных роботов [Текст] : А.В. Каляев, Ю.В. Чернухин и др.; гл. ред. физ.-мат. лит. - М.: Наука, 1990. - 152 с.
- 12 Управление роботами от ЭВМ [Текст] : учеб. / Е. И. Юревич, С.И. Новаченко, В.А. Павлов и др.; под. ред. Е. И. Юревича - Л. : Энергия, 1980. - 264с.
- 13 Интеллектуальный мобильный робот [Электронный ресурс] / - Евстигнеев Д.В. - Режим доступа: [www/ URL: http://robot-rad.narod.ru/index.html/](http://www.robot-rad.narod.ru/index.html/) - 15.02.2008г. - Загл. с экрана.
- 14 Синь Ю, Лян Хуахуа, Мэй Тао и др. Метод обнаружения и отображения динамических препятствий в беспилотных автомобилях на основе лазерных датчиков [J]. Робототехника, 2014, 36 (6): 654-661.
- 15 Су Чжиюань, Сюй Ючунь, Ли Юнлэ. Метод обнаружения транспортных средств на основе трехмерного лазерного радара [J]. Журнал Академии военного транспорта, 2017, 19 (1): 45-49.

16 Ван Сяо, Ли Кэцян, Ван Цзяньцян и др. Интеллектуальная идентификация параметров цели транспортного средства на основе трехмерного лазерного радара. Автомобильная инженерия, 2016, 38 (9): 1146-1152.

17 Ван Шэн. Исследование методов обнаружения роботизированных препятствий в сложной полевой среде. Ханчжоу: Чжэцзянский университет, 2015.

Қосымша А

Nodemcu esp82661

```
#include <ESP8266WiFi.h>
```

```
#include <ESP8266HTTPClient.h>
```

```
#include <WiFiClient.h>
```

```
#include <ArduinoOTA.h>
```

```
#include <ESP8266WiFiMulti.h>
```

```
ESP8266WiFiMulti WiFiMulti;
```

```
const char* ssid = "wifi name"; // тұрақты айнымалыға көрсеткіш.
```

```
const char* password = "wifi password";
```

```
//Your IP address or domain name with URL path
```

```
String IP;
```

```
const char* serverNameTemp = "http://192.168.1.20/?getsonar";
```

```
///  
//*****
```

```
#include <Wire.h>
```

```
String sonar;
```

```
String humidity;
```

```
String pressure;
```

```
int distance;
```

```
unsigned long previousMillis = 0;
```

```
const long interval = 1000;
```

```
void setup() {
```

```

Serial.begin(115200);
WiFi.mode(WIFI_STA);
WiFi.begin(ssid,password);
Serial.println("");
Serial.print("Connecting to: ");
Serial.println(ssid);
Serial.print("Password: ");
Serial.println(password);

unsigned long currentMillis = millis();
previousMillis = currentMillis;
while (WiFi.status() != WL_CONNECTED && currentMillis - previousMillis <=
10000) {
    delay(500);
    Serial.print(".");
    currentMillis = millis();
}

// if failed to connect with Wifi network set NodeMCU as AP mode
if (WiFi.status() == WL_CONNECTED) {
    Serial.println("");
    Serial.println("*WiFi-STA-Mode*");
    Serial.print("IP: ");
    Serial.println(WiFi.localIP());
}

ArduinoOTA.begin();
}

```

```

void loop() {
  ArduinoOTA.handle();

  if ((WiFiMulti.run() == WL_CONNECTED)) {
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
      previousMillis = currentMillis;

      sonar = httpGETRequest("http://192.168.1.20/?flagstart=ok");
      //*****

      //Serial.println(sonar);
      if(sonar=="ok"){
        sonar="";
        for(int i=0;i<180;i++){
          String temp="" +String(i);
          sonar = httpGETRequest(serverNameTemp+temp);
          //Serial.println("S: " + sonar);
          //delay(30);
          distance = sonar.toInt();
          Serial.print(i);
          Serial.print(",");
          Serial.print(distance);
          Serial.print(".");
        }
        /*for(int i=180;i>0;i--){
          String temp="" +String(i);
          sonar = httpGETRequest(serverNameTemp+temp);

```

```

        //Serial.println("S: " + sonar);
        //delay(30);
        distance = sonar.toInt();
        Serial.print(i);
        Serial.print(",");
        Serial.print(distance);
        Serial.print(".");
    } /*
}

}
}
}

```

```

String httpGETRequest(String serverName) {
    WiFiClient client;
    HTTPClient http;

    // Your IP address with path or Domain name with URL path
    http.begin(client, serverName);

    // Send HTTP POST request
    int httpResponseCode = http.GET();

    String payload = "--";

    if (httpResponseCode>0) {

```

```

//Serial.print("HTTP Response code: ");
//Serial.println(httpResponseCode);
payload = http.getString();
}
else {
  Serial.print("Error code: ");
  //Serial.println(httpResponseCode);
}
// Free resources
http.end();

return payload;
}

```

Nodemcu esp82661

```

#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <ArduinoOTA.h>
#include "SHCSR04.h"

```

```

SHCSR04 hcsr04;
#include <Servo.h>
const int trigPin=13;
const int echoPin=12;
const int servopin=14;
long duration;
int distanse;
Servo s1;

```

```

// connections for drive Motors

int PWM_A = 5;
int PWM_B = 0;
int DIR_A = 4;
int DIR_B = 2;

const int ledPin = 15;    // set digital pin D8 as LED pin (use super bright LED)
const int wifiLedPin = 16; // set digital pin D0 as indication, the LED turn on if
NodeMCU connected to WiFi as STA mode

String command;          // String to store app command state.
int SPEED = 255;        // 330 - 1023.
int speed_Coeff = 3;
int K=115;
bool f3=false;

ESP8266WebServer server(80);    // Create a webserver object that listens for HTTP
request on port 80

unsigned long previousMillis = 0;

String sta_ssid = "wifi name";    // set Wifi networks you want to connect to
String sta_password = "wifi password"; // set password for Wifi networks

void setup(){
  Serial.begin(115200);    // set up Serial library at 115200 bps
  Serial.println();
  Serial.println("*WiFi Robot Remote Control Mode*");
  Serial.println("-----");

```

```

s1.attach(servopin,500,2500);
s1.write(0);
pinMode(ledPin, OUTPUT);    // sets the LED pin as an Output
pinMode(wifiLedPin, OUTPUT); // sets the Wifi LED pin as an Output
digitalWrite(ledPin, LOW);
//digitalWrite(wifiLedPin, HIGH);

// Set all the motor control pins to outputs
pinMode(PWM_A, OUTPUT);
pinMode(PWM_B, OUTPUT);
pinMode(DIR_A, OUTPUT);
pinMode(DIR_B, OUTPUT);

// Turn off motors - Initial state
digitalWrite(DIR_A, LOW);
digitalWrite(DIR_B, LOW);
analogWrite(PWM_A, 0);
analogWrite(PWM_B, 0);

// set NodeMCU Wifi hostname based on chip mac address
String chip_id = String(ESP.getChipId(), HEX);
int i = chip_id.length()-4;
chip_id = chip_id.substring(i);
chip_id = "wificar-" + chip_id;
String hostname(chip_id);

Serial.println();
Serial.println("Hostname: "+hostname);

```



```

// first, set NodeMCU as STA mode to connect with a Wifi network
WiFi.mode(WIFI_STA);
WiFi.begin(sta_ssid.c_str(), sta_password.c_str());
Serial.println("");
Serial.print("Connecting to: ");
Serial.println(sta_ssid);
Serial.print("Password: ");
Serial.println(sta_password);

// try to connect with Wifi network about 10 seconds
unsigned long currentMillis = millis();
previousMillis = currentMillis;
while (WiFi.status() != WL_CONNECTED && currentMillis - previousMillis <=
10000) {
    delay(500);
    Serial.print(".");
    currentMillis = millis();
}

// if failed to connect with Wifi network set NodeMCU as AP mode
if (WiFi.status() == WL_CONNECTED) {
    Serial.println("");
    Serial.println("*WiFi-STA-Mode*");
    Serial.print("IP: ");
    Serial.println(WiFi.localIP());
    digitalWrite(wifiLedPin, LOW); // Wifi LED on when connected to Wifi as STA
mode
    delay(3000);
}

```

```

}

server.on ( "/", HTTP_handleRoot );    // call the 'handleRoot' function when a
client requests URI "/"

server.onNotFound ( HTTP_handleRoot ); // when a client requests an unknown
URI (i.e. something other than "/"), call function "handleNotFound"

server.begin();                        // actually start the server

ArduinoOTA.begin();                    // enable to receive update/uploade firmware via
Wifi OTA
}

void loop() {

    ArduinoOTA.handle();                // listen for update OTA request from clients
    server.handleClient();              // listen for HTTP requests from clients

    command = server.arg("State");      // check HTTP request, if has arguments
    "State" then saved the value

    if (command == "F") Forward();      // check string then call a function or set a
value

    else if (command == "B") Backward();

    else if (command == "R") TurnRight();

    else if (command == "L") TurnLeft();

    else if (command == "G") ForwardLeft();

    else if (command == "H") BackwardLeft();

    else if (command == "I") ForwardRight();

    else if (command == "J") BackwardRight();

    else if (command == "S") Stop();

}

```

```

// function prototypes for HTTP handlers
void HTTP_handleRoot(void){
    // Send HTTP status 200 (Ok) and send some text to the browser/client
    int I;
    if( server.hasArg("getsonar") ){

        String K=server.arg("getsonar");
        int I=K.toInt();
        Serial.println(K);
        s1.write(I);
        distanse = calDist();
        //Serial.println(distanse);
        String D=String(distanse);
        server.send ( 200, "text/html", D );
    }

    if( server.hasArg("State") ){
        Serial.println(server.arg("State"));
        server.send ( 200, "text/html", "" );

        if (command == "V"){
            //Serial.println("1");
            f3=true;
        }
    }

    if( server.hasArg("flagstart") ){
        Serial.println(server.arg("flagstart"));
    }
}

```

```

command = server.arg("flagstart");
if(command=="ok" && f3==true){
f3=false;
server.send ( 200, "text/html", "ok" );
Serial.println("send ok");
}else{
server.send ( 200, "text/html", "NULL" );
Serial.println("send NULL ");
}
}
}

```

```

void handleNotFound(){
server.send(404, "text/plain", "404: Not found"); // Send HTTP status 404 (Not
Found) when there's no handler for the URI in the request
}

```

// function to move forward

```

void Forward(){
digitalWrite(PWM_A, LOW);
analogWrite(DIR_A, (SPEED-K));

digitalWrite(DIR_B, LOW);
analogWrite(PWM_B, SPEED);
}

```

// function to move backward

```
void Backward(){
    digitalWrite(DIR_A, LOW);
    analogWrite(PWM_A, (SPEED-K));

    digitalWrite(PWM_B, LOW);
    analogWrite(DIR_B, SPEED);
}
```

// function to turn right

```
void TurnRight(){
    digitalWrite(DIR_A, LOW);
    analogWrite(PWM_A, (SPEED-K));

    digitalWrite(DIR_B, LOW);
    analogWrite(PWM_B, SPEED);
}
```

// function to turn left

```
void TurnLeft(){
    digitalWrite(PWM_A, LOW);
    analogWrite(DIR_A, (SPEED-K));

    digitalWrite(PWM_B, LOW);
    analogWrite(DIR_B, SPEED);
}
```

// function to move forward left

```

void ForwardLeft(){
    digitalWrite(PWM_A, LOW);
    analogWrite(DIR_A, (SPEED-K));

    digitalWrite(DIR_B, LOW);
    analogWrite(PWM_B, SPEED/speed_Coeff);
}

// function to move backward left
void BackwardLeft(){
    digitalWrite(DIR_A, LOW);
    analogWrite(PWM_A, (SPEED-K)/speed_Coeff);

    digitalWrite(PWM_B, LOW);
    analogWrite(DIR_B, SPEED);
}

// function to move forward right
void ForwardRight(){
    digitalWrite(PWM_A, LOW);
    analogWrite(DIR_A, (SPEED-K)/speed_Coeff);

    digitalWrite(DIR_B, LOW);
    analogWrite(PWM_B, SPEED);
}

// function to move backward left

```

```
void BackwardRight(){
    digitalWrite(DIR_A, LOW);
    analogWrite(PWM_A, (SPEED-K));

    digitalWrite(PWM_B, LOW);
    analogWrite(DIR_B, SPEED/speed_Coeff);
}
```

// function to stop motors

```
void Stop(){
    digitalWrite(DIR_A, LOW);
    digitalWrite(DIR_B, LOW);
    analogWrite(PWM_A, 0);
    analogWrite(PWM_B, 0);
}
```

```
int calDist(){
    distanse=hcsr04.read(13, 12);
    return distanse;
}
```

Қосымша В

```
import processing.serial.*;
Serial myPort;

String ang="";
String distance="";
String data="";

int angle, dist;

void setup() {
  size (1200, 700);
  myPort = new Serial(this,"COM3", 115200);
  myPort.bufferUntil('.');
  background(0);
}

void draw() {
  fill(0,5);
  noStroke();
  rect(0, 0, width, height*0.93);

  noStroke();
  fill(0,255);
  rect(0,height*0.93,width,height);

  drawRadar();
  drawLine();
  drawObject();
  drawText();
}

void serialEvent (Serial myPort) {
  data = myPort.readStringUntil('.');
  data = data.substring(0,data.length()-1);

  int index1 = data.indexOf(",");
  ang= data.substring(0, index1);
  distance= data.substring(index1+1, data.length());

  angle = int(ang);
  dist = int(distance);
}
```



```

void drawRadar(){
  pushMatrix();
  noFill();
  stroke(10,255,10);
  strokeWeight(3);

  translate(width/2,height-height*0.06);

  line(-width/2,0,width/2,0);

  arc(0,0,(width*0.5),(width*0.5),PI,TWO_PI);
  arc(0,0,(width*0.25),(width*0.25),PI,TWO_PI);
  arc(0,0,(width*0.75),(width*0.75),PI,TWO_PI);
  arc(0,0,(width*0.95),(width*0.95),PI,TWO_PI);

  line(0,0,(-width/2)*cos(radians(30)),(-width/2)*sin(radians(30)));
  line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(radians(60)));
  line(0,0,(-width/2)*cos(radians(90)),(-width/2)*sin(radians(90)));
  line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(radians(120)));
  line(0,0,(-width/2)*cos(radians(150)),(-width/2)*sin(radians(150)));

  stroke(175,255,175);
  strokeWeight(1);
  line(0,0,(-width/2)*cos(radians(15)),(-width/2)*sin(radians(15)));
  line(0,0,(-width/2)*cos(radians(45)),(-width/2)*sin(radians(45)));
  line(0,0,(-width/2)*cos(radians(75)),(-width/2)*sin(radians(75)));
  line(0,0,(-width/2)*cos(radians(105)),(-width/2)*sin(radians(105)));
  line(0,0,(-width/2)*cos(radians(135)),(-width/2)*sin(radians(135)));
  line(0,0,(-width/2)*cos(radians(165)),(-width/2)*sin(radians(165)));

  popMatrix();
}

void drawLine(){

  pushMatrix();

  strokeWeight(9);
  stroke(0,255,0);
  translate(width/2,height-height*0.06);

  line(0,0,(width/2)*cos(radians(angle)),(-width/2)*sin(radians(angle)));

```

```

    popMatrix();

}

void drawObject(){

    pushMatrix();

    strokeWeight(9);
    stroke(255,0,0);
    translate(width/2,height-height*0.06);

    float pixleDist = (dist/40.0)*(width/2.0);
    float pd=(width/2)-pixleDist;

    float x=-pixleDist*cos(radians(angle));
    float y=-pixleDist*sin(radians(angle));

    if(dist<=40){
        line(-x,y,-x+(pd*cos(radians(angle))),y-(pd*sin(radians(angle))));
    }
    popMatrix();
}

void drawText(){
    pushMatrix();

    fill(100,200,255);
    textSize(25);

    text("10cm",(width/2)+(width*0.115),height*0.93);
    text("20cm",(width/2)+(width*0.24),height*0.93);
    text("30cm",(width/2)+(width*0.365),height*0.93);
    text("40cm",(width/2)+(width*0.45),height*0.93);

    textSize(40);
    text("Угол :"+angle,width*0.45,height*0.99);

    if(dist<=40) {
        text("Расстояние :"+dist,width*0.7,height*0.99);
    }

    translate(width/2,height-height*0.06);
    textSize(25);

```

```
text(" 30°",(width/2)*cos(radians(30)),(-width/2)*sin(radians(30)));
text(" 60°",(width/2)*cos(radians(60)),(-width/2)*sin(radians(60)));
text(" 90°",(width/2)*cos(radians(91)),(-width/2)*sin(radians(90)));
text("120°",(width/2)*cos(radians(123)),(-width/2)*sin(radians(118)));
text("150°",(width/2)*cos(radians(160)),(-width/2)*sin(radians(150)));

popMatrix();

}
```