



Универзитет у Београду

Машински факултет

Мастер академске студије
Модул за производно машинство

ИНТЕЛИГЕНТНИ ТЕХНОЛОШКИ СИСТЕМИ

ПРОЈЕКАТ

Оцена пројектног задатка:

Предметни
наставници:

проф. др Зоран Миљковић и
проф. др Бојан Бабић

Предметни
сарадници:

др Божица Бојовић, Најдан Вуковић, дипл.
маш.инж., Марко Митић, дипл. маш.инж.,
Милица Петровић, дипл. маш.инж.

Група: 3

Потпис наставника:

РБ	Презиме и име:	Бр.инд.	Потпис:
1.	Петронијевић Јелена	1035/09	
2.	Симеуновић Никола	1194/09	
3.	Срећковић Катарина	1185/09	
4.	Станојевић Марко	1200/09	

Школска година: 2010/2011.

ПРОЈЕКТНИ ЗАДАТАК (1/2)

Ради успостављања унутрашњег транспорта материјала, сировина и готових делова у оквиру експерименталног модела технолошког окружења „XY” применом интелигентних мобилних робота, на мобилном роботу *LEGO Mindstorms NXT* урадити следеће:

1. Формирати конфигурацију мобилног робота;
2. Развити и имплементирати модел кретања у *Matlab* окружењу;
3. Развити и имплементирати опсервациони (сензорски) модел мобилног робота применом система вештачких неуронских мрежа;
4. Применити алгоритам Калмановог филтера у циљу одређивања положаја мобилног робота у окружењу;
5. Имплементирати A^* алгоритам претраге;
6. У експерименталном моделу технолошког окружења верификовати резултате.

За дату диспозицију технолошког окружења „XY”:

1. Развити симулациони модел у *Anylogic* окружењу;
2. На основу резултата симулације предложити нови диспозициони план технолошког окружења (задржати исте машине) и формирати „троугаону” матрицу за нови модел;
3. Упоредити два модела диспозиционог плана на основу резултата симулације и дати закључак.

Решењем пројектног задатка обухватити:

1. Основни циљ пројекта;
2. Теоријску поставку проблема и анализу;
3. Тестирати и верификовати перформансе експерименталних и симулационих решења;
4. Дискутовати резултате и дати закључак;

Напомене:

1. Пројекат ће бити позитивно оцењен ако и само ако приликом одбране пројектних задатака пројектно решење омогући несметано функционисање мобилног робота у окружењу;
2. Студенти су у обавези да на предавања и вежбе дођу припремљени јер ће у супротном коначан исход пројектних активности бити негативан;
3. Иницијатива студената у погледу предлога решења проблема, као и у погледу рада на додатним проблемима је више него пожељна, па ће стога сваки додатни рад бити узет у обзир приликом формирања завршне оцене;
4. Рокови израде појединачних пројектних целина дефинисани су планом и програмом предмета (Course Outline);
5. Сва питања, сугестије и евентуалне проблеме предочити у директном контакту са предметним наставницима, проф. др Зораном Миљковићем и проф. др Бојаном Бабићем, као и путем електронске поште на zmiljkovic@mas.bg.ac.rs, bbabic@mas.bg.ac.rs, а посебно у разговору са сарадницима у настави и на е-пошту: nvukovic@mas.bg.ac.rs, bbojovic@mas.bg.ac.rs, mmitic@mas.bg.ac.rs, mmpetrovic@mas.bg.ac.rs.

Задатак издао: _____

(Најдан Вуковић)

Резиме

Проблем унутрашњег транспорта материјала заснива се на чињеници да је то област чије осавременавање не прати развој целог предузећа. У великом броју погона поред модерних машина алатки могу се видети транспортна средства која већ деценијама нису мењана. Стога је овај пројектни рад окренут роботизовању унутрашњег транспорта материјала. Циљ је да се виљушкар, којим се обавља транспорт у постојећем производном окружењу, замени интелигентним мобилним роботом. Вештачке неуронске мреже, које представљају парадигму вештачке интелигенције, коришћене су за класификацију и апроксимацију примљене сензорске информације. Модел кретања на основу пређеног пута имплементиран је у управљачки програм који генерише путању коришћењем А* алгоритма за задату почетну и циљну тачку. Локализација је извршена применом Калмановог филтера. Софтвери коришћени за симулацију и генерисање управљачког кода су AnyLogic и MatLab. Резултати који су овом приликом остварени показују да се интелигентни мобилни роботи могу имплементирати у данашњој индустрији уз остварење задовољавајуће тачности позиционирања.

Кључне речи: *интелигентни технолошки системи, интелигентни мобилни робот, вештачка интелигенција, Калманов филтер, А* алгоритам*

¹ **Јелена Петронијевић 1035/09**, Универзитет у Београду – Машински факултет студент друге године Дипломских академских студија.

Е-пошта: cini_mi_se_da_sam_videla_macu@yahoo.com

² **Никола Симеуновић 1194/09**, Универзитет у Београду – Машински факултет студент друге године Дипломских академских студија.

Е-пошта: nikolajsimeunovic@gmail.com

³ **Катарина Срећковић 1185/09**, Универзитет у Београду – Машински факултет студент друге године Дипломских академских студија.

Е-пошта: katarinasr@gmail.com

⁴ **Марко Станојевић 1200/09**, Универзитет у Београду – Машински факултет студент друге године Дипломских академских студија.

Е-пошта: ajkulaopcs@gmail.com

Списак слика

Слика 2.1: Layout производно-монтажног предузећа "XY"	12
Слика 2.2: Лабораторијски модел окружења	12
Слика 5.1: Стара и нова генерација Lego Mindstorm пакета	19
Слика 5.2: Оптички сензор	20
Слика 5.3: Звучни сензор	20
Слика 5.4: Ултразвучни сензор	21
Слика 5.5: Тактилни сензор	21
Слика 5.6: Изглед мотора	22
Слика 5.7: Опис протокола и изглед порта	22
Слика 5.8: Носећа конструкција мобилног робота	23
Слика 5.9: Приказ положаја управљачке јединице и оптичког сензора	24
Слика 5.10: Приказ редукционог система и енкодера мотора	24
Слика 5.11: Шематски приказ инкрименталног енкодера	25
Слика 5.12: Рефлектовање зрака од површину	25
Слика 5.13: Приказ на који начин оптички сензор детектује основне боје	26
Слика 6.1: Дефинисање степени слободе мобилног робота у x, y равни	27
Слика 6.2: Мобилни робот са два независна погонска тачка	29
Слика 7.1: Неурон вештачке неуронске мреже	31
Слика 7.2: Основни облик архитектуре ВР неуронске мреже	33
Слика 7.3: Пример промене оријентације робота	34
Слика 7.4: График преносне функције „ <i>logsig</i> “	36
Слика 7.5: Графички приказ симулације једне од три посматране ВМ	38
Слика 7.6: Графички приказ симулације једне ВМ која није добро обучена	39
Слика 7.7: Графички приказ симулације ВМ која је имплементирана у управљачки код мобилног робота	39
Слика 7.8: Архитектура одабране вештачке неуронске мреже	40
Слика 7.9: Приказ контролних тачака у радном окружењу	41
Слика 7.10: Алгоритам обраде података за детектовање промене боје у радном окружењу	42
Слика 7.11: Графички приказ добијених вредности читавања са сензора	42
Слика 7.12: Архитектура ВМ имплементиране у управљачки код мобилног робота ради обраде информација са оптичког сензора	44
Слика 7.13: Графички приказ резултата симулације ВМ за необучене улазе	45
Слика 7.14: Графички приказ обучавања мреже. Ток тестирања лоше и добро обучене мреже	45
Слика 9.3: Пример додавања вредности четворосмерног кретања	56
Слика 9.1: Четворосмерно кретање	56
Слика 9.2: Осмосмерно кретање	56
Слика 9.4: Пример додавања вредности осмосмерног кретања	57
Слика 9.5: Принципа функционисања A^* алгоритма, први и други корак	57
Слика 9.6: Принцип функционисања A^* алгоритма, трећи и четврти корак	58
Слика 9.7: Пример додељивања оцена Менхетн норме	60
Слика 9.8: Пример додељивања оцена Еуклидске норме	61
Слика 9.9: Непосредно окружење претраживача	61
Слика 9.10: Приказ дискретизованог радног окружења	62
Слика 10.1: Упоредни приказ реалног система и модела	64
Слика 10.2: Активности у оквиру процеса симулације [1]	64
Слика 10.3: Диспозициони план линије 3	66
Слика 10.4: Квалитативна матрица међузависности активности	67
Слика 10.5: Симулациони модел рада система постојеће производње	79
Слика 10.6: Симулација и анимација рада система	70
Слика 10.7: Застој система	70
Слика 10.8: Модел предложеног решења	71

Слика 10.9: Застој у раду предложеног система	71
Слика 11. 1: Изглед прозора за унос координата тачака	75
Слика 11. 2: Почетни положај мобилног робота	75
Слика 11. 3: Почетни положај мобилног робота сликан из другог угла.....	76
Слика 11. 4: Избор путање.....	76
Слика 11. 5: Кретање мобилног робота	77
Слика 11. 6: Генерисање друге тачке и елипсе сигурности	78
Слика 11. 7: Генерисање циљне тачке и елипсе сигурности.....	78
Слика 11. 8: Осварено кретање робота	78

Списак табела

Табела 1: Машине у погону предузећа.....	10
Табела 2: Скуп обучавајућих парова ВНМ за ротацију вратила мотора	35
Табела 3: Вредности углова за које мрежа није обучена.....	36
Табела 4: Вредности и опис параметара обучавања.....	37
Табела 5: Параметри обучавања ВНМ	37
Табела 6: Упоредивање добијених резултата обучавањем ВНМ.....	38
Табела 7: Координате референтних тачака.....	41
Табела 8: Мењани параметри при обучавању мреже	43
Табела 9: Упоредивање добијених резултата обучавањем ВНМ.....	43
Табела 10: Предности тест скупа за које мрежа није обучена.....	44
Табела 11: Приказ времена обраде по технолошким поступцима	67
Табела 12: Експериментални резултати	79

Садржај

1.	Увод.....	9
2.	Пројектни задатак.....	10
2.1.	Технолошко окружење	10
3.	Концепцијско решење пројектног задатка.....	13
3.1.	Ток развоја пројектног решења.....	13
4.	Индустријски и мобилни роботи, сензори и интелигентни технолошки системи	14
4.1.	Индустријски роботи.....	14
4.2.	Мобилни роботи.....	16
4.3.	Сензори.....	17
4.4.	Интелигентни технолошки системи	18
5.	Опис расположивих компоненти.....	19
5.1.	Фазе формирања конфигурације робота	22
5.2.	Детаљан опис коришћених сензора	24
6.	Модел кретања	27
6.1.	Модел кретања на основу пређеног пута.....	28
7.	Вештачке неуронске мреже	31
8.	Калманов филтер	46
8.1.	Теоријске поставке и једначине Калмановог филтера	46
8.2.	Алгоритам Калмановог и линеаризованог Калмановог филтера.....	49
8.3.	Алгоритам линеаризованог Калмановог филтера.....	51
8.3.	Имплементација линеаризованог Калмановог филтера за потребе одређивања положаја мобилног робота током експлоатације.....	52
9.	A* алгоритам.....	56
10.	Симулациони модел у AnyLogic окружењу	63
10.1.	Развој симулационог модела пројектног задатка	65
10.2.	Формирање нових решења токова материјала.....	71
10.3.	Дискусија	72
11.	Експериментални резултати.....	74
12.	Анализа и дискусија експерименталних резултата	80
13.	Закључак.....	81
14.	Литература.....	82
15.	Прилог А.....	83

1. Увод

Пројектни задатак разматра имплементацију мобилног робота у постојећи транспортни систем предузећа "XY" које се бави производњом SIVACON електро-ормана. С обзиром на то да је проблем манипулације материјалом великих размера и да на њега одлази чак до 35% [4] постојећих средстава оправданост увођења оваквог система је вишеструка. Иако би се у почетном тренутку јавили трошкови, средства би се у кратком периоду вратила с обзиром на то да је овај систем ефикаснији и да доводи до смањења циклусног времена, а самим тим и до производње већег броја склопова.

Идејно решење засновано је на експерименту изведеном у окружењу које веродостојно репрезентује погон предузећа "XY". Мобилни робот који је коришћен је конфигуриран деловима из комплета LEGO Mindstorm NXT [7]. Управљање се остварује помоћу RWTH Toolbox-а [8] и програма имплементираних у MatLab софтверском пакету [9]. Функције кретања остварене су помоћу модела кретања уз коришћење вештачких неуронских мрежа [5] како за функције ротације робота тако и за кориговање положаја помоћу Калмановог филтера. Само генерисање путање врши се применом А* алгоритма [2].

Други део пројектног задатка односи се на симулирање рада линије 3 Фабрике Металних Производа [10]. Увидели смо карактеристике рада овог производног система и понудили могућа побољшања на основу синтетисаног модела.

У другом поглављу првог дела пројектног задатка дефинисан је проблем који се обрађује у оквиру пројекта, док је детаљно описано и технолошко окружење.

Треће поглавље односи се концепцијско решење пројектног задатка.

У четвртном поглављу упознали смо се са основним појмовима везаним за мобилне роботе, сензоре и интелигентне технолошке системе.

У петом поглављу осврнули смо се на основне карактеристике LEGO Mindstorm NXT робота, са детаљима везаним за примењену конфигурацију.

У шестом поглављу говорили смо о моделима кретања са посебним освртом на имплементирани модел кретања на основу пређеног пута.

Седмо поглавље детаљније нас је увело у вештачку интелигенцију и имплементиране вештачке неуронске мреже.

Осмо поглавље односи се на Калманов филтер и његову имплементацију.

У деветом поглављу налазе се детаљи везани за А* алгоритам, његове теоријске основе и начин програмирања и имплементације.

Десето поглавље говори о симулацији погона Фабрике Металних Производа и обухвата формирање модела, симулацију, анимацију и генерисање нових решења на основу закључака изведених из реалног система.

Једанаесто поглавље описује експерименталне резултате и опис самог експеримента везаних за кретање интелигентног мобилног робота у реалном погону предузећа "XY".

Анализа и дискусија експерименталног решења дати су у дванаестом поглављу.

У оквиру поглавља број дванаест дат је закључак целокупног пројекта.

2. Пројектни задатак

Транспортни систем представља саставни део сваког производног система. О његовом значају није неопходно посебно говорити, довољно је само рећи да он представља интегративну компоненту целокупног погона. То је систем на који се у нашој привреди на жалост недовољно обраћа пажња, а без кога савремено предузеће не може уистину бити савремено. Наиме, ма колико улагали у модерне машине не можемо ухватити корак са великим светским предузећима уколико су нам транспортни путеви нерационални. Немогуће је смањити циклусно време чак и најсавременијим машинама ако се транспорт обрадака између тих машина обавља у неадекватном временском року. Прва фаза пројекта није била тема нашег рада, али је за резултат имала оптимално распоређивање машина у погону у складу са технолошким поступцима за репрезентативне делове, а водећи рачуна о минималним транспортним токовима. Овај део пројекта синтетисан је у радно окружење које смо користили за имплементацију савременог транспортног система користећи интелигентни мобилни робот, што је и тема нашег задатка.

2.1. Технолошко окружење

Пројектни задатак реализован је за реално предузеће "XY" тако да су сви подаци који су коришћени резултат претходно обављене детаљне студије. Такође, то значи да се и решење самог пројекта може применити у реалним условима.

Предузеће се бави производњом SIVACON електро-ормана тако да се целокупни процес од производње делова до монтаже одвија према Siemens AG [11] технолошком поступку. Целокупни распоред у предузећу овом приликом дефинишемо према [4]. У предузећу разликујемо основне машине које су приказане у следећој табели:

Табела 1: Маchine у погону предузећа	
M1	Маказе за сечење
M2	КНУ машина за пробијање и просецање
M3	КНУ хидраулична "апликант" преса
M4	Машина за исецање профила
M5	Стубна бушилица
M6	Стубна бушилица
M7	Кружна тестера
M8	Оштрилица
M9	Линија за обраду делова од бакра

Најјаче везе које су првенствено везане за ток материјала и за тип машине остварене су између прве четири машине тако да је то довело до закључка да њихом међусобни распоред мора бити линијски. С друге стране издвојене су машине M5, M6 и M7 јер се налазе у мањој употреби док је линија за обраду бакра потпуно дислоцирана у односу на све друге обрадне системе. Разлог за то лежи, као што је горе поменуто, у издвојености самог његовог технолошког поступка, што је како знамо основни параметар за распоређивање машина. Такође, уочено је да се оштрилица алата највише користи за оштрење алата који су везани за машину M2 тако да је смештена уз њу.

Складишни простори такође су организовани пре свега према пројектованим технолошким процесима. Мађутим, морамо имати у виду да је постојао и још један веома

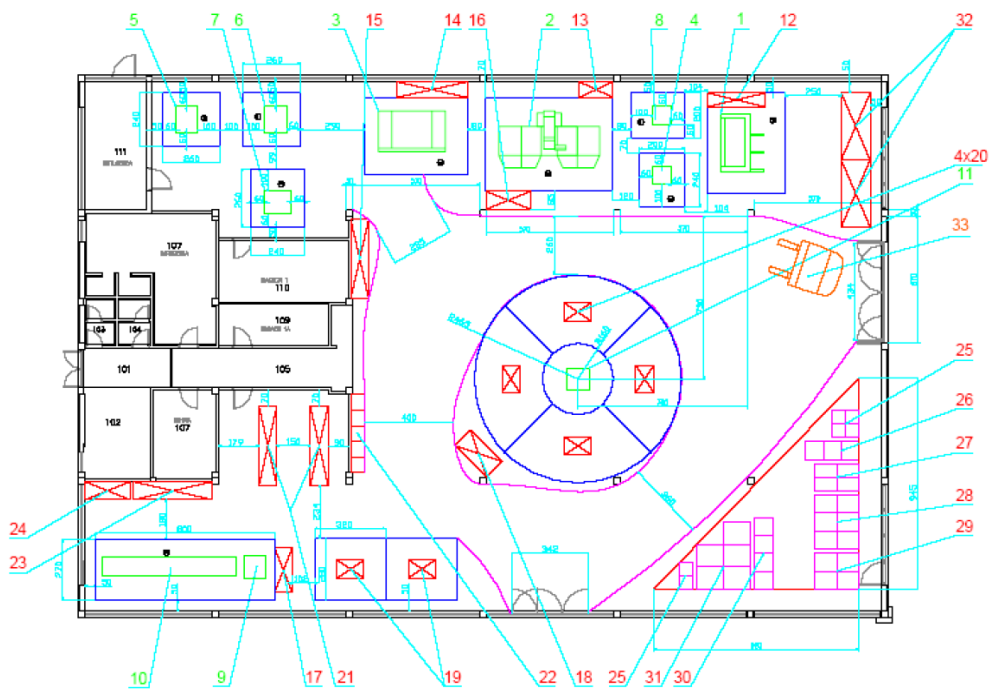
важан фактор. Постојећи простор морао се само модификовати јер је производна хала предузећа изграђена пре вршења саме анализе токова материјала. Резултати до којих се овом анализом дошло, а који су имплементирани у поставку нашег пројекта приказани су у наредних неколико редова. Предвиђена су улазна складишта полуфабриката (позиција 32), међускладиште (позиција 15), складиште готових делова од лима (позиција 21), складиште вијчане робе и компонената за монтажу (позиција 22), складиште готових делова од бакра (позиција 23) и складиште готових производа (позиције 25 до 31).

Складиште полуфабриката урађено је према постојећој комплетној конструкционој документацији и представљено је као регално складиште са полицама. Овом типу складишта припадају и међускладиште, складиште готових делова од лима и складиште готових делова од бакра. Складиште вијчане робе и компонената за монтажу је реализовано као сандучасто складиште којим је предузеће већ располагало. Складиште готових производа је палетног типа и смештено је у једини расположиви простор за ту намену. У оквиру тог простора врши се одлагање различитих врста SIVACON електро-ормана у зависности од претходно дефинисаног плана.

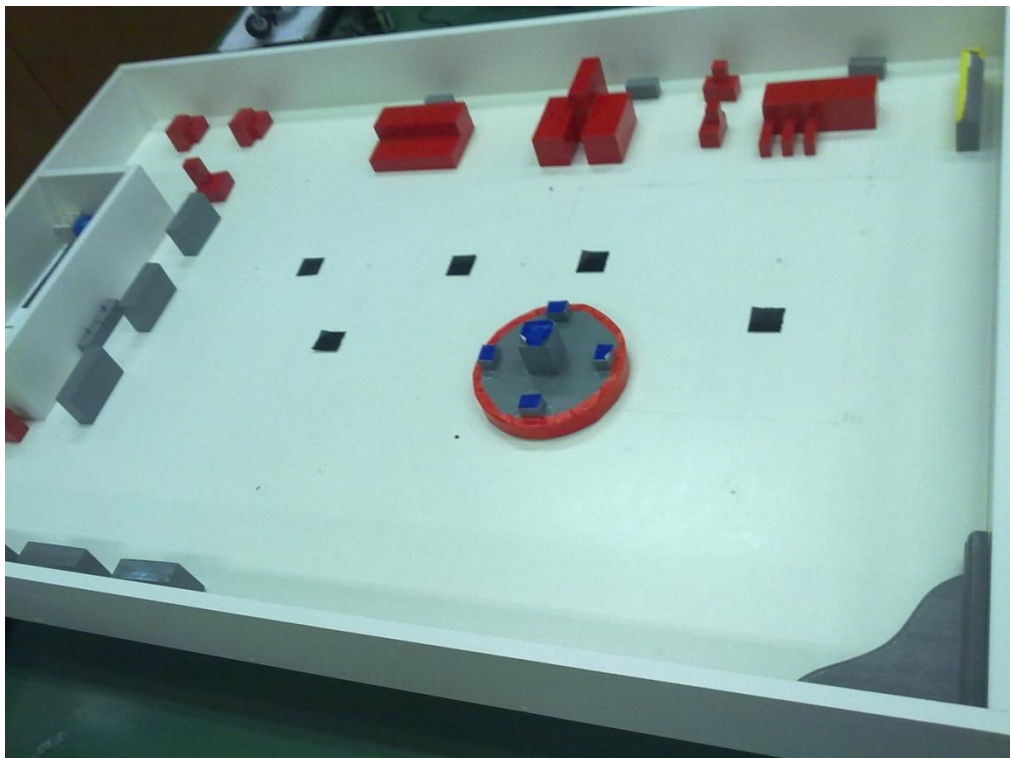
Следећи аспект који се у погону издваја је свакако монтажа. У овом случају реч је о ручној монтажи која је с обзиром на неколико различитих типова готових производа који се формирају на палети изискивала два диспозициона простора за монтажу. Један је са четири радна места у кружном распореду и он је предвиђен за веће ормане (позиција 20), док је други са два радна места за мање ормане (позиција 19). Свака од целина допуњена је још и радним столом за формирање подсклопова (позиција 18), као и складиштем вијчане робе и компонената (позиција 22), складиштем полипроизвода (позиција 21) и складиштем готових делова од бакра (позиција 23).

Овако формиран распоред у предузећу "XY" верно је пренет на модел који смо користили за експерименталну имплементацију нашег решења. Разлика је једино у томе што позиција 18, односно сто за монтажу подсклопова у оквиру монтаже великих ормана није био постављен у наше окружење. Међутим, то не преставља непремостив проблем јер се та препрека врло једноставно може унети у матрицу коју смо користили за дефинисање дозвољених кретања.

Layout производно-монтажног погона и његов модел реализован у просторијама факултета приказани су на сликама 2.1 и 2.2 респективно.



Слика 2.1: Layout производно-монтажног предузећа "XY"



Слика 2.2: Лабораторијски модел окружења

3. Концепцијско решење пројектног задатка

Пројектовање транспортног система од великог је значаја за хармонично функционисање самог предузећа. Процењено је да са имплементацијом оваквог решења долази до великих уштеда у времену, материјалним средствима и другачијој ангажованости радника јер на манипулацију материјалом и на транспорт одлази чак 15 до 35 % укупних трошкова [4]. Увођење роботског мобилног система у првом тренутку била би већи материјални издатак, али би се временом инвестиција исплатила и довела до значајних уштеда.

3.1. Ток развоја пројектног решења

Почетак пројектовања везан је за упознавање са layout-ом предузећа "XY" који је описан у поглављу 2.1. Уочено је да мобилни робот мора обављати све задатке које су до сада обављали виљушкар и ручни хидраулични виљушкар. Мора се предвидети да би робот у реалним условима требало имати карактеристике које поседује и претходно коришћено транспортно средство. Ту је првенствено реч о одговарајућој носивости и о адекватним димензијама које би омогућиле да мобилни робот има у најмању руку једнако добар приступ свакој машини као и виљушкар, ако не и бољи. Ипак, у случају пројектног задатка није се до детаља разматрала ова проблематика већ је било неопходно пре свега извршити одговарајући експеримент у окружењу које је формирано у факултетским условима. Сходно томе након извршених анализа извршена је синтеза самог мобилног роботског система.

Конфигурација мобилног робота. Прво својство које је остварено је стабилност. Коришћена је структура са три тачке ослонца која је у исто време омогућавала и добру мобилност. Помоћни точак има три степена слободе која поспешују ротацију робота, те су на тај начин смањене грешке које би се јављале при кретању. Управљачки модул робота (енг. Intelligent Brick) имплементиран је у конструкцију тако да се конекције са моторима и сензорима остварују на што логичнији начин. Наиме, претходно је предвиђена позиција сензора и у њему најближу локацију постављени конектори да би се елиминисали поремећаји које може изазвати уплитање конектора и несигурност остварене везе. На сличном принципу формирана је веза и мотора и модула. Сам модул за конструкцију је повезан спојевима који обезбеђују његово симетрично постављање и стабилно пријањање на саму носећу структуру мобилног робота. Такође, за дефинисани проблем предвиђен је оптички сензор.

Кратак преглед потребних и остварених функција мобилног робота. Почетак управљања самим роботом заснован је на моделу кретања који је имплементиран у MatLab окружење. Следећа фаза обухвата обучавање вештачких неуронских мрежа преко обучавајућих парова за ротацију робота у односу на задате углове при кретању. Након тога потребно је обучити нову мрежу која врши детекцију боје и процес класификације, а која се касније имплементира у подпрограм за корекцију кретања. Задатак не би био потпун без завршне фазе која обухватала формирање A^* алгоритма. Овај алгоритам формиран је у циљу остварења жељених позиција на основу задавања почетне и крајње тачке уз корекцију положаја применом Калмановог филтера. На самом крају извршено је формирање главног интерфејса који је омогућавао дефинисање жељених тачака кретања без познавања самог кода програма.

4. Индустриски и мобилни работи, сензори и интелигентни технолошки системи

Постизање циљева које смо сагледали у оквиру дефинисања пројектног задатка може се остварити искорак у мобилну роботску са тенденцијом достизања што већег степена аутономности. Међутим, морамо имати у виду да је, ако желимо постићи жељено, потребно корачати постепено кроз знања о индустриским роботима, мобилним роботима, сензорици и наравно интелигентним технолошким системима. У наредним редовима у кратким цртама представљен је начин на који су се стицала знања из поменутих области и која су као крајњи продукт пружила могућност за израду овог пројекта.

4.1. Индустриски работи

Познато нам је да реч "Робот" не датира из научних дела већ да своје корене налази у научно фантастичној драми чешког писца Карела Чапека који је овај назив извео из чешке речи *робота* која се односи на присилан рад. Применом овог имена и лаик може наслутити које су његове функције. Али, тачне и детаљне одговоре на ову тему не можемо тражити у белетристици већ су нам потребне дефиниције које налазимо у стручној литератури.

Наводе се две дефиниције индустриских робота према [3]:

1. RIA¹ дефиниција:

Индустриски робот је вишефункционални манипулатор пројектован да помера материјал, делове, алате и специјалне уређаје кроз различита програмирана кретања за извршавање различитих задатака.

2. ISO дефиниција:

Индустриски робот је аутоматски управљана вишенаменска манипулациона машина са неколико степени слободе, која може бити фиксна или покретна, а користи се за аутоматизацију у индустрији.

Иако не постоји савршена дефиниција индустриског робота уочавамо да фигуришу две речи: програмабилност и флексибилност, уз недостатак још две: интелигенција и аутономност о којима ће бити речи у оквиру подпоглавља о интелигентним технолошким системима. Програмабилност нам говори да се програмирана кретања и помоћне функције могу мењати без физичких интервенција, док флексибилност омогућава прилагођавање различитим применама са или без физичких интервенција [3].

Осврћући се на разлог развоја оваквих система у исто време додирује се окосница флексибилне аутоматизације и порекло речи "робот". Наиме, принудни и тежак рад, тачније његово олакшавање и смањење ризика по човека, оно је што је инспирисало људе да теже ка развоју ове области. И постепено стигло се до савремене примене индустриских робота која има за циљ повећање продуктивности, подизање и одржавање константног нивоа квалитета производа, подизање флексибилности и хуманизацију рада [3]. Наиме, у почетној фази примене робота у индустрији они су углавном обављали оне послове који су монотони, на пример узимање радног предмета са одређеног места и постављање на машину, као и послове који се обављају у нездравим условима (просторије са високим температурама,

¹ RIA - Robotic Institute Of America

просторије контаминирани радиоактивним материјалом и слично). Уопште, то су послови од којих је човека пожељно ослободити, па тако роботи имају одређену улогу у поменутој хуманизацији рада. Илустративно се ови послови описују са три D и три H:

- **dull**-досадан монотон,
- **dirty**-прљав,
- **dangerous**-опасан.
- **hot**-вруће,
- **heavy**-тешко и
- **hazardous**-рискантно.

Ипак роботи су веома продуктивни, раде у више смена, раде уједначеним ритмом, праве мало шкарта те је у капиталистичком друштву које нас окружује главни мотив за њихову примену првенствено економски. Применом робота избегавају се појаве у производњи које се у аутомобилској индустрији наводе као "monday morning car".

Индустријски роботи се данас примењују у широком спектру области и функција од којих се издвајају [3]:

- Манипулација (опслуживање машина, палетизацију/депалетизацију),
- Обављање процеса (заваривање, обрада резањем, сечење ласером, воденим млазом),
- Манипулација и обављање процеса (монтажа) и
- Специјални задаци (мерење и контрола)

Индустријске роботе можемо поделити на различите начине према [3]:

- Намени,
- Степену универзалности,
- Кинематичким, геометријским и енергетским параметрима,
- Методама управљања и тд.

Према JARA² класификацији индустријски роботи се по типу управљања деле на следеће класе [3]:

- Ручни манипулациони уређаји: то су уређаји са неколико степени слободе којима управља оператор,
- Секвенцијални роботи: манипулациони уређаји са фиксним и променљивим секвенцијалним управљањем,
- Понављајући роботи: оператор вођењем или са пулта, извршава задатак водећи робот који меморише трајекторије и касније их понавља,
- Нумерички управљани роботи: програмирају се текстуалним језицима и
- Интелигентни роботи: на бази информација са сензора и техникама вештачке интелигенције разумеју задатак и околину и доносе одлуке у реалном времену.

Роботи се даље деле и на генерације [3]:

- Прва генерација - програмски роботи,
- Друга генерација - адаптивни роботи и
- Трећа генерација - интелигентни роботи.

Предмет нашег изучавања у оквиру овог курса су роботи који спадају у категорију интелигентних робота.

² JARA - Japan Robot Association

4.2. Мобилни роботи

Мобилна роботика се разликује од индустријске роботике и обухвата шири спектар проблема којима се може приступити. Еволутивно гледајући она представља један корак напред у развоју робота којим он попут човека који се креће постепено добија своју "мускулатуру".

Из овог разлога у покушајима да се дефинише мобилни робот наилази се на веће проблеме него код индустријског робота јер се овде мора имати у виду појам мобилности и сва питања која она отвара, а о којима ће више бити речи у одељку о интелигентним технолошким системима. Из тог разлога дефиниција индустријског робота не може бити строго примењива и на мобилне роботе.

Мобилни робот тако можемо дефинисати као мобилан и манипулативан физички систем који се креће кроз неструктурирани простор, остварујући притом интеракцију с људским бићима или обављајући неки посао уместо њих.

На овај начин они се посматрају као тзв. услужни роботи. Услужни роботи би временом требало да се развију у персонализоване роботе који ће умногоме личити на људе. Наиме са становишта величине тржишта робота кућни робот представља свети грал тржишта робота.³ Ипак, до тада будућност индустријске роботике у великој мери лежи у мобилним роботима.

Мобилни роботи се могу класификовати по неколико независних карактеристика, од којих свака у великој мери одређује кључне аспекте њиховог система управљања и навигације. Тако имамо класификацију према преносном медијуму:

- ваздушни,
- водени,
- свемирски и
- копнени.

Друга подела мобилних робота је извршена са аспекта врсте локомоције (кретања):

- Роботи са точковима (енг. wheeled),
- Роботи са ногама (енг. legged),
- Гусеничари (енг. tracked) и
- Змијолики роботи (енг. serpentine).

Постоје још и поделе према:

- Врсти терена коју савладавају (унутрашњи или спољашњи простори),
- Флексибилности тела робота (једно и вишетелесни роботи, роботи с еластичним или крутим телом),
- Облику тела робота (роботи с једноставном или сложеном структуром, роботи у облику инсеката, итд.),
- Примени (роботи за едукацију, истраживачки роботи, роботи за пружање услуга, роботи за забаву, итд.),
- Начину настанка (модернизована стара возила, нова возила) и
- Степену аутономности (Крећу се од телеоператора до робота са пуном аутономношћу).

³ енгл. "From a perspective of market size, the home robot represents the Holy Grail of robot markets" - J.M. Holland, Designing Autonomous Mobile Robots.

Мотиви развоја оваквих робота на самом крају могу се таксативно навести кроз следеће ставке:

- Могућност приступа местима која су за човека: опасна по живот (нпр. минска поља, нуклеарна постројења, експлозивне зоне итд.), превише удаљена (нпр. Марс) или недоступна (нпр. микроскопски простори),
- Смањење трошкова и повећање продуктивности рада: мањи додатни трошкови рада због смањења администрације, потребног простора и сл., већа свеукупна расположивост робота (нема пауза за ручак, годишњих одмора итд.) бржи рад у односу на човека, нпр. брже бојење, сечење, кретање итд., повећан квалитет производа или услуге и
- Старење становништва у развијеним земљама.

Примена им је стога шира од индустријских робота и може бити у:

- Индустрији
- Војној примени
- Домаћинству
- Осигурању објеката
- Истраживању свемира, подморског света и планета
- Пољопривреди
- Сервисима
- Помоћи старијима и немоћнима
- Чувању и одгоју деце

4.3. Сензори

Говорећи о дефиницијама робота истичу се две карактеристике робота које у великој мери одређују његове особине и функционалност. Самосталност и интелигенција робота доводе до високе флексибилности и успешне примене. Да би робот могао радити самостално, без обзира на ниво самосталности, он мора бити "свестан" себе и своје околине. Прецизније речено, робот мора имати могућност мерења сопственог положаја и брзине као и мерења различитих величина у радној околини чиме стиче представу о спољном простору. Различити мерни уређаји и системи којима робот добија информације о себи и околини називају се сензори. Данас су то уређаји за мерење угаоног и транслаторног померања, различити сензори додир, уређаји за мерење растојања, силе, убрзања и сл. Посебну класу сензорских система чине визуелни системи, данас често у употреби. Како се од робота у будућности очекује изузетна самосталност, то је неопходан услов за то напредак у сензорској техници као што је, на пример, чуло додир са могућношћу одређивања глаткоће и мекоће површине, тродимензионална визија, чуло слуха са препознавањем говора итд. Генерално говорећи сензори у роботизи омогућавају [3]:

- Мерење параметара у управљачким повратним спрегама,
- Налажење објеката (препознавање, позиција и оријентација објеката),
- Корекцију грешака у моделу робота и околине,
- Откривање и решавање проблема у погрешним ситуацијама,
- Мониторинг интеракције са околином,
- Осматрање промена у околини које могу утицати на задатак и
- Инспекцију резултата процеса.

У роботизи сензори се деле на:

- унутрашње и
- спољашње сензоре.

Унутрашњи сензори дају информације о позицији, брзини, убрзању, сили и моментима. Спољашњи сензори представљају сензоре околине у које спадају тактилни сензори, сензори близине и сензори удаљености [3].

Такође, не сме се заборавити ни следећа подела сензора која узима у обзир енергетски утицај окружења на систем и обрнуто. У том погледу се разликују [2]:

- пасивне и
- активне сензоре.

Пасивни сензори врше мерење енергетског утицаја окружења на систем и у њих спадају микрофони и камере, док активни сензори обухватају оне сензоре који емитују енергију у окружење мерећи при томе одговор окружења на енергетски стимуланс. У овој групи се налазе ласерски и ултразвучни сензори [2].

4.4. Интелигентни технолошки системи

Говорећи о мобилном роботу уочено је да се са кретањем јавља много питања на која је потребно одговорити да би робот на жељени начин остварио локомоцију. Где се налази? Куда иде? Како до тамо доћи? Како избећи сударе са другим објектима? Закључено је да робот сам формира свест о простору у ком се креће коришћењем својих сензора. Уз то се мора имати у виду да он при томе обавља задатке различитог степена сложености. Свођењем свих чињеница закључујемо да се ради о интелигентном технолошком систему. Сходно томе извршићемо дефинисање интелигентног технолошког система према [4]:

Интелигентни технолошки систем је највиша класа флексибилних технолошких система која је остварила синергију вештачке интелигенције и компјутерски интегрисаних технологија, са циљем да систем има могућност реализације активности у неодређеном технолошком окружењу, уз перманентан пораст вероватноће успешног понашања.

Наш мобилни робот представља синтезу два својства: мобилности и интелигенције. На овај начин он се у великој мери приближава природним системима тако да се између подсистема робота и људског организма могу повући видљиве паралеле.

5. Опис расположивих компоненти

На самом почетку, по разумевању пројектног задатка било је неопходно саставити конфигурацију мобилног робота која би одговарала функцији постављеног проблема. На располагању смо имали мобилни робот MINDSTORMS® NXT, који је компанија LEGO® представила крајем јула 2006. године. То је програмабилни роботски комплет, као замена за прву генерацију Lego Mindstorm пакета који је назван Robotics Invention System који је приказан на слици 5.1.

Да би робот био мобилан и могао да изврши постављен задатак потребно је употребити следеће елементе NXT пакета: Управљачка јединица, оптички сензор, погонски систем, носећа конструкторска конструкција, и наравно напајање.



Слика 5.1: Стара и нова генерација Lego Mindstorm пакета

Главни део поменутог пакета је четвртасти компјутер звани NXT Intelligent Brick (Управљачка јединица, слика 5.1, десно). Она може да прима улазе од четири сензора (преко монодирекционих портова), и управља три мотора преко RJ12 кабла који је веома сличан, али није и компатибилан са, телефонским кабловима RJ11. Управљачка јединица има резолуцију 100x64 пиксела једнобојног LCD дисплеја са четири тастера којима се управља интерфејс хијерархијског менија. Такође поседује и звучник са кога се може чути аудио запис снимљен до 8kHz. Напаја се са 6AA батерија од по 1.5V. Управљачка јединица поседује могућност тестирања сваког од портова, били то сензорски портови или портови за покретање мотора, преко имплементираних интерфејса. Овај део је, практично, „мозак“ целог пакета.

Оптички сензор детектује ниво осветљења у једном смеру, а такође поседује и једну LED која служи као извор светлости која обасјава објекат са ког сензор детектује количину рефлектоване светлости. Сензор може да одреди и амбијенталну светлост. Мерни опсег сензора је од 0 до 100 NXT јединица, где се вредност 0 добија када је ниво осветљења најмањи а 100 када је највећи. Овај сензор детектује само нивое сивог, док новије изведбе имају могућност читавања RGB боја. Проблем овог сензора је у томе што сам квалитет сензора је на јако ниском нивоу, што онемогућује прецизно мерење. У различитим деловима дана читавања са сензора су драстично промењена за исту боју, јер свака промена осветљења мења рефлексију боје што утиче на добијене резултате. Применом овог

сензора, робот уз коришћење вештачких неуронских мрежа кроз примену софтвера *Matlab*[®] може одредити/потврдити своју позицију у познатом окружењу. Постављањем референтних тачака различитих боја, сензор би био у могућности да дефинише тренутни положај на основу података прикупљених са енкодера мотора и претходне позиције. Такође, светлосни сензор, детектовањем боје, може правити селекцију делова који долазе на покретној траци, врсти препреке којој се приближава, може разликовати да ли је предмет испред робота препрека, машина, или палета са које треба да изврши депалетизацију или палетизацију. Ово је сензор који је коришћен у изради пројекта и његов изглед је представљен на слици 5.2.



Слика 5.2: Оптички сензор

На слици 5.3 приказан је изглед звучног сензора. Овај сензор, као и претходни, долази у основном пакету. Уз помоћ њега робот може да детектује и реагује на промену звука, односно на промену ваздушног притиска. Опсег мерења је до 20kHz што је изражено на скали од 0 до 100 NXT јединица.



Слика 5.3: Звучни сензор

Овај NXT пакет такође поседује и ултразвучни сензор, слика 5.4. Он може да мери дистанцу до објекта у који је уперен као и да детектује померање објекта из окружења. Има могућност приказивања растојања у сантиметрима или инчима. Максимална дистанца коју је у могућности да измери је 233 центиметра са толеранцијом 3 центиметра. Ултразвучни

сензор ради тако што шаље ултразвучне таласе који се одбијају од објекат испред сензора и мери време које је потребно да се талас врати до пријемника сензора.



Слика 5.4: Ултразвучни сензор

На слици 5.5 приказан је изглед тактилног сензора. Он детектује притисак сензора, ударац или отпуштање сензора. У случају да је сензор активиран он региструје вредност 1, док у случају када сензор није активиран као свој излаз даје вредност 0.



Слика 5.5: Тактилни сензор

Овај комплет поседује три идентична серво мотора који имају интегрисан редукциони преносник са интерним оптичким ортним енкодером који има инкримент мерења од једног степена. Изглед овог мотора је приказан на слици 5.6.



Слика 5.6: Изглед мотора

5.1. Фазе формирања конфигурације робота

Конфигурација мобилног робота може бити изведена на више начина. Ми смо се определили за конфигурацију која ја сачињена од два погонска точка и једног помоћног. Сензори (и мотори) повезани су на NXT управљачка јединицу преко 6-пинских модуларних конектора који подржавају и аналогни и дигитални интерфејс. Аналогни интерфејс је компатибилан са претходним верзијама NXT пакета. Док је дигитални интерфејс је компатибилан и са I²C и RS-485 комуникацијом. Конектори и протокол комуникације су приказани на слици 5.7.

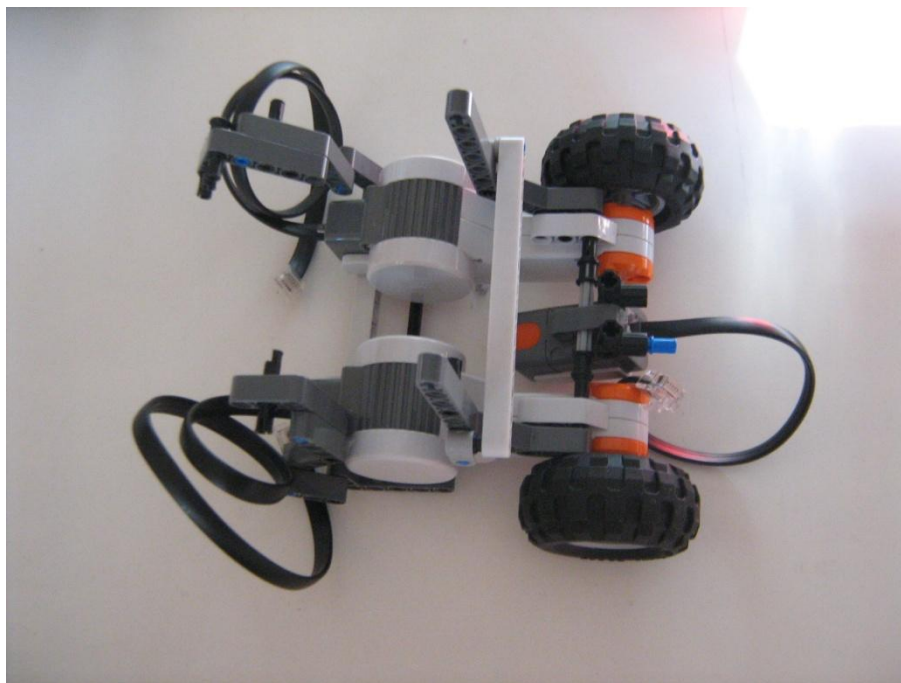
Pin	Name	Function	Color	Pin Numbering
1	ANA	Analog interface, +9V Supply	 white	
2	GND	Ground	 black	
3	GND	Ground	 red	
4	IPOWERA	+4.3V Supply	 green	
5	DIGIAI0	I ² C Clock (SCL), RS-485 A	 yellow	
6	DIGIAI1	I ² C Data (SDA), RS-485 B	 blue	

Слика 5.7: Опис протокола и изглед порта

Управљачка јединица поседује два типа интерфејса, и то

- Улазне (монодирекционе)
 - Port A
 - Port B
 - Port C
 - Port D
 - Улазно-излазне (бидирекционе)
 - Порт 1
 - Порт 2
 - Порт 3
- } Портови за конекцију са сензорима
- } Портови за конекцију са моторима

На слици 5.8 приказана је носећа струкура мобилног робота. То је био почетни део конфигурисања робота, и јако битан јер од ње зависи велики број фактора који утичу на добијање резултата. Уколико би конструкција била нестабилна и робусна долазило би до великих грешака приликом кретања робота у радном окружењу.



Слика 5.8: Носећа конструкција мобилног робота

Као најтежи део робота, управљачки јединица треба да буде што ближа тежишту целе конструкције. Из тог разлога је постављена на приказан начин на слици 9. Оптички сензор постављен на средини конструкције између точкова што му омогућава да не губи позицију приликом ротације робота.

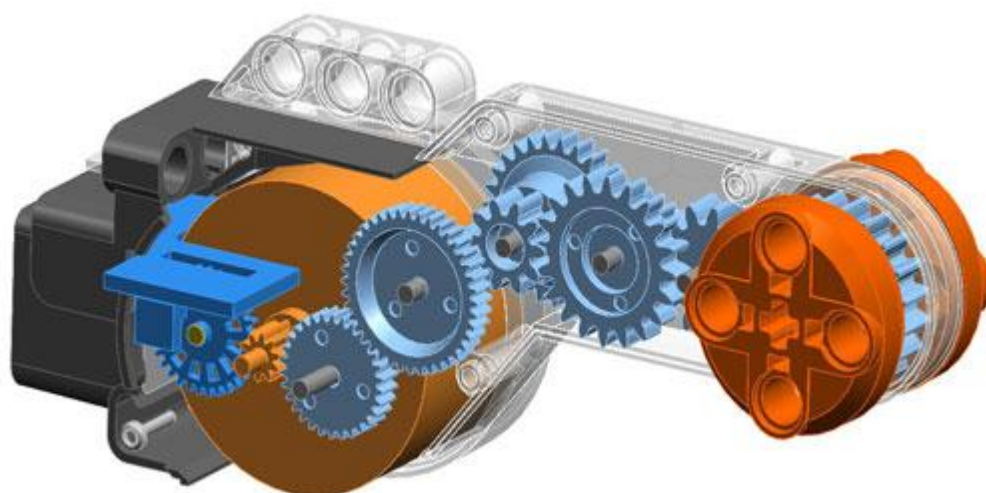


Слика 5.9: Приказ положаја управљачке јединице и оптичког сензора

5.2. Детаљан опис коришћених сензора

Сензори које смо применили за израду овог пројекта су оптички сензор и интегрисани енкодер мотора.

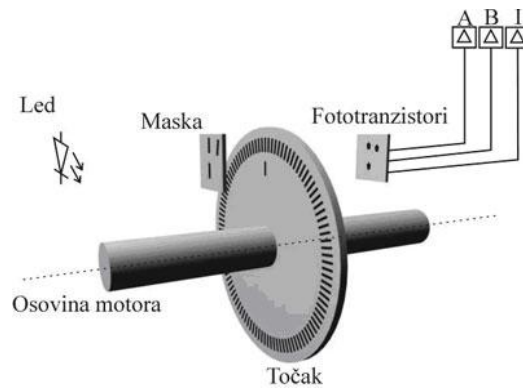
На слици 5.10 је представљен преносни систем и енкодер мотора. Енкодери представљају класу дигиталних сензорских система који се користе за мерења линијских и угаоних помераја. Највише су у примени оптички сензори. Могу се класификовати у две групе, и то у апсолутне и релативне енкодере. Апсолутни енкодери мере апсолутну позицију, док инкриментални енкодери мере релативан положај у односу на своју референтну тачку.



Слика 5.10: Приказ редукционог система и енкодера мотора

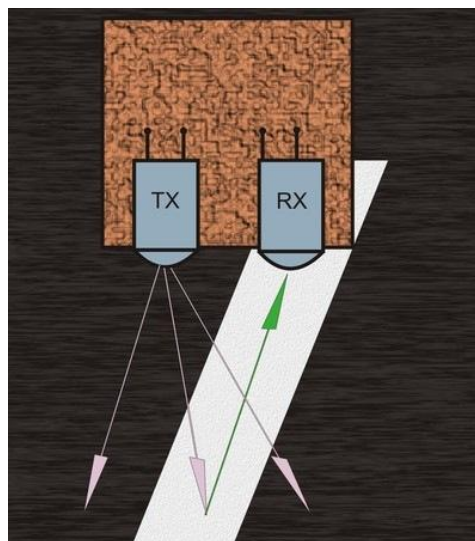
Инкриментални енкодер ради по принципу генерисања импулса приликом пропуштања светлости на сваком прозачном пољу подеоног диска енкодера. На тај начин, уз познавање тачног броја поља, уз услов да су поља истих димензија, може се одредити угаона позиција вратила на коме се налази енкодер. Прекид зрака подеоним диском фотодетектор генерише

на излазу сигнал који бива дискретизован и тиме је омогућена даља обрада сигнала. На слици 5.11 види се приказ једног типичног енкодера.



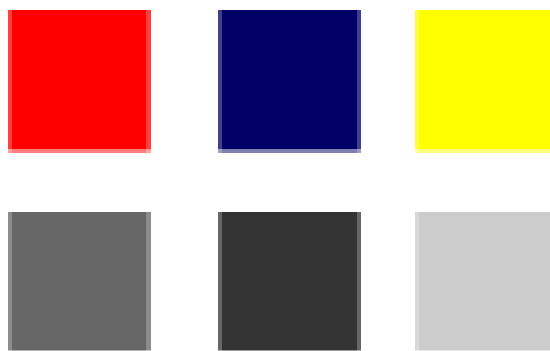
Слика 5.11: Шематски приказ инкременталног енкодера

Оптички сензор се најчешће састоји од предајника и пријемника. То је обично LED као извор светлости и пријемник. Светлосни зрак се емитује са диоде и рефлектује од површину. Ту рефлекцију прикупља пријемник, обично фототранзистор. Шематски приказ је на слици 5.12.



Слика 5.12: Рефлектовање зрака од површину

Пошто је оптички сензор у могућности да детектује само нијансе сиве, на слици 5.13 је представљено на који начин сензор „види“ основне боје.



Слика 5.13: Приказ на који начин оптички сензор детектује основне боје.

6. Модел кретања

Потребно је реализовати кретање робота у простору технолошког окружења. Робот треба да прати одређени технолошки поступак кретањем од машине до машине преносећи материјале, готове делове и полупроизводе. Укратко, његов задатак је опсуживање машина. Да би се остварио тај задатак, прво треба одредити математички модел кретања робота. Он треба да обезбеди познавање позиције и оријентације мобилног робота у сваком тренутку рада [2].

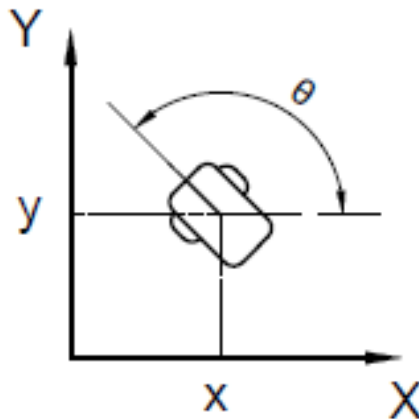
Кинематика робота се у општем случају може описати вектором стања који чине следеће компоненте:

$$x_t = \{x, y, z, \theta, \psi, \varphi\}^T \quad (6.1)$$

Где су x , y и z - компоненте вектора које дефинишу позицију мобилног робота, а θ , ψ , φ углови који одређују оријентацију мобилног робота у односу на неки спољашњи координатни систем. У овом случају, кретање робота ће бити моделирано као равно кретање, па је с тога довољно дефинисати вектор стања чије су компоненте следеће:

$$x_t = \{x, y, \theta\}^T \quad (6.2)$$

Где су x и y координате тежишта мобилног робота у односу на неки спољашњи координатни систем, а угао θ одређује оријентацију робота у односу на x осу (слика 6.1)



Слика 6.1: Дефинисање степени слободе мобилног робота у x , y равни

Модел промене стања (модел кретања робота) је представљен следећом релацијом [2]:

$$p(x_t / x_{t-1}, u_t) \quad (6.3)$$

Мобилни робот прелази из стања које је дефинисано вектором x_{t-1} у тренутку $t-1$ у стање x_t у тренутку t , управљањем u_t . Јасно је да робот, да би променио стање, мора променити позицију и оријентацију. То се једино може остварити кретањем [3].

Приликом дефинисања модела кретања мобилног робота на основу кога ће локализација бити извршена, могуће је дефинисати два основна правца:

- Модел кретања на основу брзина (брзински модел кретања)

У овом моделу, транслаторна и угаона брзина кретања мобилног робота представљају управљања $u(t)$ која изазивају промену позиције и оријентације током рада. Основна претпоставка овог модела кретања је да се транслаторна и угаона брзина кретања могу контролисати, односно да се њима може директно управљати.

- Модел кретања на основу пређеног пута (Одометрија)

У овом моделу управљање $u(t)$ је дефинисано пређеним путем тачкова мобилног робота током тачно дефинисаног временског интервала. На основу очитаних података са енкодера, и након интеграције, могу се одредити параметри пређеног пута неопходни за постављање модела. Међутим, посматрано са гледишта управљања роботом, пређени пут тачкова спада у мерења (сензорске информације) и у суштини представља резултат управљања и рада мотора, ипак у овом моделу кретања основна претпоставка је да управо пређени пут, као директан резултат управљања моторима мобилног робота, може бити усвојен за управљачку величину. Тако можемо преко управљачког софтвера мобилног робота директно контролисати кретање [2].

Општи кинематички модел кретања мобилног робота базиран на стохастички не постоји, тако да се за сваки случај појединачно врши анализа и формира модел. Поставка кинематичког модела зависи од конфигурације робота, карактеристика окружења у коме се робот креће и конфигурације алгорита. То значи да се модел кретања усваја и развија зависно од специфичног проблема [2].

6.1. Модел кретања на основу пређеног пута

Код овог модела кретања, мерења очитана са енкодера након интеграције, дефинишу се као управљања и прослеђују Калмановом филтеру у циљу одређивања позиције и оријентације мобилног робота [2].

Овај модел искључиво користи релативне координате, тј. мерени пут се мери помоћу енкодера, од неког положаја у коме је мобилни робот био у тренутку $t-1$ до неког положаја у тренутку t , што значи да је измерени пређени пут, онај који је робот прешао између ова два узастопна положаја. Наравно ова мерења треба пребацити у спољашњи координатни систем у коме је и дата мапа окружења, а у односу на који и треба извршити локализацију [2].

Модел који ће бити овде представљен је изведен на основу пређеног пута мобилног робота. Може да се примени искључиво на оне мобилне роботе који имају два погонска тачка која се могу кретати потпуно независно један од другог. То значи да управљањем смером обртања тачкова, контролише се скретање мобилног робота. Енкодери мере углове ротације на осовини тачка. Модел се може успоставити на следећи начин. Пошто се анализа

односи на раванско кретање, кретање робота се може описати са три независних параметра дата вектором :

$$x' = \{x, y, \theta\}^T \quad (6.4)$$

Одређивање позиције и оријентације за овакав тип мобилног робота своди се на интеграцију података очитаних са енкодера који се налазе на точковима, односно, сумирање пређеног пута погонских точкова у сваком инкременталном временском интервалу Δt [3]:

$$\Delta x = \Delta s \cos(\theta + \Delta\theta / 2) \quad (6.5)$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta / 2) \quad (6.6)$$

$$\Delta\theta = \frac{\Delta s_d - \Delta s_l}{b} \quad (6.7)$$

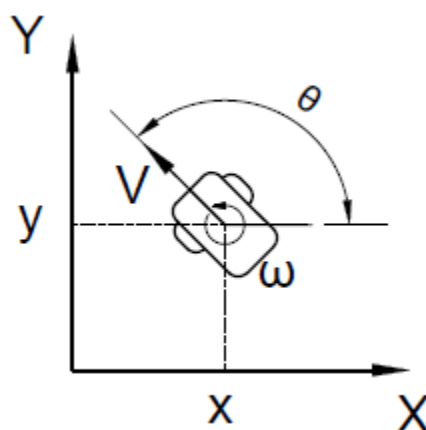
$$\Delta s = \frac{\Delta s_d + \Delta s_l}{2} \quad (6.8)$$

Где су:

$\Delta x, \Delta y, \Delta\theta$ - величине које дефинишу путању мобилног робота у току интервала Δt ,

$\Delta s_d, \Delta s_l$ - пређени пут десног и левог погонског точка респективно,

b - размак између точкова.



Слика 6.2: Мобилни робот са два независна погонска точка

На слици 6.2 је приказан основни модел, одакле се на основу геометрије једноставно одређује вектор стања мобилног робота након извршеног управљања:

$$x' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} \quad (6.9)$$

Даље се, након сређивања, одређује следећи израз:

$$x' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_d + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \\ \frac{\Delta s_d + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \\ \frac{\Delta s_d - \Delta s_l}{2b} \end{bmatrix} \Rightarrow x' = f(x, y, \theta, \Delta s_d, \Delta s_l) \quad (6.10)$$

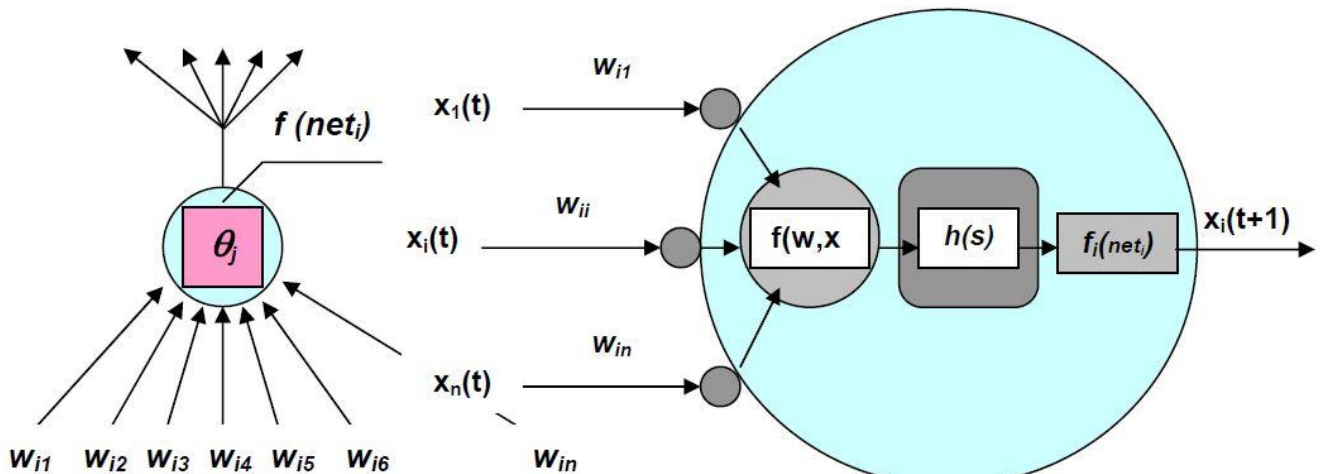
Што представља основни кинематички модел кретања мобилног робота изведен на основу прешеног пута погонских тачкова. Из горње једнакости може се видети које величине утичу на промену али и на могућности израчунавања позиције у оријентације. Претходни израз заправо представља модел кретања који фигурише у основним једначинама поменутог Калмановог филтера [3].

7. Вештачке неуронске мреже

Појавом првог облика интелигенције, људе је заинтересовала и као појам и као могућност њене примене кроз контруисање машина. Већ први резултати у примени рачунара упућивали су на могућност рачунара да преузму вршење одређених интелектуалних способности. Проблематика вештачке интелигенције је била доста распрегнута између фантастике, маште, потенцијалних могућности и практичних достигнућа. Занемаривање разлике између потенцијалне остварљивости и обима практичних проблема, који се налазе на путу до остваривања идеје, је један од честих узрока неразумевања могућности вештачке интелигенције.

Данас постоје реализовани системи који су у стању да аутономно обављају комплексне проблеме, какве су једино људи били у стању да обављају. У спречи са рачунарима, машине постају способне да раде самостално, да управљају саме собом и да производе друге машине, ослобађајући човека напорног физичког и монотоног рада, препуштајући му рад на сложенијим и креативнијим пословима. Циљ истраживања вештачке интелигенције је развијање софтвера који ће машинама омогућити да се понашају на начин који би се могао окарактерисати као интелигентан. Вештачка интелигенција је научна област у којој се изучавају прорачуни да би се њиховим решавањем омогућила перцепција, резоновање и правремена адекватна (ре)акција.

Вештачка интелигенција у овом пројекту је имплементирана кроз примену вештачких неуронских мрежа (ВНМ). Рад на вештачим неуронским мрежама је мотивисан управо са биолошким неуронским мрежама које се налазе у људском мозгу. Мозак је веома комплексан, нелинеаран и симултани „рачунар“ (систем обраде информација), који има способност да организује своје конструктивне елементе, познате као „неурони“ и приказан је на слици. Према [5] вештачке неуронске мреже обезбеђују значајне предности при решавању проблема процесирања који захтевају рад у реалном времену и интерпретацију међусобних односа између променљивих у вишедимензионалним просторима.



Слика 7.1: Неурон вештачке неуронске мреже

Најраспрострањенији развијени модели вештачких неуронских мрежа су [5]:

- Perceptron
- **Backpropagation (BP) неуронска мрежа**
- Асоцијативне неуронске мреже
- Hopfield-ове неуронске мреже
- ART неуронске мреже
- Fuzzy асоцијативне неуронске мреже
- Самоорганизујуће неуронске мреже.

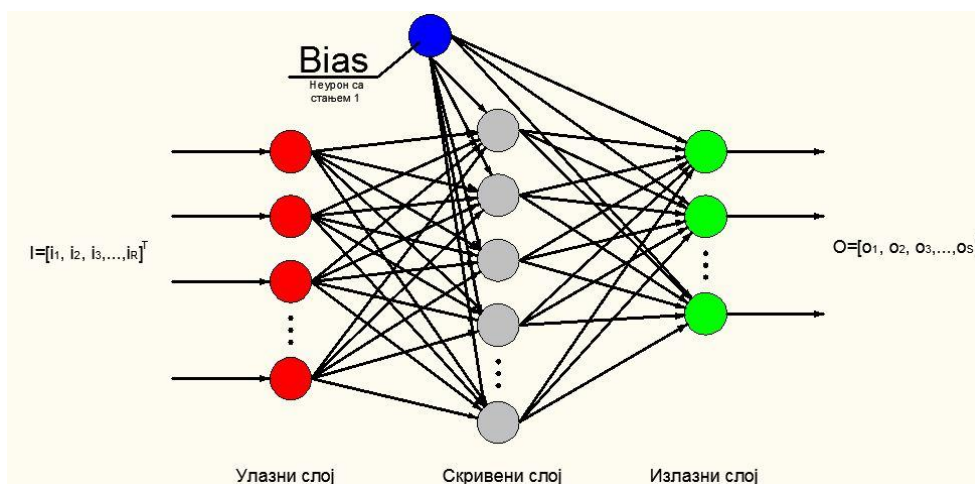
Постоји велики број ВММ и оне егзистирају још од почетка педесетих година прошлог века. Једна од првих вештачких неуронских мрежа је једнослојна мрежа „Perceptron“, чији је творац Frank Rosenblatt. Након њега многи су истраживали ту област вештачке интелигенције када се појавио и „Backpropagation“ алгоритам (Paul Werbos (1974), David Parker (1984/1985), и David Rumelhart, Ronald Williams, и други (1985)) који смо примењивали за решавање задатог проблема.

„Backpropagation“ (BP) неуронска мрежа је развијена са циљем да се проблем нелинеарног пресликавања из улазног простора у излазни простор успешно реши, при чему се остварује модификација тежинских односа и између улазног и скривеног слоја неурона. BP је, као и perceptron, неуронска мрежа са простирањем сигнала унапред, која такође реализује супервизорски вид учења, са различитим активационим функцијама и алгоритмима учења. BP мрежа користи *градијентни поступак* при обучавању, који је аналоган процесу минимизације грешке. То значи да учи пресликавања из улазног простора узорка у излазни простор, кроз *процес минимизације грешке* између актуелног излаза који је остварила мрежа и захтеваног излаза, на основу скупа обучавајућих парова, односно примера [5].

Значај неуронских мрежа је у томе да могу паралелно да обрађују податке, чије компоненте су независне једне од других. Истовремено ради више процесорских јединица, да би резултати њихове обраде прешли на следеће јединице (неуроне). Процесорске јединице у једној неуронској мрежи су једноставне и могу обављати само једно или евентуално неколико рачунарских операција и међусобно су повезани тако да у једној неуронској мрежи постоји много више веза него процесорских јединица. Број ових веза између неурона представља снагу неуронске мреже. Тежински коефицијенти веза су коефицијенти који су додељени у сваком тренутку везама неуронске мреже. Неуронске мреже се користе у ситуацијама када нису позната правила према којима би било могуће довести у везу улазне и излазне податке из жељеног система. Неуронске мреже се не програмирају, већ се тренирају, тако да је потребно доста времена за њихово обучавање, пре њихове примене. Коефицијенти веза између неурона подешавају на основу улазних података, тако да неурони „уче“ преко примера и поседују способност за генерализацију после унетог примера. Обучавање се врши тако што се ажурирају тежински коефицијенти да би се следећи пут добио излаз ближи задатој вредности. Када се обучавање заврши, тежински коефицијенти остају исти и мрежа се може применити за предвиђени задатак. Тренинг се састоји из тога да корисник задаје улазне и излазне вредности, а програм покушава да добије одговарајућу излазну вредност. При томе програм, у почетку, прави одређене грешке, а одговарајућом променом тежинских коефицијената, смањује се разлика добијене и задате излазне вредности [12].

Један од најважнијих приступа у одређивању оптималне вредности тежинских коефицијената W је алгоритам најмањег градијента. Основна идеја на којој почива овај поступак обучавања је одређивање оног градијента функције циља који једнозначно дефинише правац највећег опадања саме функције циља. На овај начин, приликом сваке итерације алгоритам тражи онај градијент функције циља који ће обезбедити испуњавање траженог услова [6].

Основни облик архитектуре BP неуронске мреже има један улазни, скривени и излазни слој, док број неурона варира од проблема који се решава применом ове вештачке неуронске мреже, слика 1. Исто важи и за број скривених слојева. Најчешће постоји само један скривени слој јер је BP мрежа са таквом структуром у стању да обезбеди репродуковање скупа захтеваних излазних облика за све обучавајуће парове [5], али не тако ретко се сусреће и архитектура са више скривених слојева. На слици 1 се може приметити да један неурон не припада ни једном слоју, и то као додатни улаз. Као такав, у литератури се среће као „Bias“. Наиме, он је увек активан и има излазно стање 1. Према [5], понаша се у мрежи као и сваки други неурон, при чему има задатак да партиципира у процесу учења тако што тежински однос између „Bias“ неурона и сваког неурона у следећем слоју формира активацију која мора бити савладана остатком улаза у сваки неурон, услед чега је и њихова активација контролисана. „Bias“ неурон обезбеђује константан члан у тежинским сумама неурона наредног слоја, што резултира побољшањем конвергентне карактеристике BP мреже.

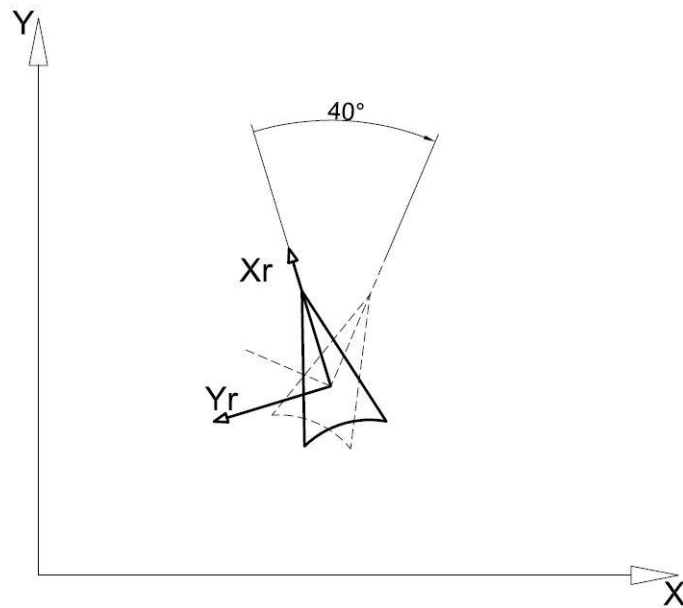


Слика 7.2: Основни облик архитектуре BP неуронске мреже

Поступак обучавања BP неуронске мреже, при коришћењу генерализаног делта правила, карактерише параметар учења η . Овај параметар се најчешће бира у опсегу од 0.25 до 0.75, и он одређује брзину обучавања, тако што је за веће вредности обучавање брже и обрнуто [5]. Оптимална вредност параметара се одређује емпиријски, и зависи од конкретног проблема који се решава применом ВМ.

При решавању задатог проблема примену вештачких неуронских мрежа видимо у два сегмента. Први је примена вештачке неуронске мреже за генерисање потребне ротације вратила мотора робота при дефинисању оријентације робота а други сегмент примене је детекција контролних тачака које служе за кориговање позиције робота због акумулирања грешке.

Прва фаза је обучавање неуронске мреже која ће генерисати потребан угао ротације вратила мотора у зависности од потребног правца и смера кретања добијених као излаз из алгоритма A^* , о коме ће бити речи даље у тексту.



Слика 7.3: Пример промене оријентације робота

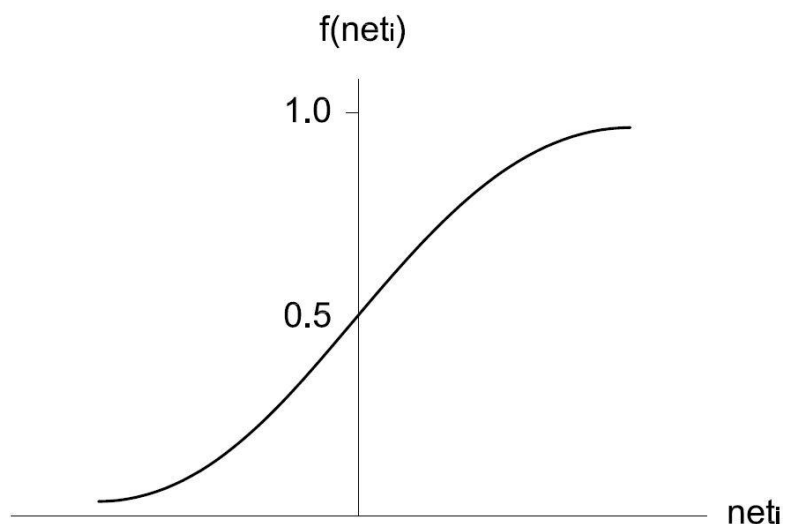
На слици 7.3 се види илустративни приказ робота који се заротирао око свог тежишта за угао од 40 степени, добијен као излаз из вештачке неуронске мреже. Пошто кретање робота обезбеђују два независна DC мотора, програмски и конструктивно је реализовано да при промени оријентације робота он ротира око свог тежишта које се налази на средини конструкције у оси тачкова на тај начин што се тачкови окрећу истом угаоном брзином али супротним смеровима. Обучавајући вектор мреже (улаз у мрежу) је потребан угао ротације робота док излаз из мреже треба да буде угао ротације вратила мотора. У табели 1 су дате вредности параметара обучавања мреже на основу којих се долази до закључка да зависност вредности обучавајућих парова није линеарна, као ни симетрична (вредности угаоне ротације тачка при промени оријентације робота у позитивном математичком смеру нису исте као за негативан математички смер). Али та несразмерност се може адекватно решити применом BP вештачке неуронске мреже.

Табела 2: Скуп обучавајућих парова ВМ за ротацију вратила мотора

Редни број обучавајућег пара	1	2	3	4	5	6	7	8	9	10	11	12
Потербан угао ротације робота (улаз у мрежу)	10	-10	20	-20	30	-30	40	-40	50	-50	60	-60
Угао ротације вратила мотора (излаз из мреже)	15	25	20	35	40	70	60	89	75	113	140	133
Редни број обучавајућег пара	13	14	15	16	17	18	19	20	21	22	23	24
Потербан угао ротације робота (улаз у мрежу)	70	-70	80	-80	90	-90	100	-100	110	-110	120	-120
Угао ротације вратила мотора (излаз из мреже)	160	153	178	170	203	186	220	205	240	223	26	243
Редни број обучавајућег пара	25	26	27	28	29	30	31	32	33	34	35	36
Потербан угао ротације робота (улаз у мрежу)	130	-130	140	-140	150	-150	160	-160	170	-170	180	-180
Угао ротације вратила мотора (излаз из мреже)	280	263	300	283	315	304	337	324	357	346	377	364

Одређивање потребног угла ротације вратила мотора за одређене углове ротације робота је вршено експериментално. На подеоној скали са инкриментом од 10 степени постављен је робот и итеративним поступком је одређивана вредност команде, за угао ротације вратила мотора, потребана да се робот ротира за вредност сваког подеока на мерној скали.

На тај начин добијени су обучавајући парови за тренирање мреже. При тренирању мреже параметри који су подешавани су број скривених слојева и број неурона у њима. Док су улазни и излазни слој имали константан број неурона (по један неурон у сваком слоју). Обучавање мреже омогућава подешавање, осим наведених, и још неких параметара. Могуће је дефинисати функције преноса за сваки скривени слој посебно, а у случају да то није експлицитно наведено, у Toolbox-у програмског пакета Matlab је дефинисана подразумевана вредност сваког параметра. Током обучавања комбиновали смо две врсте логаритамске преносне функције у зависности од броја скривених слојева мреже, и то функције „*logsig*“ и „*tansig*“. На слици 7.4 приказан је график преносне функције „*logsig*“.



Слика 7.4: График преносне функције „logsig“

Поред активационих функција дефинише се и алгоритам тренирања („*trainlm*“), у овом случају то је Levenberg-Marquardt-ов алгоритам [9]. То је један од најбржих функција тренирања ВР мреже у Toolbox-у програмског пакета Matlab, и препоручен је као први избор супервизорског учења и притом не захтева више меморије од осталих алгоритама.

Осим горе наведених променљивих за обучавање неуронских мрежа применом поменутог Toolbox-а, неопходно је дефинисати и параметар „show“, параметар учења, константе момента, број итерација, вредност задовољавајуће грешке и број валидационих грешака. У табели 4 су дате вредности и опис поменутих параметара обучавања.

Ради провере мреже, насумично су изабране вредности углова ротације робота за које мрежа није експлицитно обучена и приказане су у табели 3. Те вредности ће касније бити приказане на дијаграму, као и обучавајући парови и вредности симулације мреже за обучавајуће парове.

Табела 3: Вредности углова за које мрежа није обучена								
Вредности угла ротације робота	15	-18	45	-59	-63	98	139	-144
Редни број вредности угла	1	2	3	4	5	6	7	8

Табела 4: Вредности и опис параметара обучавања

Параметар обучавања	Опис параметра обучавања	Вредност параметра обучавања
show	Број итерација након кога се приказује промена	50
lr	Параметар обучавања (мања вредност брже учење)	0.1, 0.25, 0.4, 0.5, 0.75
mu, mc	константе момента	0.9, 0.04
epochs	Број итерација након кога се обучавање зауставља	10000
goal	Величина грешке обучавања између захтеваног и оствареног излаза	10^{-5}
max fail	Број дозвољених „падова“ на тесту приликом валидације	500

У циљу што боље имплементације вештачких неуронских мрежа обучаван је велики број неуронских мрежа а изабраних 15-ак је представљено у табели 5. Док је мрежа са најбољим резултатима коришћена у програму. Табелом 4 представљена је и архитектура вештачке неуронске мреже са коришћеним параметром учења.

Табела 5: Параметри обучавања ВМ

Редни број	Број скивених слојева	Број неурона по слоју	Параметар учења
1	1	5	0.1
2		8	0.25
3		10	0.4
4		13	0.5
5		15	0.75
6	2	3; 5	0.1
7		5; 7	0.25
8		5; 4	0.4
9		10; 8	0.5
10	3	15, 4	0.75
11		4; 5; 5	0.1
12		3; 8; 4	0.25
13		10; 5; 9	0.4
14		12; 6; 5	0.5
15		15; 8; 10	0.75

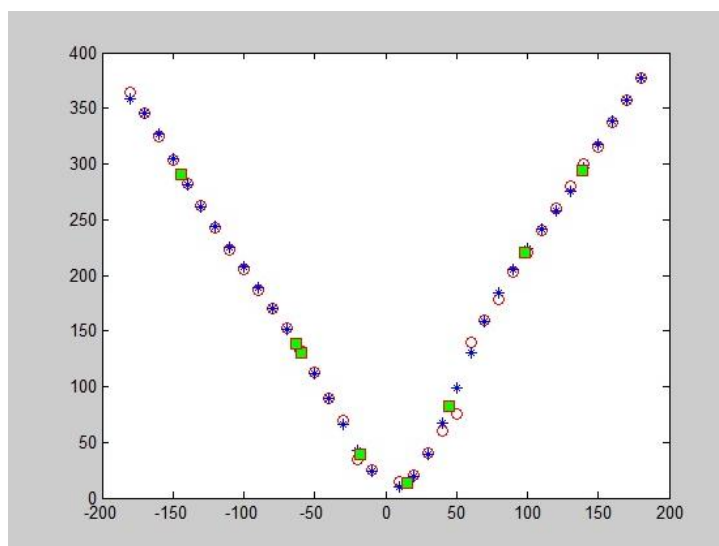
Упоредивањем добијених неуронских мрежа, на основу графичког приказа обучености мреже одабрали смо три мреже које ћемо даље разматрати. Свака мрежа је тестирана улазним величинама за које није обучена. Резултати тог тестирања представљени су на графичком приказу добијеном из Toolbox-а. На истом, приказани су обучавајући парови вештачке неуронске мреже црвеним кружићима, резултати симулације мреже плавим звездицама, и резултати тестирања мреже за необучене параметре, дате у табели 2, зеленим квадратима.

У табели 6 су дате вредности добијене обучавањем мрежа за сваки од експеримената узетих у обзир. Упоредивањем остварене грешке, остварене вредности градијента као и графичког приказа одабрали смо експерименте број 4, 6 и 8 као најбоље.

Табела 6: Упоредивање добијених резултата обучавањем ВНМ

Редни број експеримента	Број итерација	Завршен процес минимизације грешке	Остварена минимална грешка	Остварена вредност градијента	Број негативних валидација мреже
1	57	Не	$0.0549 \cdot 10^3$	3.2	50
2	510	Не	$0.00299 \cdot 10^3$	1.88	500
3	520	Не	$0.0129 \cdot 10^0$	0.0129	500
4	10	Да	$2.17 \cdot 10^{-7}$	0.0617	0
5	216	Да	$1.23 \cdot 10^{-3}$	0.215	67
6	514	Не	$0.0161 \cdot 10^3$	0.503	500
7	316	Да	$5.92 \cdot 10^0$	0.363	296
8	1534	Да	$1.52 \cdot 10^{-7}$	0.0359	2
9	946	Не	$0.0958 \cdot 10^0$	0.143	500
10	78	Да	$0.645 \cdot 10^0$	0.0342	61
11	950	Не	$0.0198 \cdot 10^0$	0.376	500
12	536	Да	$0.130 \cdot 10^0$	1.63	405
13	120	Да	$0.0279 \cdot 10^3$	0.183	94
14	56	Да	$0.352 \cdot 10^3$	0.00827	52
15	10	Да	$0.0443 \cdot 10^3$	0.0809	4

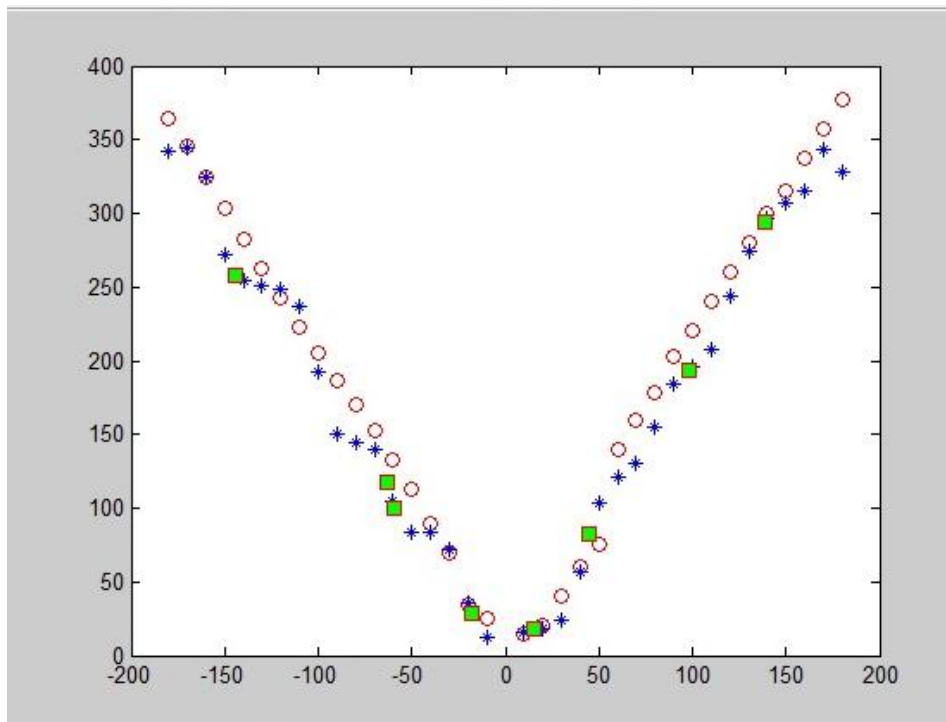
Даљом анализом одабраних експеримената долази се до закључка да је шести експеримент најрепрезентативнији и да њега треба имплементирати у управљачки код мобилног робота. С тога, на слици 7.5 приказан је графички приказ добијених резултата обучавања једне од три одабране вештетке неуронске мреже.



Слика 7.5: Графички приказ симулације једне од три посматране ВНМ

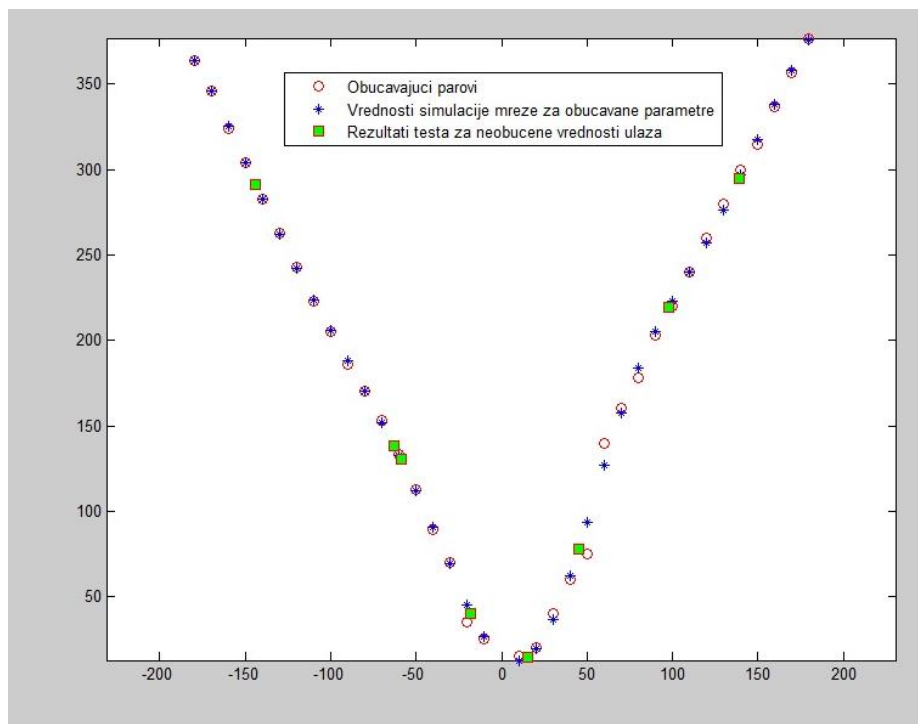
На претходној слици лако је уочљиво да је степен поклапања захтеваних вредности угла ротације вратила мотора (црвени кружићи) и остварених вредности (плавих звездица) веома велика и да мрежа као таква би била адекватна за имплементацију у управљачки код мобилног робота. Зелени квадратићи означавају резултате симулације ВНМ за необучене вредности угла ротације мобилног робота.

На слици 7.6 представљен је графички приказ симулације ВММ која није добро обучена на којој се јасно види да је поклапање захтеваних и добијених вредности знатно мање него код мрежа које имају висок ниво поклапања вредности обучавајућих парова и симулације мреже.

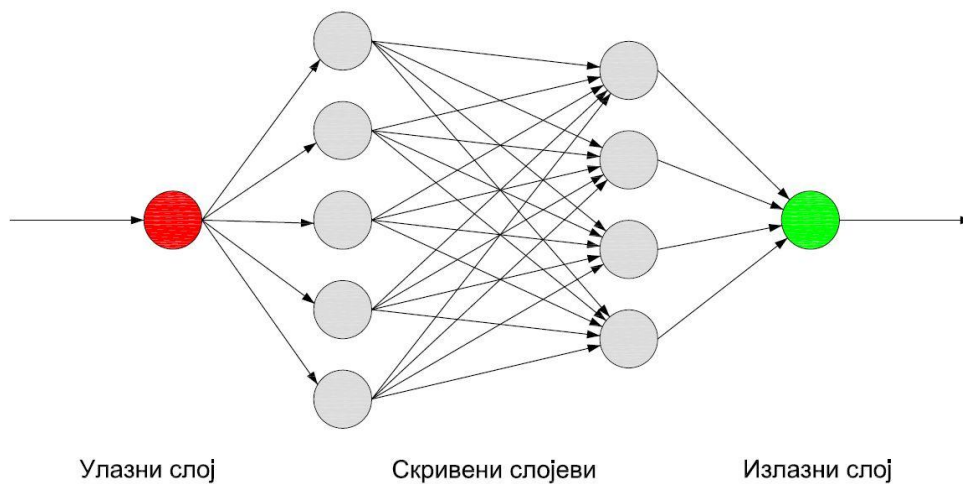


Слика 7.6: Графички приказ симулације једне ВММ која није добро обучена

На слици 7.7 је представљен графички приказ симулације ВММ која је имплементирана у управљачки код мобилног робота.

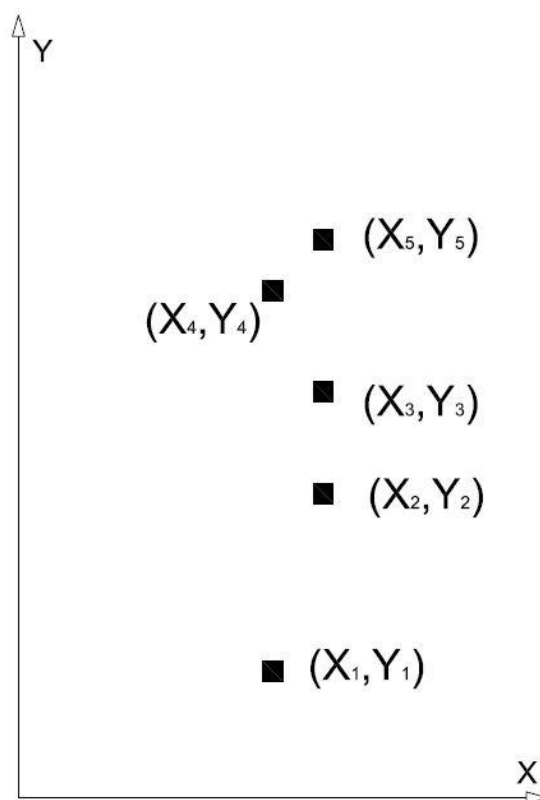


Слика 7.7: Графички приказ симулације ВММ која је имплементирана у управљачки код мобилног робота



Слика 7.8: Архитектура одабране вештачке неуронске мреже

Имплементирана ВММ је изабрана на основу параметара датих у табели 5, пошто графички прикази симулација репрезентативних мрежа није био довољан за оправдан одабир мреже. Стога, упоређивањем да ли је процес обучавања завршен, минимална грешка остварена и постигнута задовољавајуће мала вредност градијента поређењих експеримената дошли смо до закључка да је експеримент под редним бројем 8 оптималан за имплементацију у управљачки код мобилног робота. С обзиром на то да је експеримент под редним бројем 6 прекинут после задатог броја негативних валидација мреже, исти је искључен из даљег разматрања тако да сада репрезентативни скуп експеримената чине експерименти под редним бројем 4 и 8. Даљим анализирањем уочава се да експеримент 4 није имао ни један прекид због негативних валидација мреже али има нешто лошије вредности минималне грешке и вредности градијента, стога је изабрана ВММ под редним бројем 8. У табели 5 може се приметити да архитектуру мреже чине два унутрашња слоја од 5, односно 4 неурона у скривеном слоју и параметром обучавања 0.4. Резултати симулације одабране мреже приказани су на слици изнад. На слици 7.8 приказана је архитектура одабране вештачке неуронске мреже.

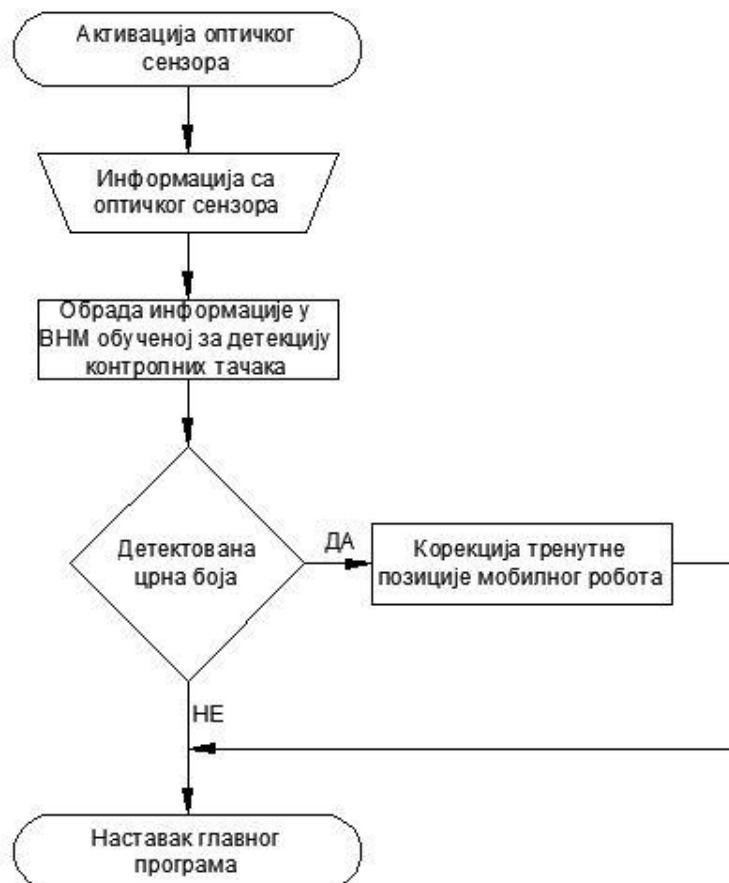


Слика 7.9: Приказ контролних тачака у радном окружењу

Друга фаза је обучавање неуронске мреже која ће детектовати контролне тачке у радном окружењу. Контролне тачке су референтне, док је остатак окружења беле боје, и оне служе за кориговање позиције робота (због акумулирања грешке) у случају да робот пређе оптичким сензором преко неке од контролних тачака. Координате сваке контролне тачке су познате и приказане су скицом на слици 7.9. У табели 6 дате су координате референтних тачака за прву и другу путању ко којој се мобилни робот креће.

Табела 7: Координате референтних тачака					
Број путање	Редни број	1	2	3	4
1	Координата тачке	(50,25)	(60,50)	(60,70)	(45,85)
2		(50,25)	(60,50)	(60,70)	(60,95)

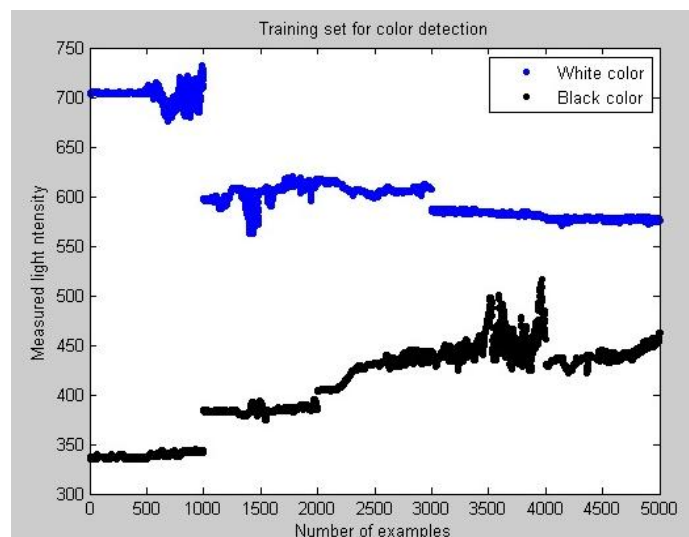
Примена ВММ при решавању овог проблема има задатак компензовања грешака при читавању сензора. Наиме, оптички сензор детектује ниво рефлектоване светлости од површину са које врше читавања. Због саме несавршености сензора долази до преклапања опсега вредности боја па сензор није у могућности да јасно препозна боју коју детектује. Грешке настају приликом промене амбијенталног осветљења или појаве сенке на месту читавања рефлексије светлости. На слици 7.10 приказан је алгоритам коришћења ВММ који показује ток информација при прецесирану података.



Слика 7.10: Алгоритам обраде података за детектовање промене боје у радном окружењу

Припрема обучавајућег вектора мора бити таква да скуп вредности рефлексије боја мора узети у обзир све интензитета светлости који се могу наћи при кретању робота у радном окружењу. Прикупљање вредности за обучавајући вектор вршено је експериментално, на тај начин што је робот био управљан у окружењу док је оптички сензор прикупљао податке. С обзиром на велики утицај осветљења просторије у коме сензор врши читавања, било је потребно симулирати утицај количине светлости у току дана као и утицај вештечког светла.

На тај начин је прикупљен обучавајући скуп чији графички приказ је представљен на слици 7.11.



Слика 7.11: Графички приказ добијених вредности читавања са сензора

Извршено је пет мерења под различитим условима, а свако мерење има по 1000. Тиме је направљен довољно обиман обучавајући скуп на основу кога је могуће добити задовољавајуће резултате при обучавању вештачке неуронске мреже. Након прикупљања вредности детектованих оптичким сензором можемо приступити плану обучавања мреже, који се види на табели испод.

У табели 7 се види да су параметри који се подешавају при тренирању мреже која ће детектовати боју по којој се мобилни робот креће само број скривених слојева и број неурона у њима, за разлику од претходне мреже која је била обучавана да одреди вредност угла ротације вратила мотора. У обзир је узето само два случаја, што се тиче броја скривених слојева, и то један скривени слој или два скривена слоја. Док је број неурона варирао као више променљивих вредности. Тако мали број експеримената је извршен јер су од самог почетка тренирања мрежа добијани задовољавајући резултати па није било потребно тражити мрежу која би дала боље резултате.

Табела 8: Мењани параметри при обучавању мреже		
Редни број	Број скривених слојева	Број неурона у слоју
1	1	2
2		5
3		8
4	2	10, 8
5		5, 15
6		5, 10

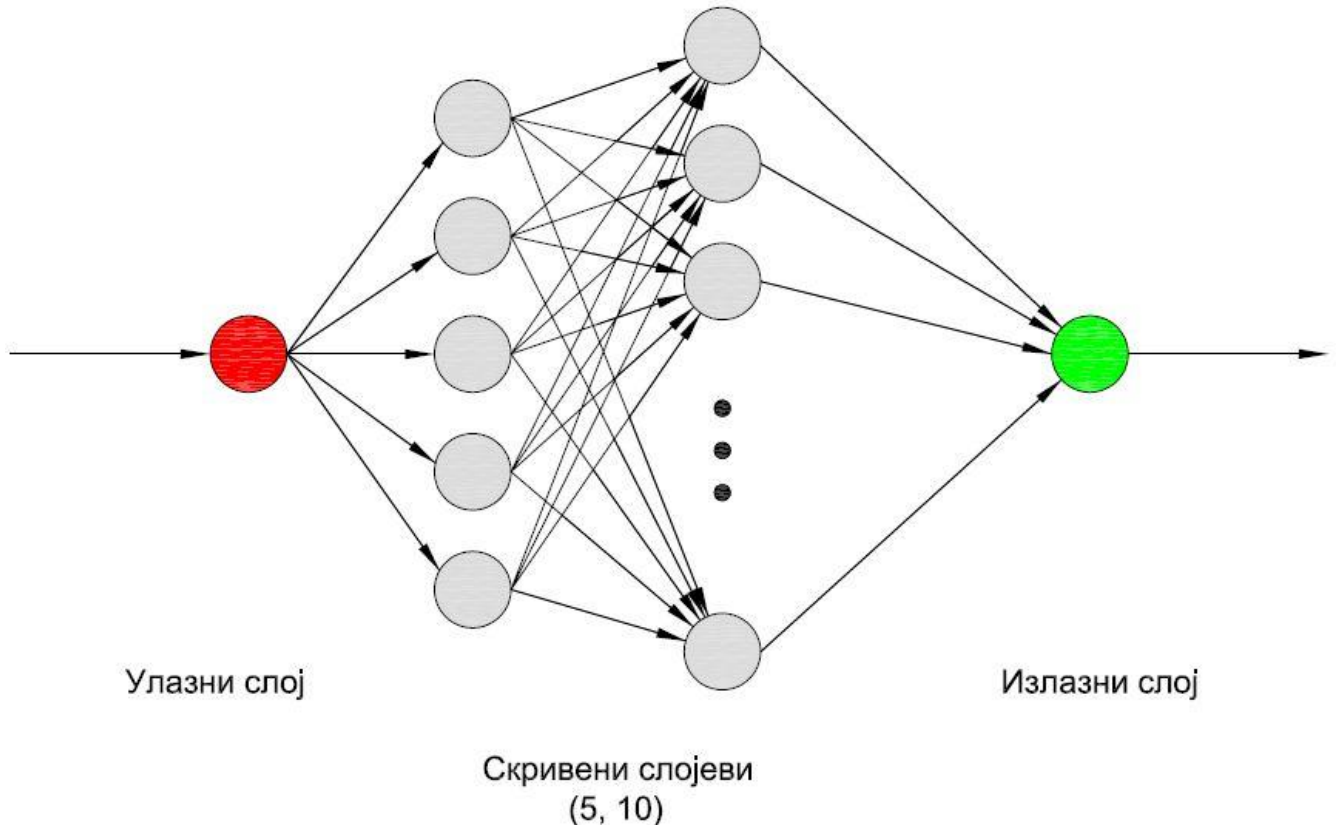
Упоређивање обучених мрежа врши се на основу добијених вредности минималне грешке и вредности градијента. Добијене вредности су приказане у табели 8.

Табела 9: Упоређивање добијених резултата обучавањем ВНМ					
Редни број експеримента	Број итерација	Завршен процес минимизације грешке	Остварена минимална грешка	Остварена вредност градијента	Број негативних валидација мреже
1	21	Да	$3.69 \cdot 10^{-15}$	$9.08 \cdot 10^{-10}$	0
2	23	Да	$3.43 \cdot 10^{-15}$	$5.51 \cdot 10^{-10}$	0
3	86	Да	$7.96 \cdot 10^{-9}$	$1.50 \cdot 10^{-05}$	0
4	216	Не	$5.33 \cdot 10^{-4}$	$9.62 \cdot 10^{-5}$	6
5	1000	Не	$2.92 \cdot 10^{-12}$	$1.05 \cdot 10^{-7}$	0
6	1000	Не	$1.39 \cdot 10^{-15}$	$1.66 \cdot 10^{-10}$	0

Одабир мреже ћемо извршити на основу добијених вредности приказаних у табели 8. Може се уочити да експерименти под редним бројем 1, 2 и 6 имају најбоље резултате, вредности минималне грешке и вредности градијента. Иако у шестом експерименту није завршен процес минимизације грешке, остварена минимална грешка је мања него код првог и другог експеримента. Анализом минималне грешке и остварене вредности градијента

поменутих експеримената долазимо до закључка да је оптимална мрежа добијена експериментом 6, те смо њу имплементирати у управљачки код мобилног робота.

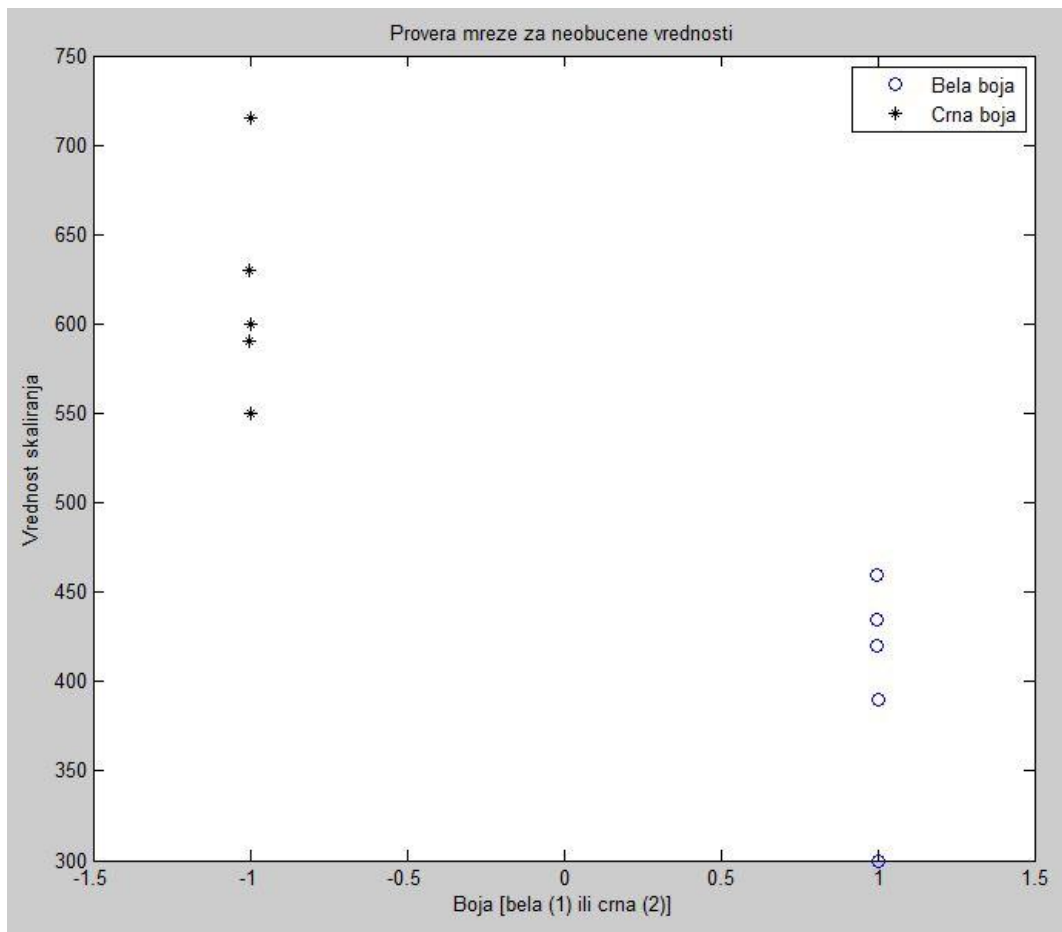
Из табеле 8 се види да изабрана мрежа садржи два скривена слоја. У првом слоју је пет неурона док је у другом слоју десет неурона. Таква архитектура је приказана на слици 7.12.



Слика 7.12: Архитектура ВММ имплементиране у управљачки код мобилног робота ради обраде информација са оптичког сензора

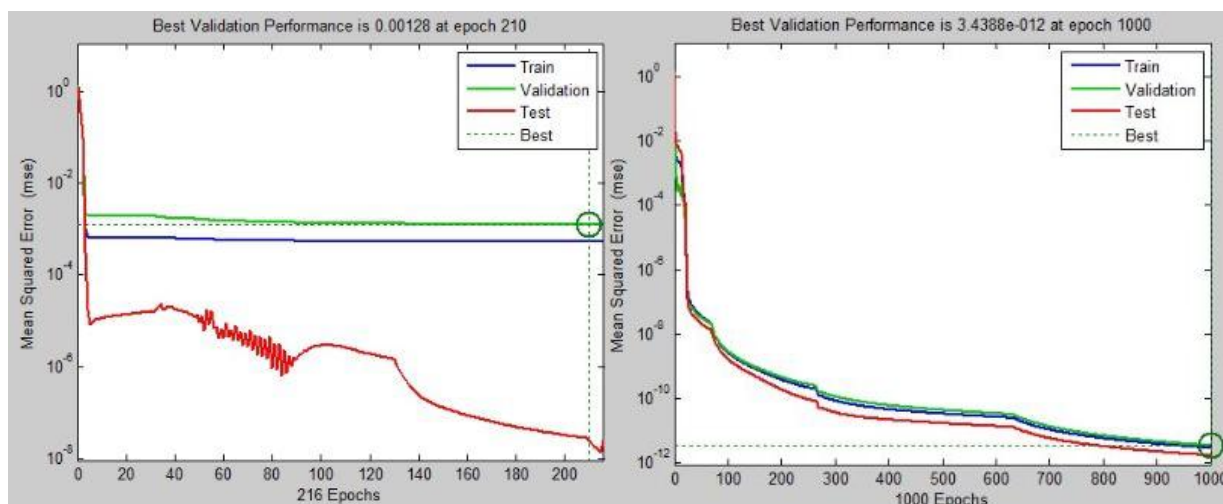
Табела 10: Предности тест скупа за које мрежа није обучена					
Редни број	1	2	3	4	5
Црна боја	300	390	420	460	435
Бела боја	590	715	630	600	550

Одабрана мрежа је тестирана за вредности које нису биле у обучавајућем скупу. Резултате које смо добили су били показатељ да је мрежа у могућности да да валидне вредности и за параметре који су на границама прелаза између црне и беле боје као и на границама мерног опсега оптичког сензора за које мрежа није била обучавана. Извођењем теста за необучене парове закључујемо да мрежа у потпуности даје валидне резултате што је илустративно приказано на слици изнад.



Слика 7.13: Графички приказ резултата симулације ВМ за необучене улазе

На основу приказаних резултата може се закључити да су симулиране ВМ у могућности да дају валидне резултате које можемо користити у раду мобилног робота у датом окружењу.



Слика 7.14: Графички приказ обучавања мреже. Ток тестирања лоше и добро обучене мреже.

8. Калманов филтер

Калманов филтер је добио име по Рудолфу Е. Калману који је представио овај математички апарат у свом раду, објављеном 1960. године. То је специјалан случај општијег нелинеарног филтера који је развио Руслан Л.Стратонович, шта више неке једначине за специјалан случај линеарних филтера је објавио у својим радовима пре него што се срео са Калманом на конференцији у Москви.

Понекад је потребно одредити неку физичку величину која се тешко може директно измерити. Рецимо да жељена величина представља непосредно недоступну променљиву стања система (уколико систем опишемо моделом с променљивим стања). До тражене величине потребно је доћи посредно. Притом проблем представља сам реалан систем чије понашање није до краја детерминистичко. Такође, мерење неке величине која нам служи за добијање жељене променљиве стања је често загађено шумом и непоуздано (мерена величина је заправо случајна променљива).

8.1. Теоријске поставке и једначине Калмановог филтера

Калманов филтер служи управо за екстракцију ове недоступне променљиве стања система из доступних извора (сигнал са шумом мерење излаза система и стохастички улаз са шумом у систем) и претходног стања система. Калманов филтер, као оптимални естиматор и предиктор непознате величине пронашао је велику примену у управљању система, навигацији, праћењу и предвиђању путање објекта. У нашем случају, употребићемо овај алат за управљање кретањем робота у технолошком систему за унутрашњи транспорт материјала. [2]

Примена Калмановог филтера за процену положаја робота у технолошком окружењу заснива се на примени модела кретања на основу пређеног пута. На основу овог модела добијају се вредности унутрашњих координата за задате вредности спољашњих координата, односно за дати положај и оријентацију мобилног робота. Једначине које описују зависност унутрашњих од спољашњих координата добију се решавањем инверзног кинематичког проблема.

Калманов филтер претпоставља да постоји временски дискретан процес којим управљамо, односно који надгледамо. Процес је описан линеарним моделом с променљивим стања и није потпуно детерминистички. Може се описати следећим једначинама:

$$\begin{aligned}x_{k+1} &= A_k x_k + B_k u_k + w_k \\ y_k &= C_k x_k + v_k\end{aligned}\tag{8.1}$$

где су:

x_{k+1} - стање система у тренутку $k+1$;

A_k - матрица која успоставља везу између претходног и тренутног стања;

x_k - стање система у тренутку k ;

B_k - матрица везе управљања и тренутног стања;

u_k - матрица управљања помоћу које дефинише како систем прелази из стања x_k у стање x_{k+1} ;

w_k - шум система, неодређеност у систему, случајни вектор, случајна променљива, поремећај на који не можемо утицати пошто је систем стохастички;

y_k - измерена излазна величина;

C_k - матрица која успоставља везу између тренутног стања и тренутног мерења;

v_k - шум мерења; [2]

Ознака x је n -димензионална променљива стања система. Ради се о величини која је инхерентна систему и која се не може непосредно измерити, а жели се одредити. То је задатак естиматора. Прва једначина говори о вези између вредности променљиве стања у тренутном и претходном кораку. Видљиво је да тренутна вредност x зависи од претходне вредности променљиве x помножене $n \times n$ димензионом матрицом A . Даље, тренутно стање зависи од улаза у систем из претходног корака представљеног l -димензионим вектором u , помноженом матрицом B димензија $n \times l$.

Осим ових детерминистичких чланова, тренутно стање зависи и од случајног члана w . Ради се о случајном вектору којим се моделира шум својствен процесу. Реч је о белом шуму (фреквенцијски је и временски независан) с нормалном расподелом: $p(w) \sim N(0, Q)$. Q је матрица коваријације шума процеса. Може се одредити поступком идентификације процеса.

Као што се види из горњег модела, и систем и мерење "загађени" су шумом, што онемогућава доношење поузданог закључка о тренутној вредности променљиве стања x овог система.

Задатак Калмановог филтера је управо да процени ту вредност. Уопштено говорећи, филтер ради у два корака:

1. **Предикција** - на темељу последњег познатог стања $k-1$ предвиђа се следеће стање k . Ова процена садржи грешку због шума, који притом није узет у обзир.
2. **Корекција** - након што се стање стварно промени у следеће, измери се излаз система (вектор y у стању k) и на основу тог резултата, матрица које описују понашање процеса и података о шуму коригује се процена стања x .

Након тога, циклус се понавља: предвиђа се следеће стање, мери се излаз, коригује процена следећег стања. Мерење, дакле, дели алгоритам реализације Калмановог филтера на два дела: део пре мерења (процена стања *a priori*) и након мерења (процена стања уз познато мерење, *a posteriori*). У наставку текста величине добијене пре мерења означене су знаком "-".

Први корак алгоритма може се описати следећим једначинама:

$$\begin{aligned} \bar{x}_k &= A\bar{x}_{k-1} + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \quad (8.2)$$

У првом кораку се процењује вредност променљиве стања система. Ова процена доноси се само на основу вредности *a posteriori* процене променљиве стања из претходног корака (најпрецизнија процена) и улаза из претходног корака. Квалитет ове процене зависи од квалитета модела којим је процес описан, као и утицају шума (моделовано вектором w)

који је у овој процени изостављен у стварном процесу. Будући да се овим кораком на основу претходних величина може предвидети стање система у тренутном кораку, он се може се назвати предикцијом (предвиђа следеће стање система уз познато претходно стање). Разликом стварног и *a priori* процењеног стања дефинише се *a priori* грешка процене:

$$e_k^- \equiv x_k - \bar{x}_k \quad (8.3)$$

На основу ове грешке може се дефинисати коваријација грешке *a priori* процене:

$$P_k^- = E[e_k^- e_k^{-T}] \quad (8.4)$$

Једначина (8.4) показује како се на темељу *a posteriori* коваријације грешке из претходног корака и коваријације шума система може израчунати коваријација грешке *a priori* процене за тренутни корак. Ово је потребно за други корак - корекцију. Други корак алгоритма своди се на корекцију процене на основу изведеног мерења. Мерење, дакле, игра улогу повратне спреге. Овај корак описују једначине:

$$K_k = P_k^- C^T (C P_k^- C^T + R)^{-1} \quad (8.5)$$

$$x_k = \bar{x}_k + K_k (y_k - C \bar{x}_k) \quad (8.6)$$

$$P_k = (I - K_k C) P_k^- \quad (8.7)$$

Средишња једначина у овом кораку је једначина (8.6). Она описује корекцију вредности процене: *a priori* вредности процене стања процеса додаје се корекциони члан $K_k (y_k - C \bar{x}_k)$, чиме се побољшава процењена вредност променљиве x у стању k . Разлика $(y_k - C \bar{x}_k)$ назива се иновација мерења и њоме се моделује разлика (непоклапање) између стварног мерења y_k и теоријски предвиђеног резултата мерења који следи из једначина процеса: $C \bar{x}_k$. K_k је фактор који се још назива и Калмановим појачањем. Ради се уствари о матрици димензија $n \times m$ чији је задатак да у *a posteriori* фази минимизује коваријансу грешке процене, која је дефинисана као:

$$P_k = E[e_k e_k^T] \quad (8.8)$$

уз

$$e_k \equiv x_k - \bar{x}_k \quad (8.9)$$

Може се доказати да K_k зависи од *a priori* вредности грешке P_k^- , матрице C и коваријансе шума мерења на начин описан формулом (8.5). Детаљнијом анализом види се да важе следеће релације:

$$\lim_{R_k \rightarrow 0} K_k = C^{-1} \quad (8.10)$$

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad (8.11)$$

Другим речима, ако се смањује грешка мерења, Калманово појачање добија такав облик који у једначини корекције наглашава резултат стварног мерења y_k јер оно постаје ближе стварној вредности излаза него симулирана вредност Cx_k^- . Такође, уколико варијанса грешке *a priori* процене тежи нули, појачање такође тежи нули, што значи да *a priori* процена стања постаје блиска коригованој процени. Што је предикција боља, то ће корекција *a priori* стања бити мања.

Будући да ће у следећем кораку предикције (за следеће стање) опет бити потребно знати коваријансу грешке из претходног корака (P_k из овог корака ће се у следећем циклусу означавати као P_{k-1}), P_k се на крају рачуна према формули (8.7).

Дакле, Калманов филтер је оптимални рекурзивни филтер који ради искључиво у временском домену. Карактерише га то што тренутни корак зависи само од претходног корака. Може се користити за интерполацију претходних стања, процену тренутног, као и предвиђање следећег стања система. Постоји више извођења Калмановог филтера. Тако се поједностављена верзија у којој се уместо матрица коваријанси користе скаларне вредности назива алфа-бета филтер, а постоји и општија верзија која се користи за управљање нелинеарним системима.

8.2. Алгоритам Калмановог и линеаризованог Калмановог филтера

Оптимална оцена, односно филтрирање, представља процес процене стања или параметара система на основу вредности добијених мерењем. На овим принципима се заснива Бајесов филтер, који је општији од Калмановог. Када је систем линеаран, шум адитиван и подлеже Гаусовој расподели из Бајесовог филтера се може извести Калманов филтер. Зато се може рећи да је Калманов филтер ефикасно рекурзивно решење за проблем дискретног линеарног филтрирања. За разлику од нерекурзивне естимације мерења где су сви резултати мерења већ унапред познати, рекурзивна естимација подразумева предвиђање стања на бази података који стално пристижу, тако да се и њихов број временом повећава.

На основу већ дефинисаних основних величина Калмановог филтера, може се написати следећи алгоритам [2].

Предикција:

улазне вредности: $\bar{\mu}_{t-1}, \bar{\Sigma}_{t-1}, u_t, y_t$

$$\bar{\mu}_t = A_t \bar{\mu}_{t-1} + B_t u_t \quad (8.12)$$

$$\bar{\Sigma}_t = A_t \bar{\Sigma}_{t-1} A_t^T + R_t \quad (8.13)$$

Корекција

Излазне вредности: μ_t, Σ_t

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} \quad (8.14)$$

$$\mu_t = \bar{\mu}_t + K_t (y_t - C_t \bar{\mu}_t) \quad (8.15)$$

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \quad (8.16)$$

Где су:

μ_{t-1} - средња вредност у тренутку t-1;

$\bar{\Sigma}_{t-1}$ - матрица коваријансе у тренутку t-1, тј. колико смо сигурни у μ_t ;

u_t - управљање у тренутку t;

y_t - мерење у тренутку t – читавање са сензора;

A_t - матрица која успоставља везу између стања претходног и тренутног корака без утицаја управљања и шума;

B_t - матрица која повезује управљање и стање система у тренутцима t и t-1;

$\bar{\mu}_t$ - предикција стања система;

$C_t \bar{\mu}_t$ - предикција мерења;

$\bar{\Sigma}_t$ - коваријанса;

Q_t - матрица коваријанси шума мерења, несигурност у мерењу;

Када се калманов филтер примењује на нелинеарним моделима, то се ради тако што се изврши линеаризација у околини радне тачке, тј. нелинеарну функцију представити линеарном у једној тачки. Такав Калманов филтер се назива линеаризовани Калманов филтер.

Једначина стања је тада нелинеарна функција управљања и претходног стања, што је приказано једначином (8.17). Такође, опсервациони модел је нелинеарна функција стања система што је приказано једначином (8.18). Функција $g(u_t, x_{t-1})$ показује како се мењају x, y, θ у функцији управљања и средње вредности.

$$x_t = g(u_t, x_{t-1}) \quad (8.17)$$

$$y_t = c(x_t) \quad (8.18)$$

Развијањем у Тејлоров ред линеаризујемо функцију у околини изабране тачке. Рачунајући само прва два члана реда изводи функције $g(u_t, x_{t-1})$ за изабрану тачку μ_{t-1} и функције $c(x_t)$ у изабраној тачки $\bar{\mu}_t$ имају следећи облик:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1}) \quad (8.19)$$

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1}) \quad (8.20)$$

$$c(x_t) \approx c(\bar{\mu}_t) + \frac{\partial c(\bar{\mu}_t)}{\partial x_t} (x_t - \bar{\mu}_t) \quad (8.21)$$

$$c(x_t) \approx c(\bar{\mu}_t) + C_t (x_t - \bar{\mu}_t) \quad (8.22)$$

G_t у линеаризованом Калмановом филтеру представља промену функције модела кретања у односу на положај робота (Јакобијан), док H_t представља промену мерења са променом координата положаја мобилног робота. Тако се врши калибрација сензора. Оба алгоритма важе уколико модел подлеже Гаусовој расподели. Ако то није случај, резултати добијени применом филтера нису довољно тачни.

Алгоритам линеаризованог Калмановог филтера

Предикција

Улазне вредности $\bar{\mu}_{t-1}, \bar{\Sigma}_{t-1}, u_t, y_t$

$$\bar{\mu}_t = g(u_t, \mu_{t-1}) \quad (8.23)$$

$$\bar{\Sigma}_t = G_t \bar{\Sigma}_{t-1} G_t^T + R_t \quad (8.24)$$

Корекција

Излазне вредности: μ_t, Σ_t

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} \quad (8.25)$$

$$\mu_t = \bar{\mu}_t + K_t (y_t - c_t(\bar{\mu}_t)) \quad (8.26)$$

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \quad (8.27)$$

На основу Калмановог појачања, $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$ могу се разматрати следећи случајеви:

- Уколико се вредност коваријансе повећава, расте несигурност, па расте и вредност Калмановог појачања. Такође, уколико опада вредност коваријансе грешке мерења, вредност Калмановог појачања се повећава.
- Аналогно, уколико се вредност коваријансе смањује, смањује се и вредност Калмановог појачања. Такође, уколико расте вредност коваријансе грешке мерења, смањује се вредност Калмановог појачања.

Ако се још једном погледа једначина $\mu_t = \bar{\mu}_t + K_t (y_t - C_t \bar{\mu}_t)$, која се односи на корекцију, може се закључити да део једначине представљен са $y_t - C_t \bar{\mu}_t$ односи на координату из мапе окружења. Ова разлика представља иновацију. Вредност y_t се добија мерењем, док је $C_t \bar{\mu}_t$ вредност која се добија предвиђањем. Уколико је вредност грешке сензора велика, вредност Калмановог појачања ће бити веома мала, па ће та грешка, помножена са малом вредности Калмановог појачања такође имати малу вредност и неће утицати на вредност жељеног сигнала енкодера.

8.3. Имплементација линеаризованог Калмановог филтера за потребе одређивања положаја мобилног робота током експлоатације

Модел општег кретања дефинише се преко положаја и оријентације робота вектором $x_t = (x, y, z, \theta, \psi, \varphi)$. За наш случају раванског кретања, вектор поприма једноставнији облик

$$x_t = (x, y, \theta).$$

Где су:

x, y, z - компоненте вектора стања које одређују позицију мобилног робота;

θ, ψ, φ - компоненте вектора стања које одређују оријентацију мобилног робота;

Локализација применом линеаризованог Калмановог филтера – корак предикције:

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,z}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,z}} \\ \frac{\partial z'}{\partial \mu_{t-1,x}} & \frac{\partial z'}{\partial \mu_{t-1,y}} & \frac{\partial z'}{\partial \mu_{t-1,z}} \end{pmatrix} \quad (8.28)$$

Јакобијан функције g у односу на положај робота

$$V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial z'}{\partial v_t} & \frac{\partial z'}{\partial \omega_t} \end{pmatrix} \quad (8.29)$$

Јакобијан функције g у односу на управљање

$$M_t = \begin{pmatrix} (\alpha_1 |v_t|) + (\alpha_2 |\omega_t|)^2 & 0 \\ 0 & (\alpha_3 |v_t|) + (\alpha_4 |\omega_t|)^2 \end{pmatrix} \quad (8.30)$$

Шум, грешка управљања

$$\bar{\mu}_t = g(u_t, \mu_{t-1}) \quad (8.31)$$

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T \quad (8.32)$$

Јакобијан функције g у односу на положај робота дефинисан је једначином (8.33).

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial z'}{\partial \mu_{t-1,x}} & \frac{\partial z'}{\partial \mu_{t-1,y}} & \frac{\partial z'}{\partial \mu_{t-1,\theta}} \end{pmatrix} \quad (8.33)$$

При чему се парцијални изводи једначина кретања, према моделу кретања на основу пређеног пута, по вектору стања одређују на следећи начин:

$$\frac{\partial x'}{\partial x} = 1; \quad \frac{\partial x'}{\partial y} = 0; \quad \frac{\partial x'}{\partial \theta} = \frac{\Delta s_d + \Delta s_l}{2} \cdot \left(-\sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \cdot 1 \right); \quad (8.34)$$

$$\frac{\partial y'}{\partial x} = 0; \quad \frac{\partial y'}{\partial y} = 1; \quad \frac{\partial y'}{\partial \theta} = \frac{\Delta s_d + \Delta s_l}{2} \cdot \cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \cdot 1; \quad (8.35)$$

$$\frac{\partial \theta'}{\partial x} = 0; \quad \frac{\partial \theta'}{\partial y} = 0; \quad \frac{\partial \theta'}{\partial \theta} = 1; \quad (8.36)$$

Јакобијан функције g у односу на управљање робота се дефинише преко следећих парцијалних извода, што је представљено једначинама (8.37) – (8.41):

$$\frac{\partial x'}{\partial \Delta s_d} = \frac{1}{2} \left(\cos \left(\theta + \frac{\Delta s_d - \Delta s_l}{2b} \right) + \frac{\Delta s_d + \Delta s_l}{2} \right) \cdot \left(-\sin \left(\theta + \frac{\Delta s_d - \Delta s_l}{2b} \right) \cdot \frac{1}{2b} \right) \quad (8.37)$$

$$\frac{\partial x'}{\partial \Delta s_l} = \frac{1}{2} \left(\cos \left(\theta + \frac{\Delta s_d - \Delta s_l}{2b} \right) + \frac{\Delta s_d + \Delta s_l}{2} \right) \cdot \left(-\sin \left(\theta + \frac{\Delta s_d - \Delta s_l}{2b} \right) \cdot \left(-\frac{1}{2b} \right) \right) \quad (8.38)$$

$$\frac{\partial y'}{\partial \Delta s_d} = \frac{1}{2} \left(\sin \left(\theta + \frac{\Delta s_d - \Delta s_l}{2b} \right) + \frac{\Delta s_d + \Delta s_l}{2} \right) \cdot \left(\cos \left(\theta + \frac{\Delta s_d - \Delta s_l}{2b} \right) \cdot \frac{1}{2b} \right) \quad (8.39)$$

$$\frac{\partial y'}{\partial \Delta s_l} = \frac{1}{2} \left(\sin \left(\theta + \frac{\Delta s_d - \Delta s_l}{2b} \right) + \frac{\Delta s_d + \Delta s_l}{2} \right) \cdot \left(\cos \left(\theta + \frac{\Delta s_d - \Delta s_l}{2b} \right) \cdot \left(-\frac{1}{2b} \right) \right) \quad (8.40)$$

$$\frac{\partial \theta'}{\partial \Delta s_d} = \frac{1}{2b}; \quad \frac{\partial \theta'}{\partial \Delta s_l} = -\frac{1}{2b}; \quad (8.41)$$

Локализација применом линеаризованог Калмановог филтера - корак корекције:

$$\hat{z}_t = \begin{pmatrix} \sqrt{(m_x - \bar{\mu}_{t,x})^2 + (m_y - \bar{\mu}_{t,y})^2} \\ a \tan 2(m_y - \bar{\mu}_{t,y}, m_x - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{pmatrix} \quad (8.42)$$

Предвиђена очекивана вредност мерења

$$H_t = \frac{\partial h(\bar{\mu}_t, m)}{\partial x_t} = \begin{pmatrix} \frac{\partial r_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,z}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,z}} \\ \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,y}} & & \end{pmatrix} \quad (8.43)$$

Јакобијан функције h у односу на положај робота

$$Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{pmatrix} \quad (8.44)$$

$$S_t = H_t \bar{\Sigma}_t H_t^T + Q_t \quad (8.45)$$

Предвиђена матрица варијанси мерења

$$K_t = \bar{\Sigma}_t H_t^T S_t^{-1} \quad (8.46)$$

Калманово појачање

$$\mu_t = \bar{\mu}_t + K_t (y_t - y_t) \quad (8.47)$$

Нова очекивана вредност

$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \quad (8.48)$$

Нова матрица коваријанси

Једначинама (8.49) и (8.50) представљају се координате положаја светлосног сензора у глобалном координатном систему

$$x_s = x_r + p \cdot \cos(\theta_r) \quad (8.49)$$

$$y_s = y_r + p \cdot \sin(\theta_r) \quad (8.50)$$

При чему p представља растојање између сензора и тежишта робота.

Јакобијан функције h у доносу на положај робота добија се парцијалним изводима једначина (8.49) и (8.50) по вектору стања.

$$\frac{\partial x_s}{\partial x} = \frac{\partial}{\partial x} (x_r + p \cdot \cos(\theta_r)) = 1 \quad (8.51)$$

$$\frac{\partial x_s}{\partial y_r} = \frac{\partial}{\partial y_r} (x_r + p \cdot \cos(\theta_r)) = 0 \quad (8.52)$$

$$\frac{\partial x_s}{\partial \theta_r} = \frac{\partial}{\partial \theta_r} (x_r + p \cdot \cos(\theta_r)) = -p \cdot \sin(\theta_r) \quad (8.53)$$

$$\frac{\partial y_s}{\partial x_r} = \frac{\partial}{\partial x_r} (y_r + p \cdot \sin(\theta_r)) = 0 \quad (8.54)$$

$$\frac{\partial y_s}{\partial y_r} = \frac{\partial}{\partial y_r} (y_r + p \cdot \sin(\theta_r)) = 1 \quad (8.55)$$

$$\frac{\partial y_s}{\partial \theta_r} = \frac{\partial}{\partial \theta_r} (y_r + p \cdot \sin(\theta_r)) = p \cdot \cos(\theta_r) \quad (8.56)$$

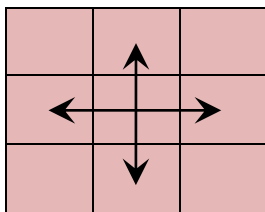
На послетку, матрица Јакобијана функције h има следећи облик:

$$H_t = \begin{pmatrix} 1 & 0 & -p \cdot \sin(\theta_r) \\ 0 & 1 & p \cdot \cos(\theta_r) \end{pmatrix} \quad (8.57)$$

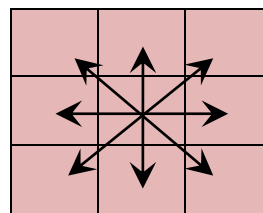
9. A* алгоритам

A* алгоритам претраживања је алгоритам који је у широкој примени за одређивање најкраће путање између почетне и циљне тачке. Познат по својим перформансама, тачношћу и флексибилношћу, веома је распрострањен и има широку примену. Настао је као наставак Дајкстра алгоритма из 1959 год. али, уз коришћење хеуристике, постиже боље резултате од свог предходника по питању времена претраживања. Карактерише га претраживање великих простора за кратко време. A* користи „BestFirstSearch“ алгоритам, помоћу којег проналази тачку која има најмању задату вредност (оцену). Помоћу хеуристике одређује растојање од почетне до сваке тачке у радном простору и бира пиксел са најмањом вредношћу. Одабрани пиксел усваја као следећи који треба остварити.

За рад алгоритма неопходно је предходно дефинисати окружење. Реално окружење је потребно дискретизовати, као 2D окружење, и представити пикселима произвољног облика. Најчешће коришћени облици пиксела су квадратног облика, мада могу бити и други попут тругла, правоугаоника, шестоугла и др. Од величине пиксела зависи време и прецизност претраживања. Време за пиксел реда величине 1cm бити краће од времена претраживања за пиксел реда величине 1mm, али ће зато прецизност у првом случају бити мања. Како пиксели могу бити четвороспојиви или осмоспојиви [2], кретање од једног до другог пиксела може бити четворосмерно или осмосмерно. Четворосмерно кретање значи да је могуће кретање само по хоризонтали и вертикали, док осмосмерно кретање дозвољава и кретања по дијагоналама.



Слика 9.1: Четворосмерно кретање



Слика 9.2: Осмосмерно кретање

Као и сви претраживачки алгоритми, A* алгоритам претражује могуће путање ка циљу. За разлику од „BestFirstSearch“ алгоритма, он узима у обзир укупан пређени пут током претраживања.

Свако поље у матрици хеуристике окружења садржи вредност оцене. Ове вредности се означавају са $h(i,j)$, где су i,j координате поља окружења у дискретизованој матрици и представљају растојање до циљног поља. Вредност растојања од тренутног до пиксела у окружењу означавамо са $g(i,j)$, где су i,j координате поља у коме се тренутно налази претраживач, узимајући у обзир и препреке. Пређени пут се рачуна од стартног поља до тренутног, сабирањем вредности $h(i,j)$ и $g(i,j)$.

На слици 9.3 приказан је начин додељивања вредности за $g(i,j)$ за сваки пиксел од пиксела у окружењу, применом правила четворосмерног кретања. Зеленом бојом означено је стартно поље.

	g=1	
g=1	g=0	g=1
	g=1	

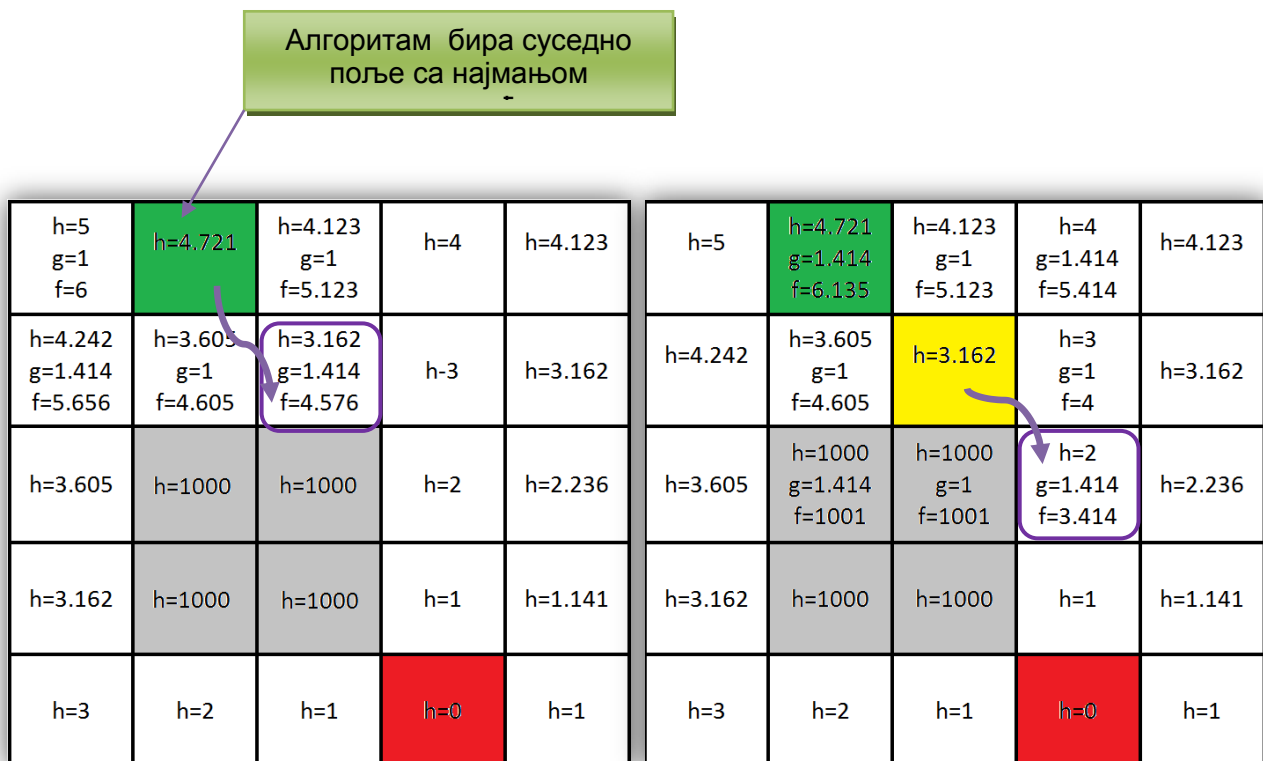
Слика 9.3: Пример додавања вредности четворосмерног кретања

На слици 9.4 приказан је начин додељивања вредности за $g(i,j)$ за сваки пиксел од пиксела у окружењу, применом правила осмосмерног кретања. Зеленом бојом означено је стартно поље.

g=1.41	g=1	g=1.41
g=1	g=0	g=1
g=1.41	g=1	g=1.41

Слика 9.4: Пример додавања вредности осмосмерног кретања

На сликама 9.5 - 9.6 представљен је принцип функционисања алгоритма по Еуклидској норми са додавањем вредности пређеног пута. За све чворове око тренутног у коме се налази претраживач, на вредности добијене хеурстиком (h) додају се вредности кретања (g). За овај пример, вредности кретања у хоризонталном и вертикалном правцу износе 1, а по дијагонали 1.4. Када се израчунају све вредности оцена за чворове око тренутног ($f=h+g$), алгоритам тражи ону која је најмања и то поље проглашава за његово следеће поље. Овај процес се понавља док год се не стигне до циља, тј. чвора чија је вредност $h=0$.

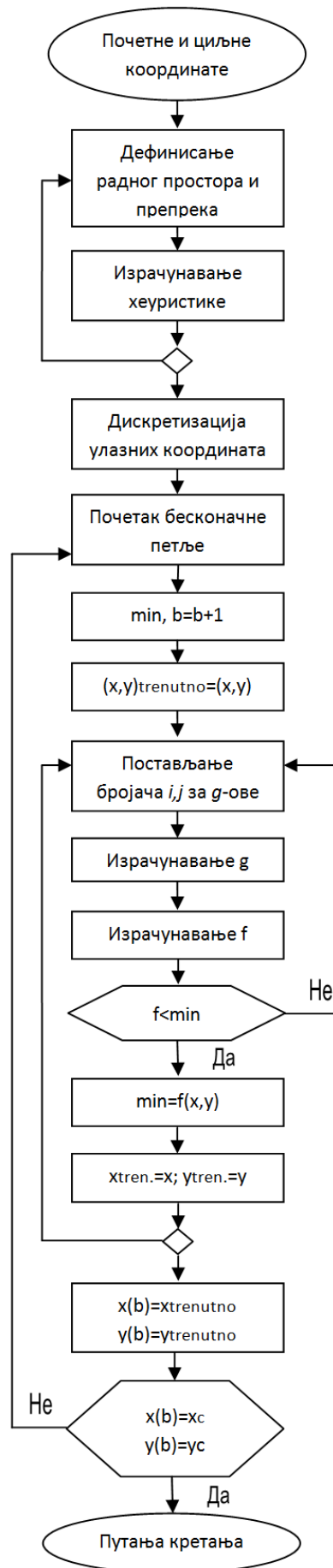


Слика 9.5: Принципа функционисања A* алгоритма, први и други корак

h=5	h=4.721	h=4.123	h=4	h=4.123	h=5	h=4.721	h=4.123	h=4	h=4.123
h=4.242	h=3.605	h=3.162 g=1.414 f=4.576	h=3 g=1 f=4	h=3.162 g=1.414 f=4.576	h=4.242	h=3.605	h=3.162	h=3	h=3.162
h=3.605	h=1000	h=1000 g=1 f=1001	h=2	h=2.236 g=1 f=3.236	h=3.605	h=1000	h=1000 g=1.414 f=1001	h=2 g=1 f=3	h=2.236 g=1.414 f=3.65
h=3.162	h=1000	h=1000 g=1.414 f=1001	h=1 g=1 f=2	h=1.141 g=1.414 f=2.828	h=3.162	h=1000	h=1000 g=1 f=1001	h=1	h=1.141 g=1 f=2.141
h=3	h=2	h=1	h=0	h=1	h=3	h=2	h=1 g=1.414 f=2.414	h=0 g=1 f=1	h=1 g=1.414 f=2.414

Слика 9.6: Принцип функционисања A* алгоритма, трећи и четврти корак

На дијаграму који следи, приказан је алгоритам на основу кога је писан програм у MatLab [9] програмском пакету за имплементацију A* претраживања.

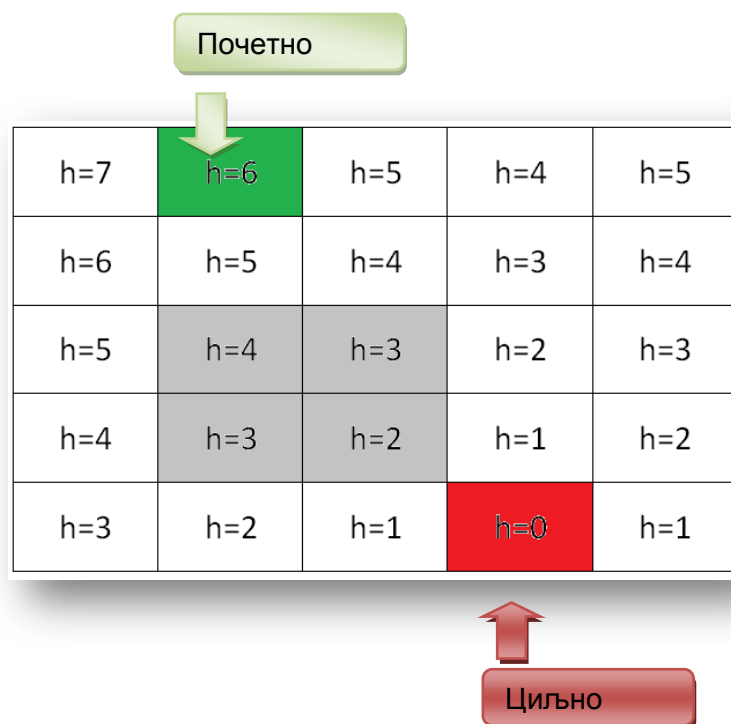


Дијаграм програмирања A* алгоритма

A* алгоритам је имплементиран у главни програм који управља кретањем мобилног робота. Улазни подаци који се користе у програму позивају се из главног програма и заправо су почетне и циљне тачке. За функционисање алгоритма такође је потребно дефинисати хеуристику.

Хеуристика разликује две норме додељивања оцена и то су Еуклидска и Менхетн норма. Код Менхетн норме, растојање од циљног до почетног поља рачуна се тако што се за кретање по хоризонталним и вертикалним правцима додаје по једна фиксна вредност, обично 1, док се кретања по дијагоналним правцима не узимају у обзир. Код Еуклидске норме, ове вредности се додају мало другачије. За свако поље прерачунава се даљина од циљног поља, коришћењем тригонометрије.

На слици 9.7 је представљено окружење са додељеним вредностима по Менхетн норми. Окружење је величине 5x5, а величина пиксела 1, Зелено поље представља почетно, а црвено, циљно поље. Вредности свих поља су увећана за по 1 по хоризонталном и вертикалном правцу од циљног ка почетном пољу.



Слика 9.7: Пример додељивања оцена Менхетн норме

На слици 9.8 је представљено окружење са додељеним вредностима по Еуклидској норми. Окружење је величине 5x5, а величина пиксела 1, зелено поље представља почетно, а црвено, циљно поље. Вредност хеуристике за свако поље дато је растојањем између тог и циљног поља, израчунато преко тригонометрије $h = \sqrt{x^2 + y^2}$, где су x и y растојања тренутног од циљног поља по координатама x и y .

h=5	h=4,721	h=4,123	h=4	h=4,123
h=4,242	h=3,605	h=3,162	h=3	h=3,162
h=3,605	h=2,828	h=2,236	h=2	h=2,236
h=3,162	h=2,236	h=1,141	h=1	h=1,414
h=3	h=2	h=1	h=0	h=1

Слика 9.8: Пример додељивања оцена Еуклидске норме

Ради смањења обимности и лакшег прегледа програмског кода, рачунање хеуристике се врши у подпрограму под називом „Definisanje_H“. На почетку програма позива се матрица дискретизованог окружења „rp“ из подпрограма „Okruzenje“. Затим се из те матрице узимају мере дискретизованог окружења, тј. димензије позване матрице. У следећем кораку прави се матрица хеуристике чији је тип исти као и димензије матрице окружења. Чланови матрице представљају чворове дискретизованог окружења. Уз помоћ два бројача сваком члану матрице додељује се вредност удаљености циљног чвора од оног у који се уписује оцена хеуристике. Удаљеност се рачуна применом Еуклидске форме, користећи формулу $h = \sqrt{x^2 + y^2}$, где су x и y димензије чланова у матрици.

Матрица окружења „rp“ је добијена тако што је реално окружење, у коме се мобилни робот креће, измерено и дискретизовано пикселима. Вредности матрице су 0 и 1000, где су нуле додељиване за чланове који представљају чвор кроз који робот може да прође, а вредности 1000 су додељиване члановима који представљају чворове кроз које робот не сме да прође, тј. препреке.

Израз из подпрограма „Definisanje_H“ је матрица „rp1“. Ова матрица се добија сабирањем матрице окружења „rp“, која садржи податке о препрекама и хеуристике која је претходно израчуната.

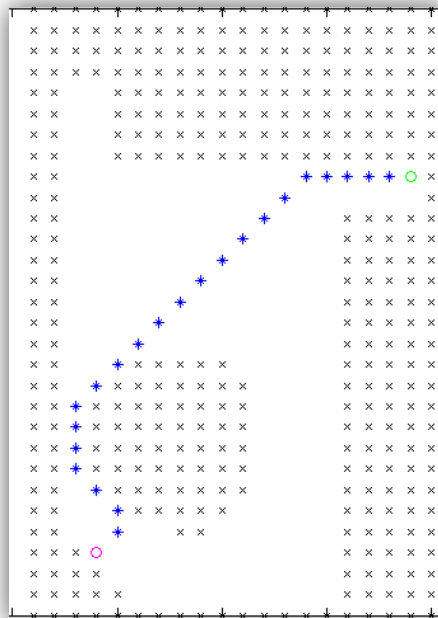
Проналажење најкраће путање од почетне до циљне тачке реализује се употребом бесконачне петље. Она има за циљ да за тренутни чвор у којем се робот налази испитује вредности коначне матрице „rp1“ у његовом непосредном окружењу и налази пиксел са минималном вредношћу који постаје нова позиција робота. На слици 9.9 приказано је непосредно окружење претраживача који се налази у чвору означеним жутом бојом, док је црвеном бојом означено поље које је изабрано за следеће, које треба посетити.

h=1000 g=1.414 f=1001	h=2 g=1 f=3	h=2.236 g=1.414 f=3.65
h=1000 g=1 f=1001	h=1	h=1.141 g=1 f=2.141
h=1 g=1.414 f=2.414	h=0 g=1 f=1	h=1 g=1.414 f=2.414

Слика 9.9: Непосредно окружење претраживача

Током процеса реализације програма у MatLab [9] окружењу за прелажење из датог положаја у наредну позицију неопходно је дефинисати вредност минимума. Она мора бити већа од вредности пиксела коначне матрице за поља у која мобили робот може доћи, а мања од вредности препрека у истој матрици. У нашем случају овај минимум има вредност 100. После дефинисања минимума, отварају се две кружне петље којима је задатак да за тачку у којој се претраживач тренутно налази, испитају све околне тачке. Упоређујући вредности чворова са минималном вредношћу, онај који има најмању вредност записује се као следећи пиксел у који треба отићи. Овај поступак понавља се све док тренутна позиција не достигне циљну позицију. Тада се прекида бесконачна петља и добијени резултати се шаљу назад у програм који управља кретањем.

Визуелизација целокупног поступка остварена је графичким приказом радног окружења коме су придружени сви чворови које мобилни робот треба да оствари да би успешно извршио задатак. На слици 9.10 је приказ екрана где се виде сви пиксели које робот може да достигне, пиксели који представљају препреке и путања која је излаз из A*. Сивом бојом означени су чворови у које робот не може да дође. Плавом бојом приказани су пиксели који су излаз из A*. Празна поља су чворови којима мобилни робот може да се креће.



Слика 9.10: Приказ дискретизованог радног окружења

Коришћени алгоритам задовољава потребе за радно окружење у коме ће се интелигентни робот користити. Када би проблем био сложенији, извршила би се модификација тако да се кретање планира пар корака унапред. На основу тога била би остварена могућност одабира најкраће путање од могућих. Овај поступак би рачунски мало дуже трајао од постојећег алгоритма, али с обзиром на то да су процесори који обрађују такве податке толико брзи време које је потребно за изачунавање дела путање много је мање од времена за које робот стигне од једне до друге задате тачке из програма.

10. Симулациони модел у AnyLogic окружењу

Симулациони модел представља појам који је у уској вези са управљањем токовима материјала. Основа за то лежи у циљу остварења што бољих производних резултата. Постепено се од усмених и писаних наредби које су се јављале ради побољшања производности дошло до софистицираних алата за управљање производњом чиме су знатно порасле могућности за прибављање, обраду и пренос података потребних за адекватну организацију рада система. Често се нема јасна свест колико се неко предузеће може сматрати неефикасним и неуспешним. Разлог за то лежи у напору да се софистицира производња компаније, а најчешће се заборавља на обезбеђивање ефикасног функционисања основних производних процеса, што је предуслов за примену савремених, рачунарских организационих средстава. Стога је значај примене симулационих модела пре реализације нових идеја или побољшања старог решења вишеструки.

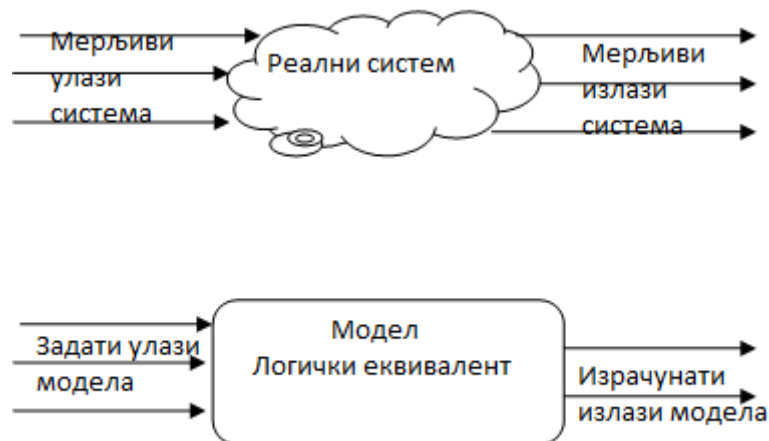
Према [1] симулација и модел могу се дефинисати на следеће начине:

Симулација је имитација рада реалног процеса или система током времена.

Модел представља концептуални оквир који описује систем.

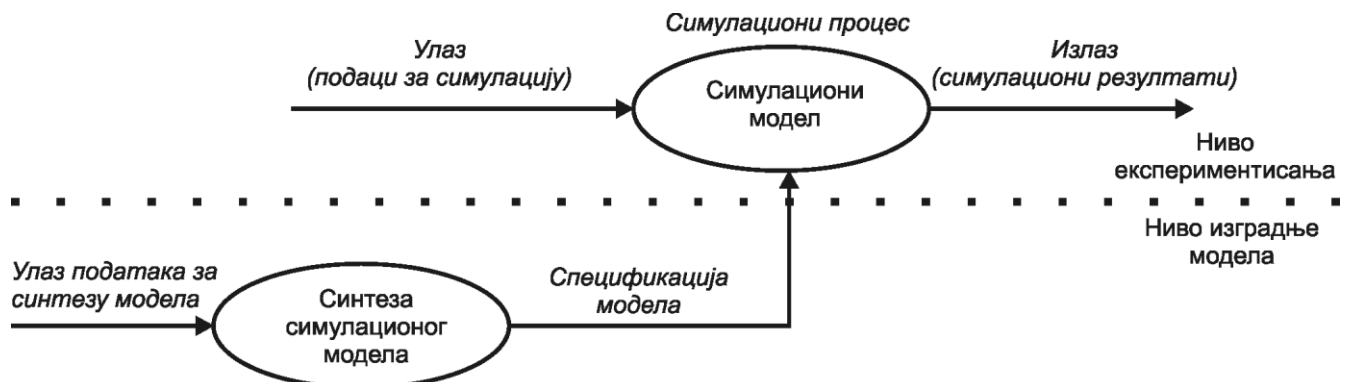
Праћењем понашања симулације генерисаног модела, такозваном посматрању вештачке историје, изводе се закључци о радним карактеристикама система. Моделирање представља један од основних процеса људског ума. Оно је уско везано за начин људског размишљања и решавања проблема. Као резултат процеса који се назива интелигентно људско понашање, моделирање представља свакодневну активност и велики део онога што нас чини људским, интелигентним бићима. Моделирање изражава способност мишљења и замишљања, коришћења симбола и језика, комуницирања, вршења генерализације на основу искуства, суочавања са неочекиваним. Оно омогућава уочавање образаца, процењивање и предвиђање, управљање процесима и објектима, излагање значења и сврхе. Ипак, мора се водити рачуна о примени адекватних модела јер неадекватни модели смањују ефикасност, а са друге стране могу имати катастрофалне последице у технолошком, социјалном или историјском погледу. Генерално гледано коришћење модела је у великој мери исплативо јер се без формирања реалног система долази до одређеног сазнања. Он омогућава суочавање са реалним светом (системом) на поједностављен начин, избегавајући његову комплексност и иреверзибилност, као и све опасности које могу проистећи из експеримента над самим реалним системом. Другим речима, модел је опис реалног система са свим оним карактеристикама које су релевантне за жељени угао посматрања. То заправо значи да се у процесу моделирања мора извршити избор између оних елемената и карактеристика система које су од значаја за истраживање и које ће бити обухваћене моделом и преосталих за ирелевантних, које модел неће садржати. Стога се и каже да модел представља упрошћену слику реалног система. Његов је циљ да уобличи на видљив, често формалан начин, оно што је суштинско за разумевање неког аспекта његове структуре или понашања. Ниво апстракције у процесу моделирања утиче на валидност модела, односно на успешност представљања реалног система моделом. Проблем валидације модела јавља се у сваком процесу моделирања, а проистиче из чињенице да је модел увек поједностављени поглед на реални систем који је предмет посматрања. Сувише сложени или савршени модели који имају способност да за исти скуп улазних величина производе исте излазне вредности као и реални системи, чак иако су оствариви, по правилу су прескупи и неадекватни за експериментисање. С друге стране, сувише поједностављени модели не одсликавају на прави начин посматрани систем, а резултати који се добијају њиховом применом могу да буду неадекватни и погрешни. Стога, опредељујући се за ниво апстракције у посматрању реалног система, потребно је у одређеном тренутку повући границу у реалном систему и то тако да резултујући модел што верније одсликава посматрани систем, али и да, с друге стране, његова сложеност и цена не буду

ограничавајући фактори. Однос реалног система и модела може се представити следећом сликом:



Слика 10. 1: Упоредни приказ реалног система и модела

Модел је најчешће у облику скупа претпоставки изражених математички, логички или кроз скуп симболичких релација између ентитета. Концепт симулације према [1] обухвата низ активности моделирања, експериментисања са моделом и анализа симулационих резултата.



Слика 10. 2: Активности у оквиру процеса симулације [1]

Према горе поменутом извору видимо да се симулација може применити у великом броју области:

- Производња,
- Управљање пројектима,
- Војна примена,
- Логистика, ланци снабдевања, дистрибуција,
- Транспортни модели и саобраћај,
- Симулација пословних процеса,
- Здравство,
- Симулација рада компјутера,
- Симулација компјутерских мрежа и
- Симулација аутоматизованих система за руковање материјалом.

Савремено моделирање незамисливо је без рачунара. У моделирању рачунари се користе у две сврхе: у развоју модела и у извођењу прорачуна на основу створеног модела.

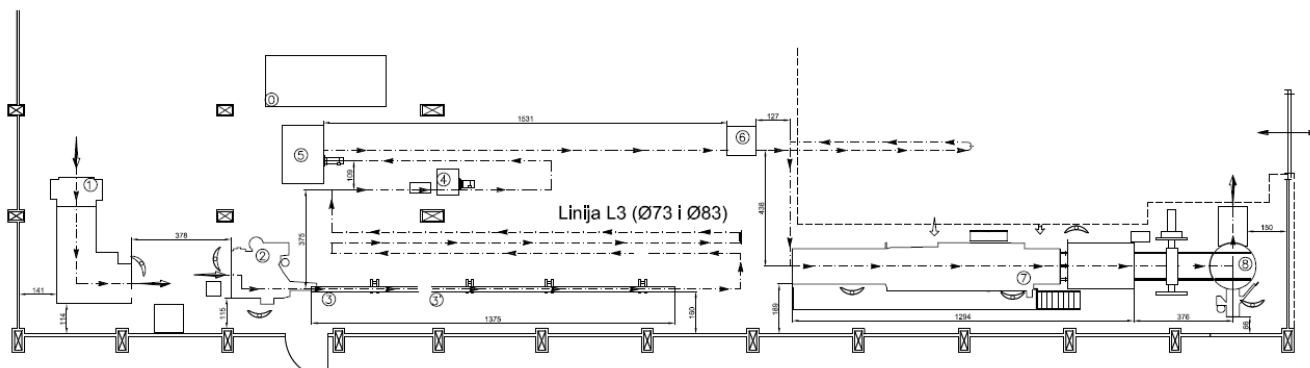
На тај начин, моделирање помоћу рачунара постаје дисциплина којом се могу адекватно и ефикасно приказивати сложени системи и обликовати и испитивати њихово понашање. Тема нашег пројекта управо је моделирање помоћу модела који ће детаљније бити описано у оквиру следећих подпоглавља.

10.1. Развој симулационог модела пројектног задатка

Тема другог дела пројекта такође је везана за реалне производне проблеме. Реч је о производној линији број три **Фабрике Металних Производа** [10]. Ова линија врши производњу лименки $\varnothing 73$ и $\varnothing 83$. Списак и опис машина у производној линији:

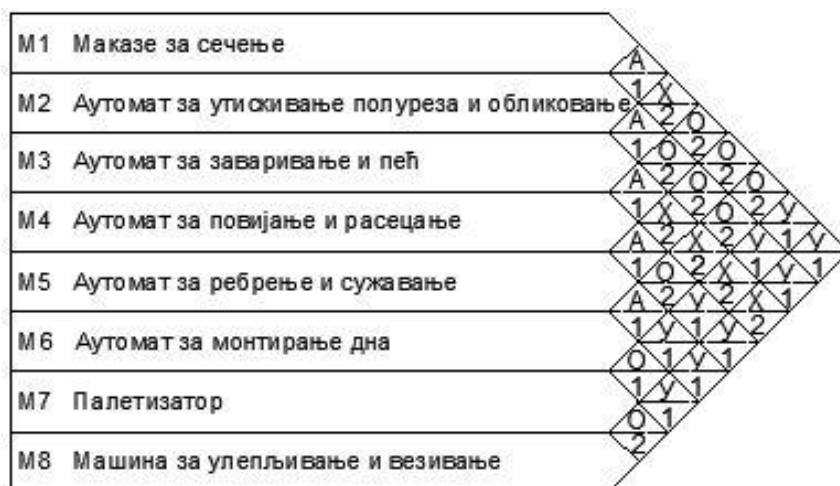
- M1 Маказе за сечење
- M2 Аутомат за утискивање полуреца и обликовање
- M3 Аутомат за заваривање и пећ
- M4 Аутомат за повијање и расецање
- M5 Аутомат за ребрење и сужавање
- M6 Аутомат за монтирање дна
- M7 Палетизатор
- M8 Машина за увезивање и улепљивање

На слици 10.3 можемо видети приказ диспозиционог плана линије за коју је извршена симулација.



Слика 10. 3: Диспозициони план линије 3

Квалитативни однос између машина могу се представити помоћу троугаоне матрице. Горњи део поља у матрици се користи за уношење јачине појединих веза, а доњи део за разлог за одређивање дате зависности [4]. Квалитативна матрица међузависности активности са јачинама веза и разлозима приказана је на слици 10.4.



Озн	Степен зависности
A	Апсолутно неопходно
E	Веома важно
I	Важно
O	Потребно
U	Неважно
X	Непожељно
XX	Веома непожељно

Озн	Разлог
1	Ток материјала
2	Тип машине
3	Одржавање обрадног система
4	Еколошки

Слика 10. 4: Квалитативна матрица међузависности активности

После овог прегледа потребно је табеларно извршити приказ времена обраде по операцијама технолошког поступка производње лименки. Овај приказ налази се у табели у наставку.

Табела 11: Приказ времена обраде по технолошким поступцима

Редни број	Технолошки поступак	Машина	Време трајања поступка [s]
1	Транспорт палете са лимовима од улаза у халу до маказа и спуштање	Виљушкар - бензинац	25
2	Постављање палете на машину	Виљушкар - електрични	179
3	Отпакивање палете	Ручна манипулација	25
4	Подизање палете до вакуумских сисалки	Аутомат	25
5	Сечење припремка из табле лима	Маказе	9
6	Скидање трака са маказа и постављање на сто	Ручна манипулација	20
7	Преношење трака и пуњење шаржера	Ручна манипулација	3
8	Утискивање полуреца и савијање	Аутомат	2
9	Заваривање и печење споја	Аутомат и пећ	20
10	Транспорт и хлађење	Магнетни транспортер	111
11	Повијање и расечање	Аутомат	2
12	Транспорт	Магнетни транспортер	10
13	Ребрење и сужавање	Аутомат	4
14	Транспорт	Магнетни транспортер	21
15	Монтирање дна	Аутомат	2.5
16	Транспорт	Магнетни транспортер	17
17	Палетизација	Палетизатор	64
18	Увезивање и облепљивање	Машина	269
19	Одвожење палете са лименкама до врата погона	Виљушкар - бензинац	27

Пре почетка моделирања мора се уочити да се део технолошког поступка са циклусним временима односи на палете са таблама лима, део на табле лима, део на једну лименку и део на целокупну палету лименки. Наиме, време трајања транспорта палете са лимовима од улаза у халу до маказа и спуштање, постављања палете на машину, отпакивања палете и подизања палете до вакуумских сисалки односи се на целокупну палету лимова. Време сечења припремка из табле лима је време потребно да се исече цела табла, док се циклусна времена за скидање трака са маказа и постављање на сто, преношење трака и пуњење шаржера, утискивање полуреца и савијање, заваривање и печење споја, сви видови транспорта магнетним транспортерима, транспорт и хлађење, повијање и расечање, ребрење и сужавање и монтирање дна односе на једну лименку. С друге стране проблем постаје још сложенији када се уочи да се палетизација, увезивање и облепљивање, заједно са одвожењем одосе на једну целу палету лименки. Поставља се

питање на који начин формирати модел и како одредити циклусна времена. Да ли за једну таблу или за лименку? С обзиром на то да би формирање модела у односу на једну таблу лима умногоме закомпликовало сам процес моделирања јер се ипак већина машина води циклусним временима за једну лименку, као и уз чињеницу да је за ово решење потребно познавање Java програмирања одлучено је да наш модел формирамо на основу једне лименке. Идејна замисао је била да се део технолошког процеса који се односи на транспорт палете лима, њено постављање на машину и отпакивање обухвати једним ентитетом - извором.

Поставило се питање колико се лима, изражено у лименкама, доставља првој машини у једној секунди. Да би се добио одговор на то питање пошло се од низа претпоставки. Број табли лима у једној палети заокружен је на 50. Табле које су коришћене биле су стандардне 1000x2000 mm и дебљине 0.5 mm. Ако просечни пречник лименке на линији 3 износи 78 mm добијено је да је уз висину лименке од 150 mm за једну лименку потребно 244x150 mm лима. Када се ове димензије пренесу на једну таблу закључује се да се из ње може добити 48 лименки. Тај број се затим множи са бројем табли у палети и дели са циклусним временима за прве три операције да би се добило колико лименки стиже на машину у једној секунди. Реч је о 9.841 лименци. Овај број се и користи у симулацији за конфигуравање извора.

Следећи корак био је почетак стварања симулације. При овом поступку уочене су следеће компоненте система према [1]:

- Ентитет

Представља објекат од значаја за систем. Може бити привремени и стални. Привремени ентитети обухватају трансакције и то су у нашем случају делови, док су стални средства (енг. facilities) која чине машине у фабрици.

- Атрибут

Говори о својству ентитета: брзина, капацитет.

- Активност

Временски период одговарајуће дужине: просецање.

- Стање

Скуп променљивих које описују систем у било ком тренутку: стање машине (заузета, слободна, у отказу...)

- Догађај

Тренутна појава која може да промени стање система: отказ машине.

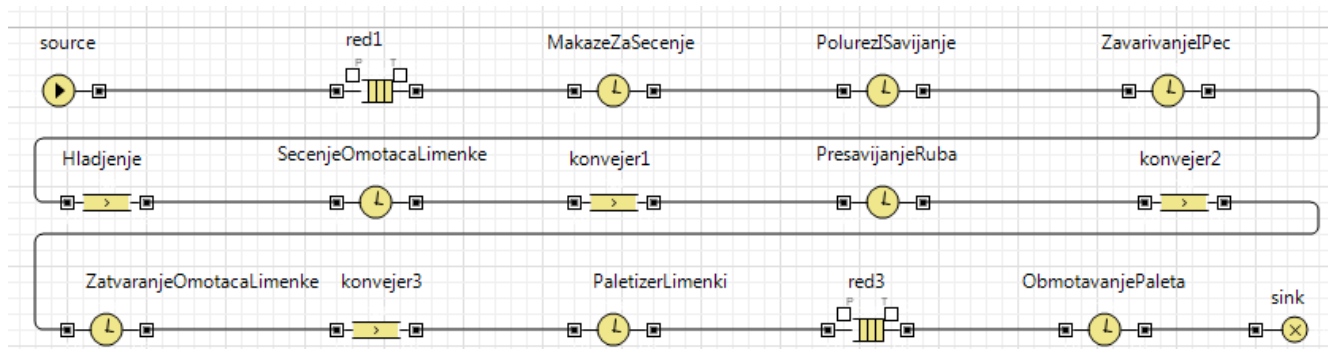
- Редови

Пасивна стања ентитета – чекање на стицање услова за започињање активности.

Пројектовање симулације у AnyLogic софтверском пакету остварено је коришћењем следећих објеката: улаз (енг. source), кашњење (енг. delay), конвејер (енг. conveyor), ред (енг. queue) и излаз (енг. sink). Објекти су међусобно повезани помоћу конектора (енг. connector).

Привремени ентитети у моделу су лимеке и они се дефинишу са спецификацијом карактеристика извора. Стални ентити представљени су свим машинама које се налазе у погону. Својства ентита дефинисана су у сваком објекту на свој начин према технолошком поступку. Редови се односе на пасивна стања привремених ентитета која доводе до чекања да се подвргну одговарајућем поступку.

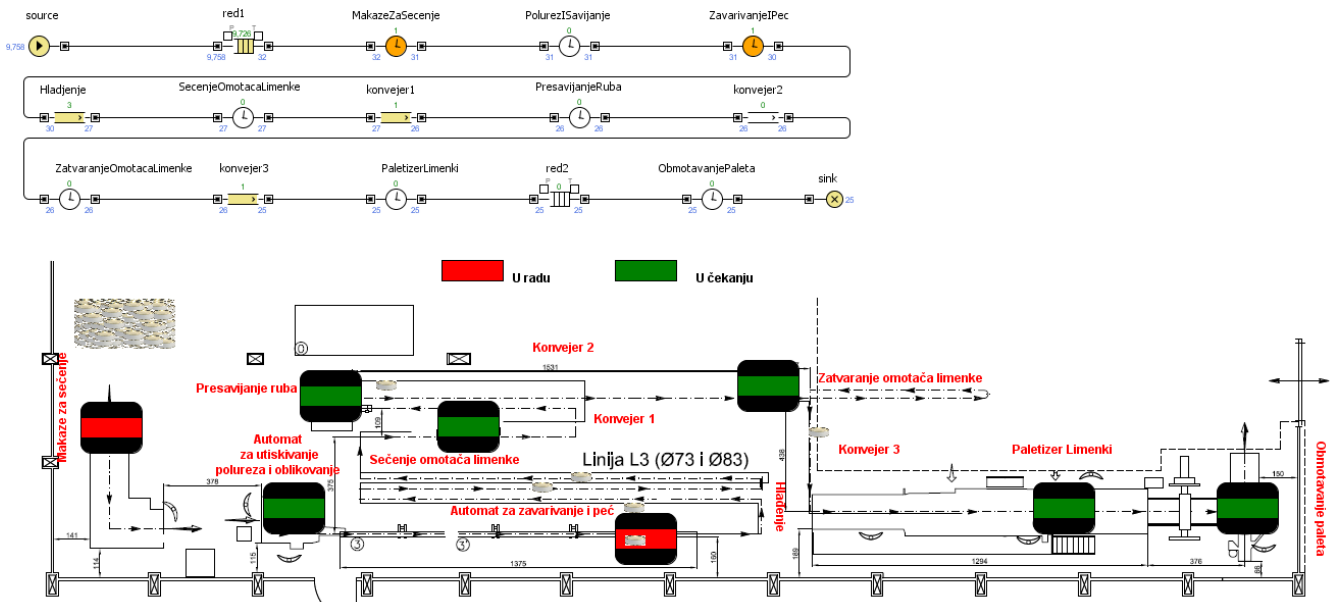
Након извршених почетних анализа извршена је синтеза симулације постојеће производње. Симулација је приказана на слици 10.5, а у наставку ће се приказати њена даља анализа.



Слика 10. 5: Симулациони модел рада система постојеће производње

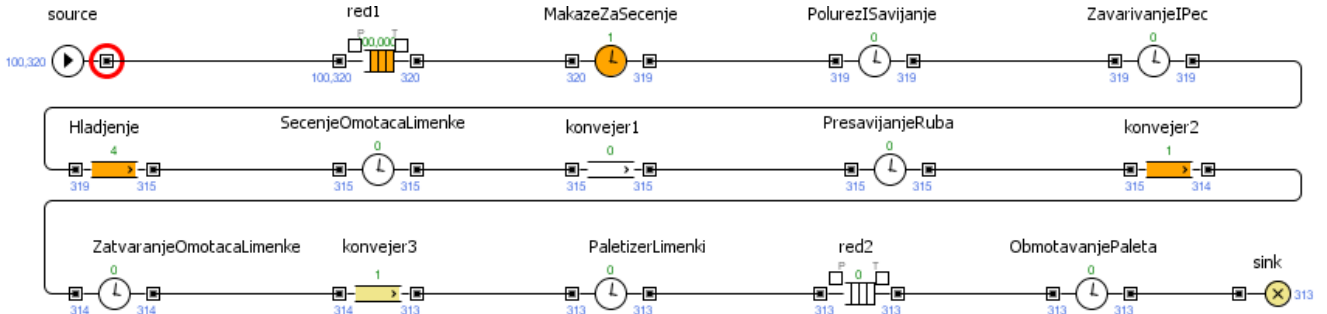
Ред1 датог модела мора се формирати испред маказа за сечење јер са палетом стиже велики број лименки, односно материјала за њену производњу. Капацитет овог објекта ограничен је на 100 000 лименки, што и не представља велико складиште с обзиром на то да се у реалном погону оне налазе у облику табли чија укупна дебљина не прелази 2 m. Маказе за сечење су представљене преко објекта delay код кога се у оквиру особина дефинише време кашњења. Оно је еквивалент нашем циклусном времену за ову машину. Потребно је уочити да је циклусно време увећано за период потребан за скидање трака, њихово преношење и пуњење шаржера. Сличан поступак примењен је и код других машина где је то било потребно. Тако је код палетизера и код обмотавања палета дефинисано време просечно за једну лименку. За палетизер претпоставља се да се ради о 100 лименки, док се обмотавање врши за 10 пута више лименки.

Ради боље прегледности експеримента формирана је и одговарајућа анимација која на илустративан начин приказује функционисање производног погона. За сваку машину коришћен је поједностављени приказ који индуковањем црвеног или зеленог стања говори о заузетости, одосно незаузетости машине респективно. Такође, да би се у потпуности схватили токови материјала читава анимација састављена је са layout-ом окружења. Конвејери су се представили онако као што су дати и на цртежу те је целокупна анимација заокружена.



Слика 10. 6: Симулација и анимација рада система

На почетку анализе циљ је био сазнати какве су перформансе система. Потребно је било видети када се и где јавља уско грло производње, да би се затим предложило ново решење. На слици 10.7 уочава се да застој настаје после 10240 секунди, односно 2.8 часова. Разлог је попуњавање складишта код прве машине (Ред1). У том тренутку на стању се налази 313 израђених лименки.

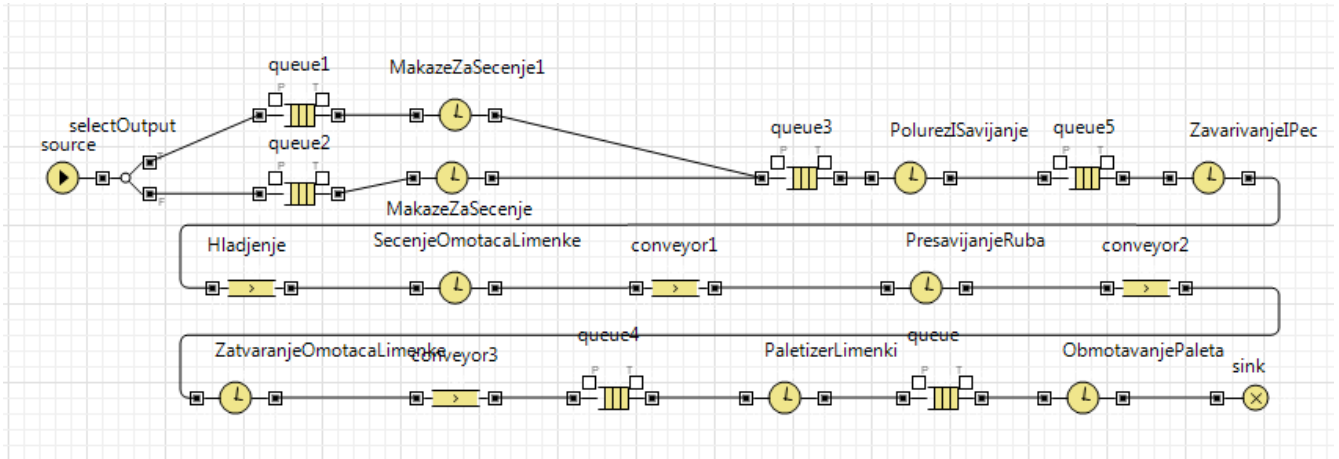


Слика 10. 7: Застој система

Разлог за застој у раду лежи у великом циклусном времену прве машине у односу на остале. Као закључак може се извести да се време појаве застоја одлаже повећањем складишта које у овом тренутку није велико или неким другим решењем. Један вид решења представљен је као предлог у наредном подпоглављу на слици 10.8.

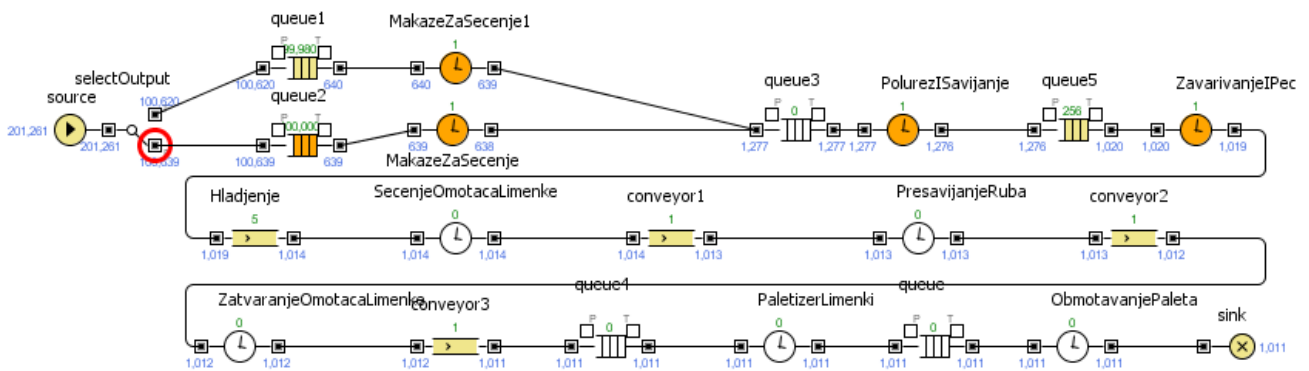
10.2. Формирање нових решења токова материјала

Новим решењем токова материјала жеља је била да се види какви ефекти се постижу у случају да уместо једних маказа за сечење у погону постоје две машине. У овом случају било је потребно формирати додатну машину и додатни ред испред ње који у физичкој реализацији не мора представљати ново складиште већ постојеће проширеног капацитета. Исто тако ради валидности рада у самом софтверском пакету било је неопходно додати редове како би се омогућило неометано функционисање.



Слика 10. 8: Модел предложеног решења

У случају овог система такође ће доћи до застоја, али после дупло већег временског интервала. Слични ефекти на овом систему могли би се постићи и увођењем додатне смене за ту машину тако да не би била потребна новчана улагања за набављање машине. Отказ система и симулација рада приказани су на слици 10.9.



Слика 10. 9: Застој у раду предложеног система

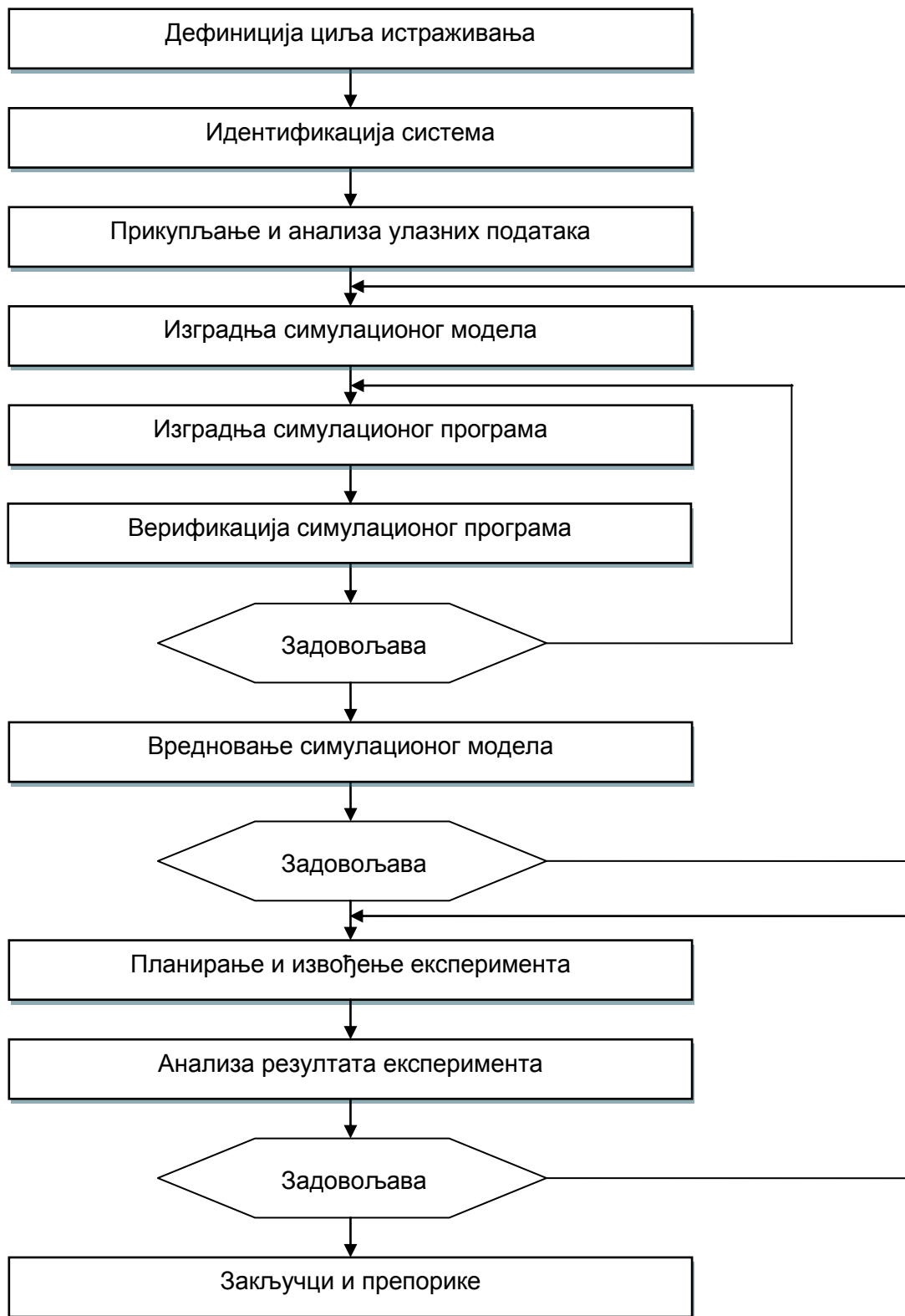
10.3. Дискусија

Анализа токова материјала линије 3 Фабрике Металних Производа [10] показала је да је распоред машина добар и да одговара поступку израде лименки, као и да нема великих губитака у транспорту с обзиром на то да је већина машина повезана међусобно конвејерима. На оним местима на којим то није случај јављају се мали временски губици који се у будућности могу смањити, иако не представљају кочницу у производњи.

Проблематика самог симулирања представљена је кроз дијаграм у наставку овог текста. Видимо да се кроз одговоре на питања о продуктивности предузећа и начинима и могућностима његовог побољшања остварује циљ овог дела пројекта постављен на почетку рада. С обзиром на то да су се улазни параметри већ налазили у поставци проблема мора се нагласити да би рад био комплетан тек када би се пре процеса симулације извршило и прикупљање потребних података.

Потребно је још скренути пажњу на једну чињеницу. Неопходно је запитати се о валидности самог модела. Проблеми који су се јављали на почетку, при одабиру променљивог ентитета, могли би се отклонити са познавањем Java програмирања. Тада би се симулирао комплетан производни процес са трансформацијом материјала од палете лима до палете лименки. На тај начин остварили би се резултати који би се могли поистоветити са реалним. Овако, ипак остаје питање у којем је степену описани модел примерен за нашу проблематику. О овој теми би се при имплементацији приказаног решења детаљно морало продискутовати, јер је добро познато да неадекватан модел лако може довести до извођења погрешних закључака са фаталним последицама.

С друге стране, ако се ово спорно питање одклони овај систем могао би наћи примену у реалној производњи. Добра страна сваке симулације лежи у томе што иако смо тренутно задовољни овим решењем при свакој новој идеји можемо упоредити њену употребљивост све док не нађемо решење које је најадекватније.



Симулациони дијаграм

11. Експериментални резултати

Пошто су размотрена различита решења и изабран начин реализације кретања робота у задатом технолошком окружењу, приступа се експерименталној верификацији поступка. Да би се то успешно урадили, прво се полази од подешавања параметара.

$b[cm]$ - Врши се мерење растојања између тачкова робота и уноси се та вредност у програм (мерење се понавља сваки пут када се поново покреће робот јер вредности могу да одступају)

$r[cm]$ - Уноси се вредност полупречника тачкова

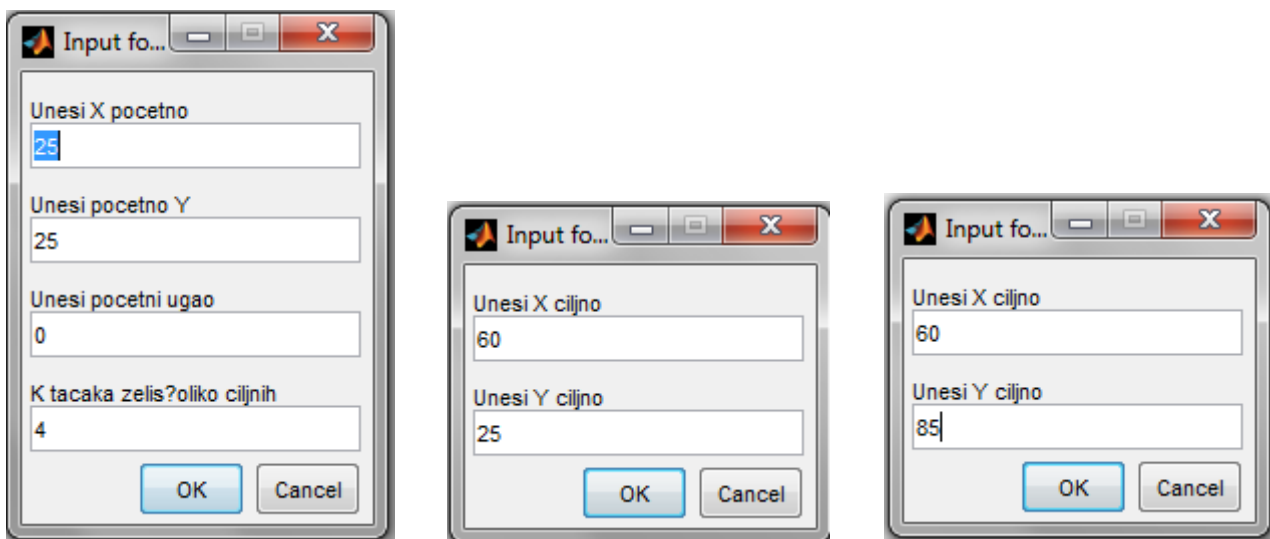
$[map]$ - Уносе се координате контролиних тачака

(x, y) - Задају се координате почетне, међутачака и крајње тачке

(d_{max}) - Дозвољено одступање положаја робота у односу на задату тачку

(\dot{f}_i_{max}) - Највеће дозвољено одступање тренутног правца кретања у односу на задати правац

Да би се установило да ли се робот креће по задатој путањи и са довољном тачношћу стиже на циљ, задају се укупно четири тачке: почетна, три међутачке и циљна тачка.



Слика 11. 9: Изглед прозора за унос координата тачака

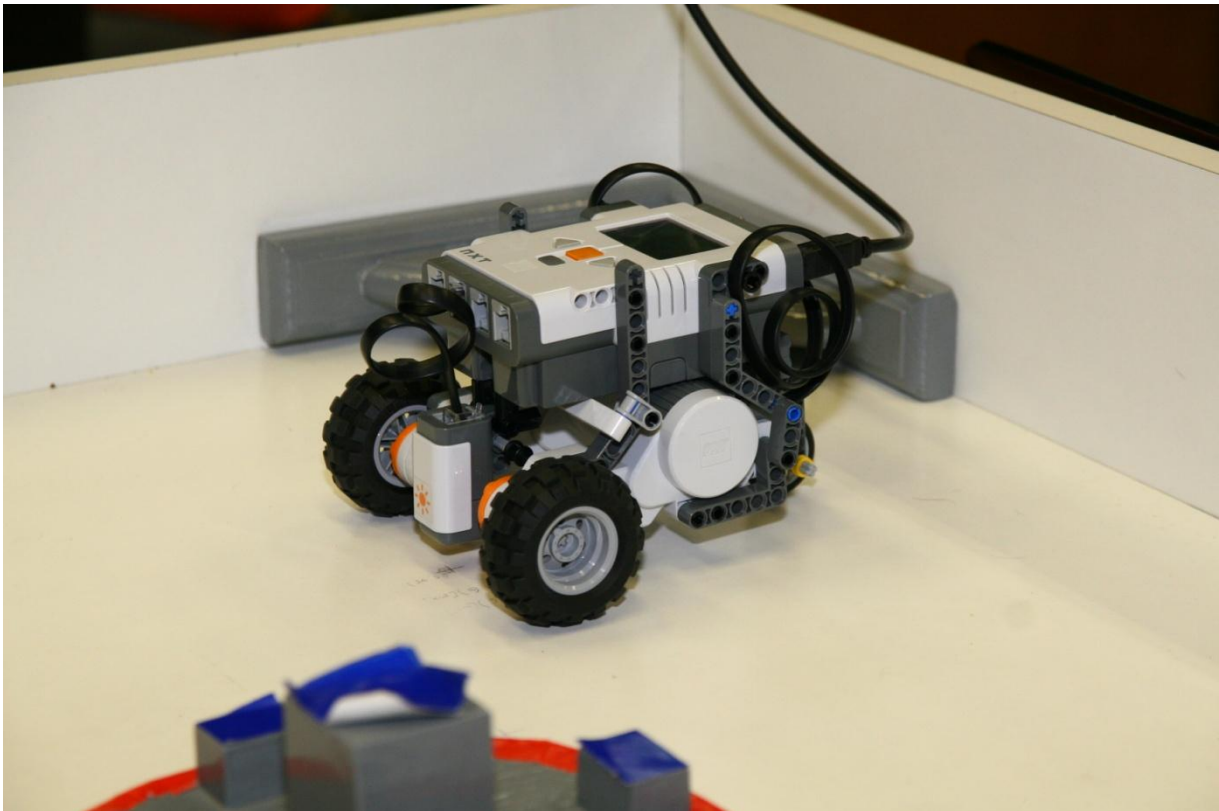
Робот се поставља у почетну тачку и задају се почетни параметри:

- $b = 11.6cm$
- $r = 3cm$
- $[map] = [50\ 25; 60\ 50; 60\ 70; 60\ 95; 45\ 85][cm]$
- $x = [25, 60, 60, 60, 20]$
- $y = [25, 25, 60, 85, 85]$
- $(d_{max}) = 10[cm]$

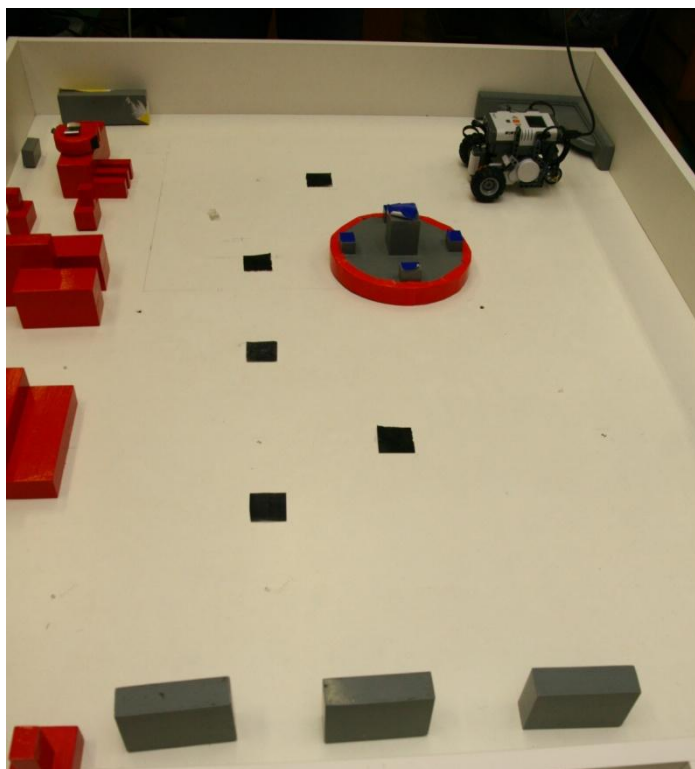
- $(f_i \max) = 30^\circ$

Програм којим се управља кретањем мобилног робота је заправо неколико програма, где је један главни програм, а остали су подпрограми који се командама позивају по потреби. Главни програм успоставља везу за роботом, контролише моторе, позива подпрограм за читавање података са сензора за светлост, одлучује на основу излаза из неуроске мреже, садржи координате референтних тачака тачака на основу којих врши корекцију положаја у случају да их при кретању робота сензор чита. У њега се уписују и претходно наведени параметри: пречник тачка, растојање између тачкова и др.

Пошто је робот укључен и повезан са компјутером преко USB кабла, поставља се робот на почетну тачку (слика 11.2 и 11.3) и покреће се програм.

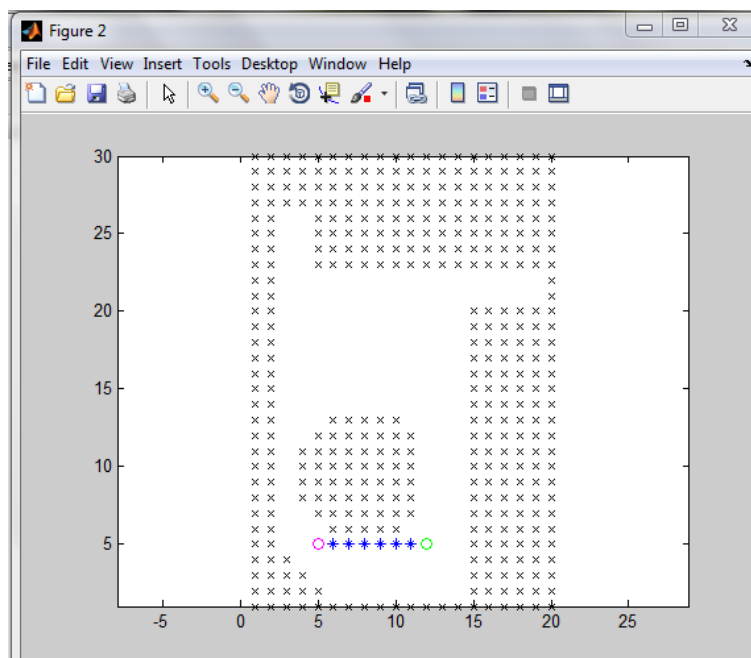


Слика 11. 10: Почетни положај мобилног робота



Слика 11. 11: Почетни положај мобилног робота сниман из другог угла

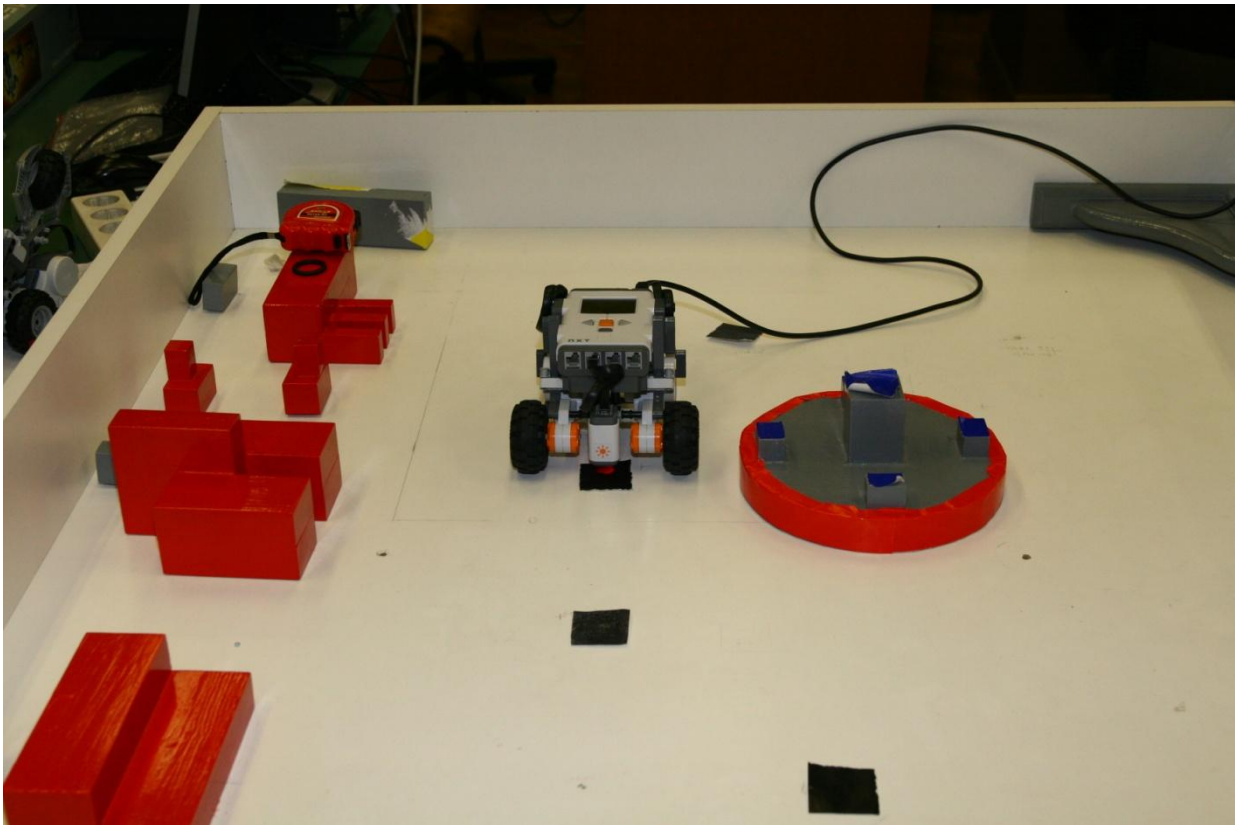
После иницијалног успостављања везе са роботом започиње се извршавање програма прорачуном A* алгоритма. Пошто је раније у овом тексту објашњен начин рада овог подпрограма овде нема потребе за разјашњењима већ преостаје процена како ради у реалном времену. Уверили смо се да увек бира најкраћу путању, брзо и ефикасно (слика 11.4).



Слика 11. 12: Избор путање

Одређивањем оптималне путање генеришу се и додатне међутачке (слика 11.4) на основу којих се изводи кретање. За сваку од тих тачака програм израчунава, упоређујући тренутну позицију и оријентацију робота са наредном тачком, растојање и потребну

оријентацију робота. Уколико се тренутна оријентација, то јест правац кретања, разликује од захтеване за више од $f_i \max$ тада је потребно извршити ротацију робота пре кретања напред. У том случају команду за ротацију вратила мотора одређује вештачка неуронска мрежа која је развијена и тренирана за те потребе. Након изведене ротације, уколико је за њом уопште и било потребе, врши се кретање унапред све док се растојање до наредне тачке не смањи на вредност $d \max$ или мању од ње. На слици 11.5 је приказан мобилни робот у току кретања по задатој путањи.

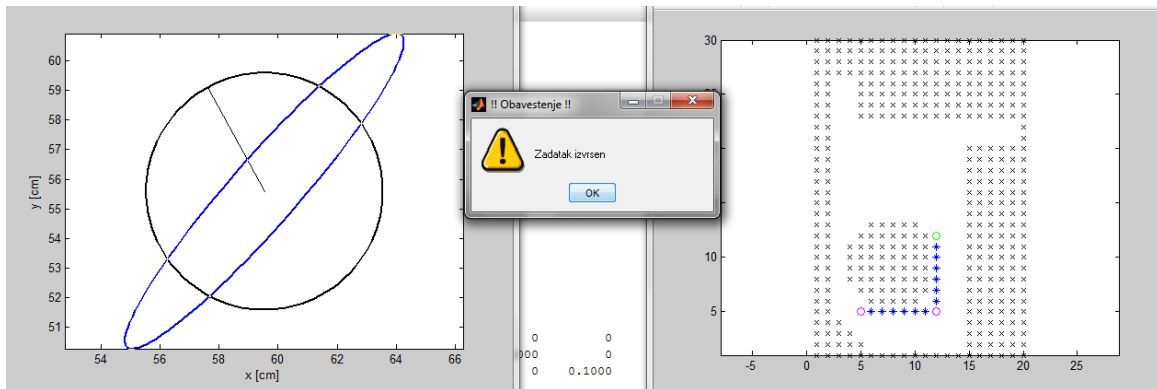


Слика 11. 13: Кретање мобилног робота

Програм у току кретања робота стално израчунава вредност тренутне позиције (x, y) и угао оријентације (θ) у глобалном координатном систему. Ово рачунање се врши према усвојеном моделу кретања на основу резултата мерења угаоног помераја вратила мотора путем енкодера. Наравно, мотори не извршавају задате команде идеално нити сензори врше идеално тачна мерења. То доводи до тога да се стварни положај разликује од израчунатог. Због тога је потребно, у току кретања, вршити корекцију израчунатог положаја робота. Корекција израчунатог положаја робота се врши помоћу референтних тачака које су постављене произвољно на путањи робота и вештачке неуронске мреже која на основу информација са сензора одлучује да ли треба вршити корекцију. Важно је напоменути да се мерења помоћу светлосног сензора изводе искључиво када робот стоји, у паузи између два кретања. Дакле, када се сними тамна подлога процењује се на основу израчунатих координата тренутног положаја и координата свих референтних тачака, која од њих има највећу вероватноћу да је детектована. Након тога израчунати положај преузима координате референтне тачке са највећом вероватноћом. Овај поступак се изводи помоћу математичког апарата - Калмановог филтера и представља локализацију мобилног робота у глобалном координатном систему. Све време у току кретања робота имамо увид у овај процес процене сигурности у стварни положај преко графичког приказа. Ова сигурност опада у току кретања, како се грешке нагомилавају, све док се не прочита нека од референтних тачака.

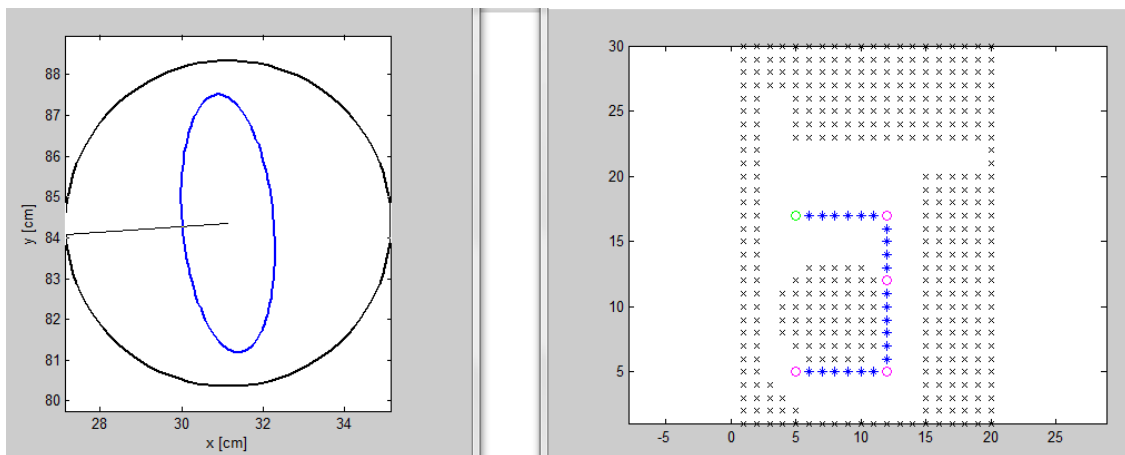
Након што је робот стигао до прве задате тачке, програм даје обавештење да је задатак извршен, што се односи у ствари значи да је један циклус завршен. Одмах затим се

активира поново A* алгоритам само сада за нове вредности, тј. следећу задату тачку. На графику се појављује још једна путања која се наставља на претходну (слика 11.6).

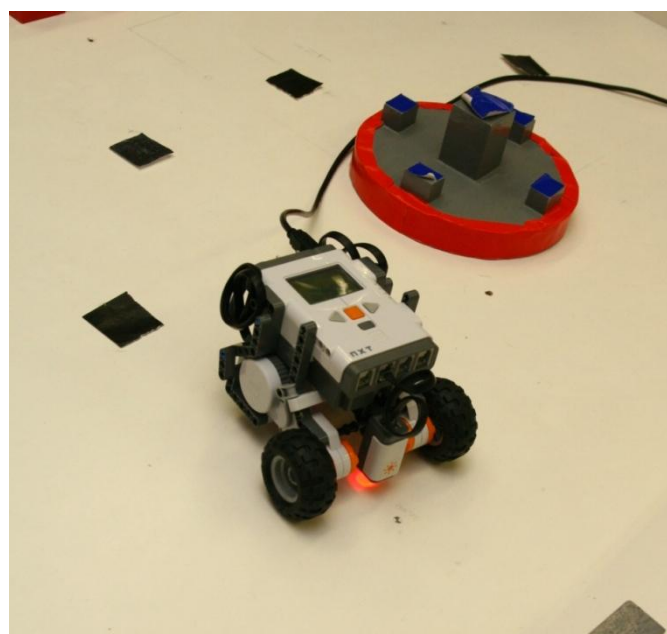


Слика 11. 14: Генерисање друге тачке и елипсе сигурности

Тај процес се понавља за све задате тачке док се не дође до крајњег циља (последње задате тачке). Кретање робота за преостале тачке је приказано на слици 11.7 и 11.8.



Слика 11. 15: Генерисање циљне тачке и елипсе сигурности



Слика 11. 16: Осварено кретање робота

Приказ експерименталних резултата дат је у табели у наставку.

Табела 12: Експериментални резултати

Редни број	Стартна позиција	Број детектованих референтних тачака	Жељена позиција	Израчуната позиција	Остварена позиција	Задатак извршен
1	(25, 25)	I, IV	(25, 85)	(31, 85)	(35, 93.5)	Да
2		I, II, IV		(30, 85.5)	(33, 89.5)	Да
3		I		(30, 84.5)	(35, 84.5)	Да
4		I, II, II		(30, 85.5)	(29.5, 82)	Да
5		I, II, III, IV		(29, 86)	(31.5, 92)	Да
6		I, II, IV		(31, 86)	(25, 85.5)	Да
7		II, III, IV		(31, 86)	(27.5, 86.5)	Да
8		I		(29.5, 85.5)	(31, 80.5)	Да
9		I, II, III		(29, 85)	(25.5, 81.5)	Да
10		I, II, III, IV		(31, 84)	(27.5, 86.5)	Да

12. Анализа и дискусија експерименталних резултата

На крају извођења експеримента пројектног задатка извршена је успешна реализација идејног решења. Постигнуте грешке позиционирања налазе се у границама задовољавајућих. У неким пробама су остварени резултати и бољи од очекиваних. Може се закључити да су мобилни роботи на оваквим идејним основама, уз модификације потребне за реалну имплементацију, погодни за потребе транспорта. Међутим, треба знати да овај вид транспорта мора бити у већој мери прецизан и мање склон проблемима. За почетак, несавршености самог актуаторског система онемогућују прецизно позиционирање робота. Сваки мотор има по девет зупчаника који служе за пренос обртног момента до излазног вратила. Зупчаници су направљени од пластике и имају велике зазоре. Све то утиче на појаву грешака кретања, које програм не региструје. Последица тога је појава грешке између задате позиције робота и постигнуте позиције, што је у индустријским условима недопустиво. Проблем се увећава у тренуцима када мерни систем робота услед поменутих проблема није у могућности да израчуна постигнуту позицију довољно прецизно, тако да робот иако "мисли" да зна где се налази његов стварни положај то не потврђује. Све три позиције (задата, израчуната и постигнута) се разликују. Проблем је још већи јер се ове грешке за исти експеримент и исте тачке које робот треба да посети некада јављају, а некада не.

Недостатак се огледа у делу кода за дефинисање угла окретања који је спрегнут са неуронском мрежом која је подложна различитим излазима мање или веће прецизности и поред претходног детаљног обучавања. Посебно је тешко остварити приближно тачно кретање по путањи ако је угао скретања близу деведесет степени или више. Локализација помоћу Калмановог филтера у великој мери помаже да се постигне већа прецизност при кретању, али само у случају када се контролне тачке поставе на одговарајућу позицију и под условом да их оптички сензор детектује. Дешава се пак, да се контролне тачке услед свеобухватне несигурности система не достигну и тада се проблем опет своди на горе описани. На послетку, можост се закључити да се овај систем и даље може унапређивати са циљем смањења грешака. Начин остварења тог циља је првенствено програмерска ингениозност и довитљивост, заједно са умешношћу синтезе адекватне конструкције робота. Проблеми позиционирања се сигурно могу решити применом система вештачког гледања (енг. Vision System). То би био извор додатних информација на основу којих би програм могао да врши континуалне корекције кретања мобилног робота по задатој путањи, дакле препознавањем своје околине. Могу се применити и друга решења као што је повећање укупне тачности система квалитетнијом израдом компоненти, уклањањем или смањењем механичких несавршености система, коришћењем различитих сензора и слично. Питање је да ли су таква унапређења финансијски оправдана, имајући у виду да је роботима и даље највећа конкуренција јефтина радна снага. Пре избора решења треба разумети суштину проблема. Сам склоп система не мора бити превише прецизан и тачан, ако се настале грешке могу компензовати преко повратне спреге тачним и правовременим информацијама из окружења робота. Значи, боље је усавршавати онај интелигентни (управљачки) део система него механички, јер управљањем се могу кориговати механички недостаци система док је обрнуто тешко изводљиво.

13. Закључак

Интелигентно понашање мобилног робота засновано је на коришћењу неуронских мрежа као парадигме вештачке интелигенције. Кроз детерминисање угла ротације робота и обраду информација са светлосног сензора имплементирани су основне функције мрежа, функционална апроксимација и класификација. На основу модела кретања и вредности читавања са унутрашњих сензора обезбеђено је познавање позиције и оријентације робота у сваком тренутку рада. Калманов филтер имао је пресудну улогу у отклањању неусаглашености између остварене и израчунате позиције. Вршио је корекцију положаја. Коришћењем A^* алгоритма генерисна је најкраћа путања између задатих почетних и циљних тачака. Такође, за задато окружење посредством овог алгоритма вршена је и функција избегавања препрека. Целокупан програмски код реализован је у MatLab [9] софтверском пакету.

На основу спроведеног експеримента може се закључити да су резултати задовољавајући. Мобилни робот је пратио задату путању, кориговао израчунату позицију и кретање на основу контролних тачака и стигао у крајњу позицију избегавајући препреке на путу.

Ипак, понављајући поступак за различите путање и различите распореде референтних тачака, примећено је да успешност извршавања задатака зависи од неколико фактора.

Као прво, од велике важности је унос тачних вредности за пречник тачка r и растојање између истих b . Кориговање грешака које се акумулирају током кретања је тешко изводљиво уколико су те вредности нетачне.

Следећи проблем који је уочен је дужина путање, односно растојање између две узастопне тачке. Што је растојање веће, грешка кретања је већа и ту долази до изражаја неуједначеност рада мотора. Чини се да је један увек јачи, због чега долази до скретања мобилног робота у једну страну са задате путање. Такође је и акумулација грешака велика. Тај проблем делимично се решава задавањем више тачака до циља за исту путању.

Примећено је и то да је кретање робота тачније ако се очита прва референтна тачку на путањи. Статистички посматрано то је тачка која је током експеримента и била највише детектована сензором робота. Уколико се не очита прва референтна тачка постизање циљне позиције је ипак оствариво. С друге стране сасвим је сигурно да се кретање до задате циљне тачке неће завршити ако се на почетку кретања јави велика грешка позиционирања. Тада се дешава да робот дође у колизију са неком од машина у окружењу. Уочени проблем се решава постављањем прве референтне тачке ближе, али не преблизу почетне позиције. Тако расте вероватноћа да ће се очитати и следећа референтна тачка, а понекад и све тачке у окружењу.

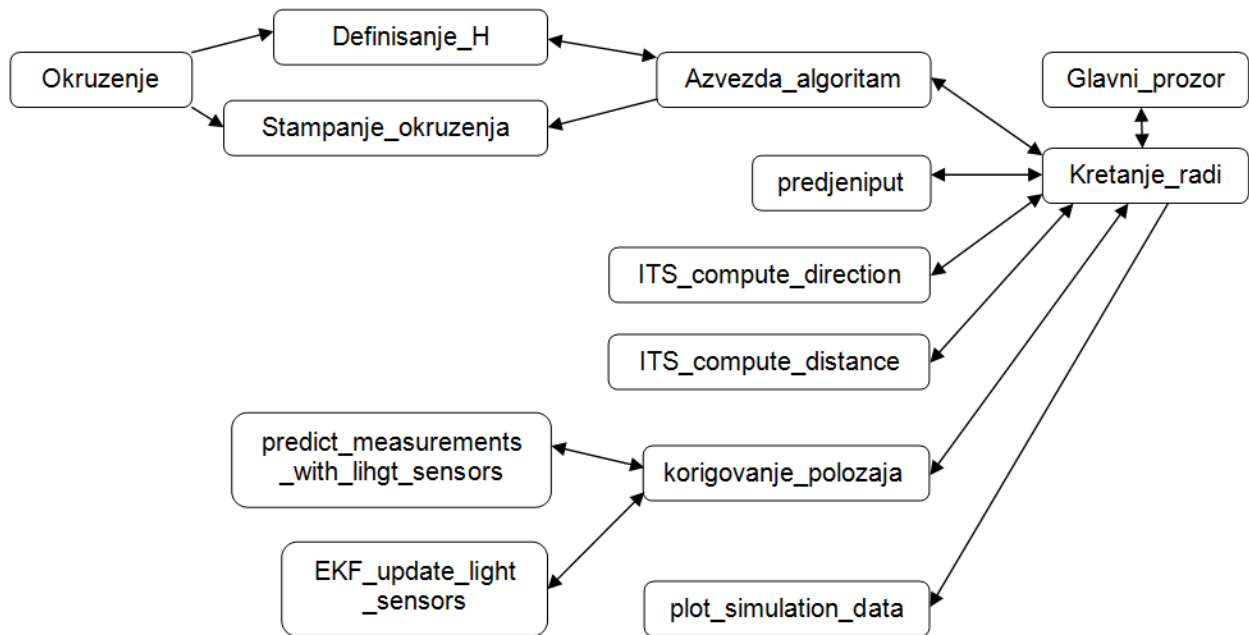
Закључујемо да се овај вид решења проблема унутрашњег транспорта може успешно применити. Потребно је само на приложен начин отклонити уочене недостатке. Предност употребе роботизованог транспорта је вишеструка и огледа се у повећану продуктивности, осавремењавању погона, стандардизовању квалитета израде, као и хуманизацији рада.

Симулација у AnyLogic окружењу омогућује сагледавање производних токова, анализу истих и предлагање решења које би довело до повећања перформансеи производног процеса.. Потреба за моделирањем осећа се у широком спектру области, а посебно у индустрији и производњи.

14. Литература

- [1] Бабић, Б., Предавања на предмету Компјутерска симулација и вештачка интелигенција, Универзитет у Београду -Машински факултет, Београд, 2007.
- [2] Вуковић, Н., Материјали са аудиторних вежби на предмету Интелигентни технолошки системи (ПРО220-0131), Универзитет у Београду -Машински факултет, Београд, 2010.
- [3] Милутиновић, Д., Предавања на предмету Идустиријски роботи, Универзитет у Београду - Машински факултет, Београд, 2009.
- [4] Миљковић, З., Предавања на предмету Интелигентни технолошки системи (ПРО220-0131), Универзитет у Београду -Машински факултет, Београд, 2009.
- [5] Миљковић,З., Системи вештачких неуронских мрежа у производним технологијама, Серија монографских дела Интелигентни технолошки системи (Уредник серије: Проф. др Владимир Милачић), Књига 8, Универзитет у Београду -Машински факултет, Београд, 2003.
- [6] Миљковић,З.,Александрић,Д., Вештачке неуронске мреже, збирка решених задатака са изводима из теорије, Универзитет у Београду -Машински факултет, Београд, 2009
- [7] <http://mindstorms.lego.com/en-us/Default.aspx>, датум последњег приступа 14.02.2011.
- [8]<http://www.mindstorms.rwth-aachen.de/>, датум последњег приступа 14.02.2011.
- [9] <http://www.mathworks.com/products/matlab/>, датум последњег приступа 05.02.2011.
- [10] <http://www.fmp.rs>, датум последњег приступа 08.02.2011.
- [11] <http://www.siemens.com>, датум последњег приступа 11.02.2011.
- [12] <http://www.wikipedia.com>, датум последњег приступа 14.02.2011.

15. Прилог А



Слика 15. 1: Алгоритам коришћења програмских кодова

„Glavni_prozor“

```

clear all
close all
clc
prompt = {'Unesi X pocetno','Unesi pocetno Y','Unesi pocetni ugao', 'K tacaka zelis?oliko ciljnih'};
dlg_title = 'Input for peaks function';
num_lines = 1;
def = {'25','25','0','2'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
Xa(1,1)=str2num(answer{1});
Xa(1,2)=str2num(answer{2});
Xa(1,3)=str2num(answer{3})/57.3;
n=str2num(answer{4});
for i=1:n
    prompt = {'Unesi X ciljno','Unesi Y ciljno'};
    dlg_title = 'Input for peaks function';
    num_lines = 1;
    def = {' ',' ',' '};
    answer = inputdlg(prompt,dlg_title,num_lines,def);
    Xb(i,1)=str2num(answer{1});
    Xb(i,2)=str2num(answer{2});
end
figure
for i=1:n
    if i==1
        Xp=Xa;
        Yc(1,1)=Xb(1,1);
    end
end
  
```

```

    Yc(1,2)=Xb(1,2);
    [tetamm]=Kretanje_radi(Xp',Yc);
else
    Xp(1,1)=Xb(i-1,1);
    Xp(1,2)=Xb(i-1,2);
    Xp(1,3)=tetamm;
    Yc(1,1)=Xb(i,1);
    Yc(1,2)=Xb(i,2);
    [tetamm]=Kretanje_radi(Xp',Yc);
end
    warndlg('Zadatak izvrsen','!! Obavestenje !!')
end

```

„Kretanje_radi“

```

function [tetamm]=Kretanje_radi(X,Xci)
handle = COM_OpenNXT('USB.ini', 'check');
COM_SetDefaultNXT(handle);
OpenLight(SENSOR_2,'ACTIVE');
s = GetLight(SENSOR_2)
load mreza2;
load netff
load crna_boja1
load crna_boja2
load crna_boja3
load crna_boja4
load crna_boja5
load bela_boja1
load bela_boja2
load bela_boja3
load bela_boja4
load bela_boja5
yp=[];
black = [crna_boja1,crna_boja2,crna_boja3,crna_boja4,crna_boja5];
white = [bela_boja1,bela_boja2,bela_boja3,bela_boja4,bela_boja5];
min_black = min(min(black))
max_white = max(max(white))
yp=[]
s = GetLight(SENSOR_2)
xi = 2*(s-min_black)/(max_white-min_black) -1
Yp=sim(netff,xi)
yp=[yp;Yp]
fi=0;
s = GetLight(SENSOR_2)
xi = 2*(s-min_black)/(max_white-min_black) -1
Yp=sim(netff,xi)
yp=[yp;Yp]
C = [1 0 0;0 1 0; 0 0 1]; % initial uncertainty.
d=0;
C=eye(3)*0.1
b=11.6; r=3;
QL=[0 0;0 0];
Q=[100 0;0 100];
map=[30 40; 0 70; 100 100]%;50 10;50 30;%koordinate crnih racaka u okruzenju
[path]=Azvezda_algoritam(X(1),X(2),Xci(1),Xci(2));
sizpath = size(path,1);

```

```

StopMotor('all', 'off');
ResetMotorAngle(MOTOR_B);
ResetMotorAngle(MOTOR_C);
dmax = 8; % u cm
fimax = 30; % u stepenima
dmin = 1;
fimin = 10;
r = 3; %poluprecnik tocka robota
s = GetLight(SENSOR_2)
xi = 2*(s-min_black)/(max_white-min_black) -1
Yp=sim(netff,xi)
yp=[yp;Yp]
if Yp<=0
    [X,C]=korigovanje_polozaja(X,C,Q,map)
end
for k = 1 : sizpath
    StopMotor('all', 'off');
    ResetMotorAngle(MOTOR_B); ResetMotorAngle(MOTOR_C);
    SetMotor(MOTOR_C); % accesses the motor b
    SyncToMotor(MOTOR_B);
    d = ITS_compute_distance(X, path(k,:))
    fi = ITS_compute_direction(X,path(k,:))*57.3
    s = GetLight(SENSOR_2)
    xi = 2*(s-min_black)/(max_white-min_black) -1
    Yp=sim(netff,xi)
    yp=[yp;Yp]
    if Yp<=0
        [X,C]=korigovanje_polozaja(X,C,Q,map)
    end
    if abs(fi) >= fimax
        if fi <= 0
            trrot = 100;
        else trrot = -100;
        end
        s = GetLight(SENSOR_2)
        xi = 2*(s-min_black)/(max_white-min_black) -1
        Yp=sim(netff,xi)
        yp=[yp;Yp]
        alrot = round(sim(mreza2,fi))
        StopMotor('all', 'off');
        ResetMotorAngle(MOTOR_B); ResetMotorAngle(MOTOR_C);
        SetMotor(MOTOR_C); % motor b
        SyncToMotor(MOTOR_B); % motor c
        SetPower 20 %
        SetTurnRatio (trrot) % Obratiti paznju na ovu
        SetAngleLimit (alrot) % kao i na ovu funkciju!
        SendMotorSettings
        WaitForMotor(MOTOR_B);
        MC = GetMotorSettings(MOTOR_C);
        MB = GetMotorSettings(MOTOR_B);
        dsl = MC.Angle/57.3*r
        dsr = MB.Angle/57.3*r
        s = GetLight(SENSOR_2)
        xi = 2*(s-min_black)/(max_white-min_black) -1
        Yp=sim(netff,xi)
        yp=[yp;Yp]
        [X,C] = predjeniput(X(1),X(2),X(3), dsl,dsr)
    end
end

```

```

end
while d > dmax
    StopMotor('all', 'off');
    ResetMotorAngle(MOTOR_B); ResetMotorAngle(MOTOR_C);
    SetMotor(MOTOR_C); % accesses the motor b
    SyncToMotor(MOTOR_B); % provides same controls to motor c
    SetPower 30 %
    SetTurnRatio 0%
    SetAngleLimit 100 %
    SendMotorSettings
    WaitForMotor(MOTOR_B);
    s = GetLight(SENSOR_2)
    xi = 2*(s-min_black)/(max_white-min_black) -1
    Yp=sim(netff,xi)
    yp=[yp;Yp]
    if Yp<=0
        [X,C]=korigovanje_polozaja(X,C,Q,map)
    end
    MC = GetMotorSettings(MOTOR_C);
    MB = GetMotorSettings(MOTOR_B);
    dsl = MC.Angle/57.3*r
    dsr = MB.Angle/57.3*r
    [X,C] = predjeniput(X(1),X(2),X(3), dsl,dsr)
    d = ITS_compute_distance(X, path(k,:))
    fi = ITS_compute_direction(X,path(k,:))*57.3
    plot_simulation_data(X, C),hold on;
    if d <= dmin || abs(fi) >= fimin
        if fi <=0
            trrot = 100
        else trrot = -100
        end
        alrot = round(sim(mreza2,fi))
        StopMotor('all', 'off');
        ResetMotorAngle(MOTOR_B); ResetMotorAngle(MOTOR_C);
        SetMotor(MOTOR_C);
        SyncToMotor(MOTOR_B);
        SetPower 20 %
        SetTurnRatio (trrot) %
        SetAngleLimit (alrot) %
        SendMotorSettings
        pause(1)
        WaitForMotor(MOTOR_B);
        pause(1)
        s = GetLight(SENSOR_2)
        xi = 2*(s-min_black)/(max_white-min_black) -1
        Yp=sim(netff,xi)
        yp=[yp;Yp]
        if Yp<=0
            [X,C]=korigovanje_polozaja(X,C,Q,map)
        end
        MC = GetMotorSettings(MOTOR_C);
        MB = GetMotorSettings(MOTOR_B);
        dsl = MC.Angle/57.3*r
        dsr = MB.Angle/57.3*r
        [X,C] = predjeniput(X(1),X(2),X(3), dsl,dsr)
        continue % go back to the main while loop
    end%if
end

```

```

if d <= dmin || abs(fi) >= fimin
    if fi <=0
        trrot = 100
    else trrot = -100
    end
    alrot = round(sim(mreza2,fi))
    StopMotor('all', 'off');
    ResetMotorAngle(MOTOR_B); ResetMotorAngle(MOTOR_C);
    SetMotor(MOTOR_C);
    SyncToMotor(MOTOR_B);
    SetPower 20 %
    SetTurnRatio (trrot) %
    SetAngleLimit (alrot) %
    SendMotorSettings
    pause(1)
    WaitForMotor(MOTOR_B);
    s = GetLight(SENSOR_2)
    xi = 2*(s-min_black)/(max_white-min_black) -1
    Yp=sim(netff,xi)
    yp=[yp;Yp]
    if Yp<=0
        [X,C]=korigovanje_polozaja(X,C,Q,map)
    end
    pause(1)
    MC = GetMotorSettings(MOTOR_C);
    MB = GetMotorSettings(MOTOR_B);
    dsl = MC.Angle/57.3*r
    dsr = MB.Angle/57.3*r
    [X,C] = predjeniput(X(1),X(2),X(3), dsl,dsr)
    d = ITS_compute_distance(X, path(k,:))
    fi = ITS_compute_direction(X,path(k,:))*57.3
    plot_simulation_data(X, C),hold on;
    continue % go back to the main while loop
end%if
s = GetLight(SENSOR_2)
xi = 2*(s-min_black)/(max_white-min_black) -1
Yp=sim(netff,xi)
yp=[yp;Yp]
end%while
StopMotor('all', 'off');
ResetMotorAngle(MOTOR_B); ResetMotorAngle(MOTOR_C);
tetamm=X(3);
end%for; end of the main loop

```

„Azvezda_algoritam“

```
function [path]=Azvezda_algoritam(xp,yp,xg,yg)
path=[];
[ok1,b]=Definisanje_H(xg/5,yg/5);
k=0;
xk=xp/5;
yk=yp/5;
figure(2)
plot(xp/5,yp/5,'om');
hold on
plot(xg/5,yg/5,'og');
hold on
for k=1:100;
    min=50;
    okm=100;
    for i=-1:1
        xt=xk+i;
        for j=-1:1
            if xt<1 || xt>b(1)
                break
            end
            yt=yk+j;
            if yt<1 || yt>b(2)
                else
                    if xt==xk && yt==yk
                        ok1(xk,yk)=50;
                    end
                    if ok1(xt,yt)<okm
                        okm=ok1(xt,yt);
                        path(k,1)=xt*5;
                        path(k,2)=yt*5;
                        min=ok1(xt,yt);
                        xk1=xt;
                        yk1=yt;
                    end
                end
            end
        end
    end
    sp=size(path);
    if xg==path(sp(1),1) && yg==path(sp(1),2)
        break
    end
    xk=xk1;
    yk=yk1;
    plot(xk,yk,'*');
    hold on
    axis equal
end
Stampanje_okruzenja(b)
```


„ITS_compute_distance“

```
function d = ITS_compute_distance(pose, cp)
dx = cp(1)-pose(1);
dy = cp(2)-pose(2);
d = sqrt((dx)^2 + (dy)^2);
```

„ITS_compute_direction“

```
function [phi] = ITS_compute_direction(pose, cp)
alfa = atan2((cp(2) - pose(2)),(cp(1)-pose(1)));
phi = alfa - pose(3);
if phi >= pi
    phi = phi - 2*pi;
elseif phi <= -pi
    phi = phi + 2*pi;
end
```

„korigovanje_polozaja“

```
function [X,C]=korigovanje_polozaja(X,C,Q,map)
map = [30 40; 0,70; 100, 100]
xls = 0.01
yls = 0
N = []
e = 10
for k = 1 : size(map,1)
    [zp,Hp] = predict_measurements_with_lihgt_sensors(X,map(k,:), xls, yls)
    z = map(k,:)'
    S= Hp*C*Hp' + Q;
    nis= (z-zp)*inv(S)*(z-zp);
    N = [N;nis]
end
j = find(N<=e)
if isempty(j)
    return
else
    Z = map(j,:)'
    [X, C] = EKF_update_light_sensors(X, C, Z, zp, Hp, Q)
end
```

„predict_measurements_with_lihgt_sensors“

```
function [varargout] = predict_measurements_with_lihgt_sensors(pose,map, xls, yls)
if isempty(map)
    varargout{1} = [];varargout{2} = [];
    return;
end;
n = size(map,1);
dx = pose(1) + xls*cos(pose(3)) - yls*sin(pose(3));
```

„EKF_update_light_sensors“

```
function [pose, Cov] = EKF_update_light_sensors(pose, C, z, zp, H, Q)
if isempty(zp) || isempty(z)
    return
else
    invec = z - zp
    CHt = C*H'
    S = H*CHt + Q
    Sinv = inv(S)
    K = CHt*Sinv
    up= K*invec
    pose = pose + up
    Cov = (eye(3) - K*H)*C
end
dy = pose(2) + xls*sin(pose(3)) + yls*cos(pose(3));
h11 = 1; h12 = 0; h13 = -xls*sin(pose(3)) - yls*cos(pose(3));
h21 = 0; h22 = 1; h23 = xls*cos(pose(3)) - yls*sin(pose(3));
h = [h11 h12 h13;...
     h21 h22 h23];
zp = [dx;dy];
h = h;
varargout{1} = zp; varargout{2} = h;
```

„plot_simulation_data“

```
function plot_simulation_data(sv,C_)
mapa = [0 0;100 0;100 150;0 150;0 0];
fig = figure(1);
set(fig, 'name', 'EKF_LOCALIZATION');
set(gca,'NextPlot','replace');
axis ([-10 110 -10 160]);
line(mapa(:,1),mapa(:,2),'LineWidth',4)
p_ = drawmob_robot(sv(1:3),8,'k'); hold on;
d2 = prob_ellipse(sv,C_,'b',1);hold on
xlabel('x [cm]');
ylabel('y [cm]');
axis equal
```