



Универзитет у Београду Машински факултет

Дипломске 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 и Најдан Вуковић, дипл. маш.инж		
Потпис наставника:	Група:			
	РБ	Презиме и име:	Бр.инд.	Потпис:
	1.	Петровић Милица	1076/08	
	2.	Божовић Илија	1109/08	
	3.	Јуковић Семир	1145/08	
	4.	Лукић Никола	1247/08	
5.	Југовић Александар	1027/08		

Школска година: 2009/2010.

УНИВЕРЗИТЕТ У БЕОГРАДУ – МАШИНСКИ ФАКУЛТЕТ

Дипломске академске студије – 2. година

Модул: **ПРОИЗВОДНО МАШИНСТВО**, шк. год. 2009/2010.

Предмет: **ИНТЕЛИГЕНТНИ ТЕХНОЛОШКИ СИСТЕМИ (ПРО220-0131)**

Предметни наставници: проф. др Зоран Миљковић и проф. др Бојан Бабић

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 и bbojovic@mas.bg.ac.rs;

Датум издавања задатка: **16.10.2009.**

Рок завршетка задатка: **04.12.2009.**

Задатак издао: _____

(Најдан Вуковић)

Резиме

Употреба интелигентних технолошких система базираних на парадигмама вештачке интелигенције и компјутерски интегрисаним технологија је неопходна како би се одговорило потребама које нам тржиште поставља. Повећање продуктивности, унапређење технологије, искључивање могућности грешке људског фактора и др. само су неке од последица употребе интелигентних мобилних робота у неком технолошком окружењу. У овом раду анализирана је могућност употребе мобилног робота у циљу решавања проблема унутрашњег транспорта материјала, сировина и готових делова. Познато је да би решење овог проблема омогућило оптимизацију технолошког процеса. Развијено решење заснива се на употреби мобилног робота *LEGO Mindstorms NXT* управљаним применом *RWTH Toolbox* – а у софтверском окружењу *MATLAB*. Применом вештачких неуронских мрежа у поменутом софтверу обезбеђено је да се непоуздана сензорска мерења пресликају у поуздане информације које ће представљати основу за наредну акцију робота. Коришћењем Калмановог филтера обезбеђено је решавање проблема локализације робота, а одабир оптималне путање A^* алгоритмом претраге. Софтверским пакетом *AnyLogic* симулирани су транспортни токови материјала унутар технолошког окружења у које би требало имплементирати мобилног робота. Експериментални и симулациони резултати показују да је могуће применити мобилног робота за обављање постављеног задатка транспорта материјала у оквиру технолошког окружења.

Кључне речи: интелигентни технолошки системи, вештачке неуронске мреже, интелигентни мобилни робот, локализација, Калманов филтер, A^* алгоритам, симулација.

¹ **Илија Божовић 1109/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија.

Е-пошта: ilijabozovic@gmail.com

² **Александар Југовић 1027/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија.

Е-пошта: jugovic169@gmail.com

³ **Семир Јуковић 1145/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија.

Е-пошта: jukovics@yahoo.com

⁴ **Никола Лукић 1247/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија.

Е-пошта: lukic2205@yahoo.com

Милица Петровић 1076/08, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија.

Е-пошта: milicapetrovic351@yahoo.com

Списак слика

Слика 2.1.1:	Layout производно монтажног погона предузећа „XY“.....	8
Слика 3.1:	Алгоритам концепцијског решења пројектног задатка.....	15
Слика 4.1.1:	Серво мотор из комплета <i>LEGO Mindstorms NXT</i>	16
Слика 4.1.2:	Сензор додира.....	17
Слика 4.1.3:	Сензор звука.....	17
Слика 4.1.4:	Светлосни сензор.....	18
Слика 4.1.5:	Ултразвучни сензор.....	18
Слика 4.1.6:	Кутија са батеријама мобилног робота.....	18
Слика 4.1.7:	Изглед управњачке јединице са интерфејсима.....	19
Слика 4.1.8:	Приказ улазно-излазних портова <i>LEGO Mindstorms NXT</i> робота.....	20
Слика 4.1.9:	Прва фаза конфигурисања мобилног робота.....	21
Слика 4.1.10:	Друга фаза конфигурисања мобилног робота.....	21
Слика 4.1.11:	Изглед коначне конфигурације мобилног робота.....	22
Слика 4.1.12:	Приказ отвореног кућишта мотора са позицијом инкременталног енкодера.....	22
Слика 4.1.13:	Приказ препознавања боја светлосним сензором.....	23
Слика 4.1.14:	Упростијени шематски приказ светлосног сензора.....	23
Слика 4.2.1:	Положај и оријентација мобилног робота у равни x-y.....	25
Слика 4.2.2:	Брзински модел кретања мобилног робота у равни x-y.....	25
Слика 4.3.1:	Ток информација кроз развијени неуронски модел за одређивање команде за ротацију вратила мотора у реалном времену.....	28
Слика 4.3.2:	Пример ротације робота у месту за 30°.....	29
Слика 4.3.3:	Пример графичког приказа симулације ВММ која даје добре резултате.....	31
Слика 4.3.4:	Пример графичког приказа симулације ВММ која даје лоше резултате.....	32
Слика 4.3.5:	Топологија изабране ВММ за одређивање команде за ротацију вратила мотора у односу на потребну ротацију робота у глобалном координатном систему.....	33
Слика 4.3.6:	Приказ резултата тренирања изабране ВММ за одређивање команде за ротацију вратила мотора у односу на потребну ротацију робота у глобалном координатном систему.....	34
Слика 4.3.7:	Пример графичког приказа симулације ВММ која је изабрана.....	34
Слика 4.3.8:	Пример произвољне диспозиције контролних тачака у радном окружењу....	35
Слика 4.3.9:	Ток информација кроз развијени неуронски модел за детекцију контролних тачака у радном окружењу робота.....	35
Слика 4.3.10:	Графички приказ изабраног репрезентативног резултата мерења сензора..	36
Слика 4.3.11:	Топологија изабране ВММ за детекцију контролних тачака у радном окружењу мобилног робота.....	39
Слика 4.3.12:	Приказ резултата тренирања изабране ВММ за детекцију контролних тачака у радном окружењу мобилног робота.....	40
Слика 4.3.13:	Резултат симулације изабране ВММ за улазе из обучавајућег и тест скупа	41
Слика 4.4.1:	Симболички приказ естиматора за предикцију и корекцију.....	42
Слика 4.4.2:	Графички приказ Гаусове нормалне расподеле.....	44
Слика 4.5.1:	Приказ осмоспојивог и четвороспојивог окружења пиксела.....	52
Слика 4.5.2:	Приказ оцене најкраћег пута за чворове у окружењу применом Менхетн норме.....	52
Слика 4.5.3:	Приказ оцене најкраћег пута за чворове у окружењу применом Еуклидске норме.....	53
Слика 4.5.4:	Додељене вредности цене помераја за окружење чвора.....	54

Слика 4.5.5:	Приказ путање од почетног до циљног чвора са прорачуном.....	55
Слика 4.5.6:	Приказ алгоритма према ком је урађен програм.....	56
Слика 4.5.7:	Графички приказ оптималне путање у софтверу <i>MATLAB</i>	57
Слика 4.6.1:	Симулациони модел без анализе транспортних токова.....	60
Слика 4.6.2:	Симулациони модел у току симулације.....	61
Слика 4.6.3:	Симулациони модел након завршене симулације.....	62
Слика 4.6.4:	Симулациони модел са анализом транспортних токова.....	63
Слика 4.6.5:	Кретање дела 1 кроз погон и удаљености између датих објеката.....	63
Слика 4.6.6:	Кретање дела 2 кроз погон и удаљености између датих објеката.....	63
Слика 4.6.7:	Кретање дела 3 кроз погон и удаљености између датих објеката.....	63
Слика 4.6.8:	Кретање дела 4 кроз погон и удаљености између датих објеката.....	64
Слика 4.6.9:	Симулациони модел са анализом транспортних токова (у току симулације)	64
Слика 4.6.10:	Симулациони модел са анализом транспортних токова (на крају симулације).....	65
Слика 4.6.11:	Layout новог технолошког окружења предузећа „XY“.....	66
Слика 4.6.12:	Кретање дела 1 кроз погон и удаљености између датих објеката (нови распоред).....	67
Слика 4.6.13:	Кретање дела 3 кроз погон и удаљености између датих објеката (нови распоред).....	67
Слика 4.6.14:	Симулациони модел новог решења диспозиционог плана на крају симулације.....	67
Слика 5.1.1:	Положај робота у почетном тренутку.....	72
Слика 5.1.2:	Оптимална путања од почетне тачке до међутачке.....	72
Слика 5.1.3:	Мобилни робот у току кретања између почетне и крајње тачке.....	73
Слика 5.1.4:	Графички приказ сигурности мобилног робота за случај када није довољно сигуран да се налази у израчунатом положају.....	74
Слика 5.1.5:	Целокупна путања мобилног робота према датом примеру	74
Слика 5.1.6:	Мобилни робот у фази кретања од међутачке до крајње тачке.....	75
Слика 5.1.7:	Графички приказ сигурности мобилног робота да се налази у израчунатом положају након извршене локализације.....	75
Слика 5.1.8:	Графички приказ сигурности мобилног робота да се налази у израчунатом положају након завршетка кретања.....	76

Списак табела

Табела 4.3.1:	Скуп обучавајућих парова ВНМ за ротацију робота током кретања у технолошком окружењу.....	29
Табела 4.3.2:	План експеримента тренирања внм за ротацију робота током кретања у технолошком окружењу.....	30
Табела 4.3.3:	Скуп углова за тестирање тренираних ВНМ.....	30
Табела 4.3.4:	Резултат тренирања ВНМ – за ротацију робота током кретања у технолошлком окружењу.....	32
Табела 4.3.5:	План експеримента тренирања ВНМ за детекцију контролних тачака.....	37
Табела 4.3.6:	Резултат тренирања ВНМ за детекцију контролних тачака у радном окружењу мобилног робота.....	38
Табела 4.3.7:	Скуп улазних вредности за проверу изабране внм за детекцију контролних тачака у радном окружењу мобилног робота.....	40
Табела 4.4.1:	Алгоритам Калмановог филтера.....	46
Табела 4.4.2:	Алгоритам линеаризовног Калмановог филтера.....	47
Табела 4.4.3:	Алгоритам линеаризовног Калмановог филтераза корак предикције.....	48
Табела 4.4.4:	Алгоритам линеаризовног Калмановог филтераза корак корекције.....	50
Табела 4.6.1:	Трајање технолошких операција за репрезентативне делове.....	59

Садржај

1. Увод	10
2. Поставка проблема	11
2.1 Опис технолошког окружења	11
2.2 Дефиниције индустријских робота и основне поставке мобилне роботике	13
2.2.1 Мобилни роботи.....	14
2.3 Поделе сензора	16
3. Концепцијско решење пројектног задатка	17
4. Детаљан опис пројектних решења	18
4.1 Формирање конфигурације мобилног робота	18
4.1.1 Опис расположивих компоненти за конфигурисање мобилног робота	18
4.1.2 Фазе формирања конфигурације мобилног робота.....	21
4.1.3 Детаљан опис коришћених сензора	24
4.2 Имплементација модела кретања мобилног робота	26
4.2.1 Модел кретања на основу пређеног пута.....	28
4.3 Развој и имплементација опсервационог (сензорског) модела мобилног робота применом система вештачких неуронских мрежа	29
4.3.1 Основе примењеног модела вештачких неуронских мрежа	30
4.3.2 Вештачка неуронска мрежа за одређивање команде ротације вратила мотора у односу на потребну ротацију робота у глобалном координатном систему	37
4.4 Примена алгоритма Калмановог филтера у циљу одређивања положаја мобилног робота у окружењу.....	43
4.4.1 Основне теоријске поставке Калмановог филтера.....	43
4.4.2 Основне једначине Калмановог филтера.....	44
4.4.3 Алгоритам Калмановог и линеаризованог Калмановог филтера	47
4.4.4 Имплементација линеаризованог Калмановог филтера за потребе одређивања положаја мобилног робота током експлоатације	50
4.5 А* алгоритам	53
4.5.1 Теоријске поставке и уопштена објашњења А* алгоритма за претраживање.....	53
4.5.2. Реализација и имплементација А* алгоритма у моделу кретања мобилног робота LEGO Mindstorms NXT.....	57
4.6 Развој симулационог модела у AnyLogic софтверском окружењу	60
4.6.1 Теоријске поставке симулације.....	60
4.6.2 Симулација рада технолошког система	61
4.6.2.1 Симулациони модел без анализе транспортних токова	62
4.6.2.2 Симулациони модел са анализом транспортних токова	64

4.6.2.3 Пројектовање новог диспозиционог плана технолошког окружења	67
4.6.2.4 Симулација новог диспозиционог плана технолошког окружења.....	69
5. Експериментални резултати реализованог пројектног решења	70
6. Анализа и дискусија експерименталних резултата	77
7. Закључак.....	78
8. Литература.....	79
9. Прилози.....	80

1. Увод

У задатом технолошком окружењу разматра се производна линија за производњу SIVACON електро – ормана. У централном делу производног погона врши се монтажа произведених делова. Задатак пројекта је обезбеђење транспорта репроматеријала до производне линије, полуфабриката (материјала) на самој производној линији и готових делова до дела погона предвиђеног за монтажу. У том технолошком окружењу транспорт се врши употребом виљушкарa. У циљу унапређења продуктивности, смањењем изгубљеног транспортног времена које настаје као последица људског фактора, уместо постојећег транспортног решења разматра се увођење интелигентног мобилног робота. Имплементација мобилног робота у оквиру овог пројекта вршена је у експерименталном технолошком окружењу.

Задатак је решен применом мобилног робота *LEGO Mindstorm NXT* који је управљан функцијама *RWTH Toolbox* – а имплементираних у софтверском пакету *MATLAB*. Кретање робота на основу задатих команди врши се на основу имплементације модела кретања. Прикупљањем информација са сензора и њиховом класификацијом путем вештачких неуронских мрежа остварује се кретање робота од почетне до захтеване координате крајње тачке. Проблем локализације након генерисања путање A^* алгоритмом решен је применом Калмановог филтера.

Изглед *layout* – а производно – монтажног погона као и распоред постојећих машина дати су у другом поглављу. Такође, у циљу што бољег појашњења разлога употребе мобилног робота, дате су дефиниције и поделе мобилних робота и расположивих сензора.

Концепцијско решење проблема дато је у трећем поглављу.

У четвртном поглављу дат је детаљан опис пројектног решења. То подразумева конфигурисање мобилног робота помоћу компоненти које су расположиве у пакету *LEGO Mindstorms NXT* са детаљним описом коришћених сензора. Такође, дате су и основне теоријске поставке за модел кретања мобилног робота, вештачке неуронске мреже, Калманов филтер, алгоритам претраге и симулација рада технолошког окружења.

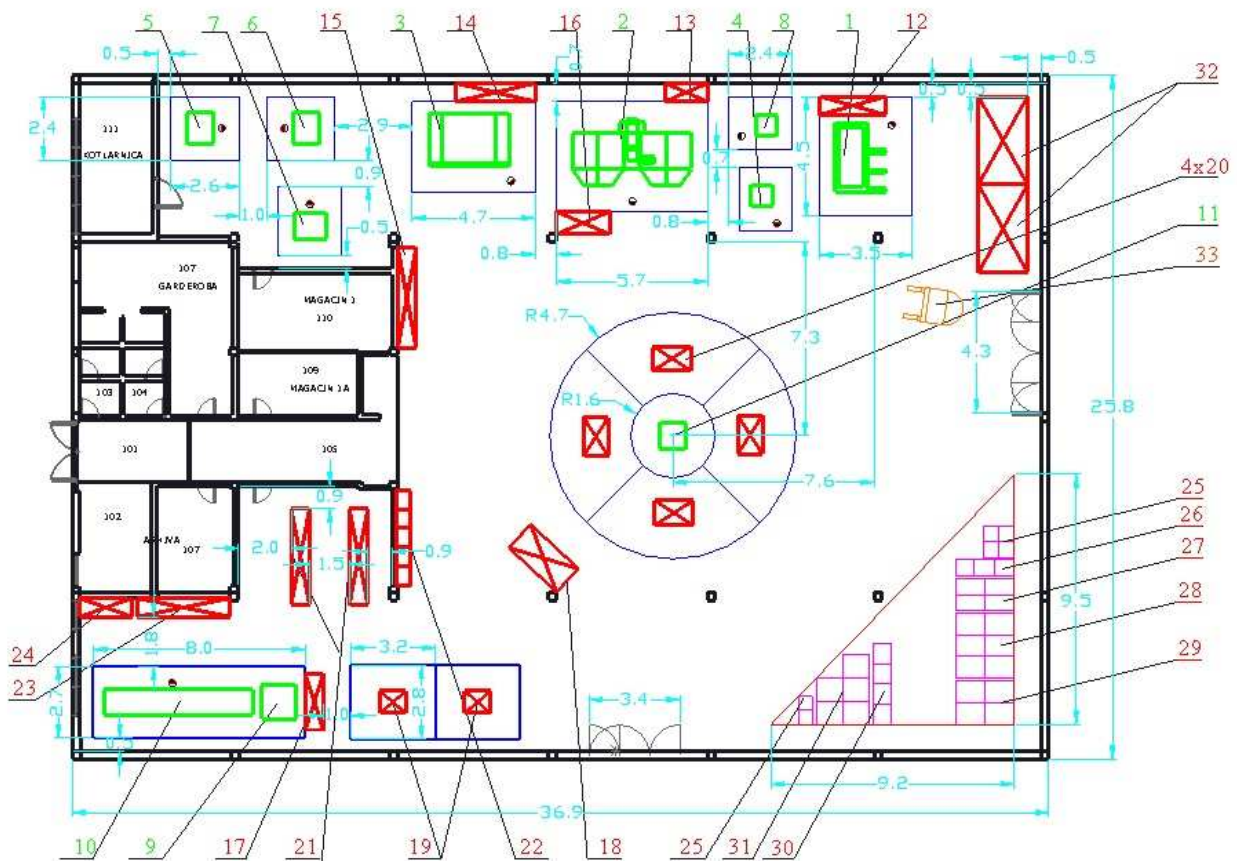
Експериментални резултати дати су у петом поглављу, док су анализа и дискусија, као и проблеми који су постојали приликом реализације пројекта, приказани у шестом поглављу. И на крају самог пројекта, у последњем поглављу, дат је закључак са прегледом постигнутих резултата и могућностима даљег развоја.

У прилогу се налазе сви програми реализовани у окружењу *MATLAB*, симулациони модел токова материјала урађен у софтверском окружењу *AnyLogic*, као и резултати тренирања свих вештачких неуронских мрежа.

2. Поставка проблема

2.1 Опис технолошког окружења

Предузеће „XY“ бави се производњом и монтажом SIVACON електро-ормана. Да би се пројектовао диспозициони план, било је потребно извршити анализу пројектованих технолошких поступака обраде делова, као и анализу дефинисане технологије за монтажу електро-ормана, што доводи до оптимизације распореда машина у погону. У предузећу „XY“ рамештај производне опреме је извршен према редоследу технолошких операција, који подразумева да се машине и уређаји постављају у линијском распореду, где је редослед извођења операција условљен пројектованим технолошким поступком за сваки део који се израђује. На слици 2.1.1 дат је *layout* производно-монтажног погона.



Слика 2.1.1: Layout производно-монтажног погона предузећа „XY“

Производни погон се састоји од следећих физичких средстава:

- Маказе за сечење – позиција 1;
- CNC машина за пробијање и просецање – позиција 2;
- CNC хидраулична „апкант“ преса – позиција 3;
- Машина за исецање профила – позиција 4;
- Стубна бушилица – позиција 5 и 6;
- Кружна тестера – позиција 7;

- Оштрилица – позиција 8;
- Линија за обраду делова од лима – позиција 9;
- Улазно складиште полуфабриката (лима) – позиција 32;
- Међускладиште – позиција 15;
- Складиште готових делова од лима (полупроизводи) – позиција 21;
- Складиште вијчане робе и компонената за монтажу – позиција 22;
- Складиште готових делова од бакра – позиција 23;
- Складиште готових производа (*SIVACON* електро-ормана) – позиције 25 до 31;
- Монтажна целина са четири радна места – позиција 20;
- Монтажна целина са два радна места – позиција 19;
- Радни сто за формирање подсклопова – позиција 18.

Критеријум за распоређивање машина и радних места је пре свега базиран на токовима материјала, што је омогућило смањење транспортних путева унутар погона. Треба имати у виду да је производна хала изграђена пре дефинисања пројектног решења распореда машина.

Са слике 2.1.1 се види да се у линијском распореду налазе машине за обраду челичног лима, које се и најчешће користе у погону, а то су машине: М1(маказе за сечење), М2 (CNC машина за пробијање и просецање), М3 (CNC хидраулична „апкант“ преса) и М4 (машина за исецање профила). Машине М5, М6 (стубне бушилице) и М7 (кружна тестера) су издвојене, јер су у мањој употреби. Машина М8 (Оштрилица алата) је лоцирана уз машину М2 јер се најчешће користи за оштрење алата који су везани за машину М2. Машина М9 (линија за обраду бакра) налази се одвојена од претходне две наведене групе машина, јер је и технолошки процес израде делова од бакра издвојен у односу на остале технолошке процедуре.

Приликом пројектовања складиштеног простора било је потребно водити рачуна о токовима материјала као и о расположивом простору унутар хале. Улазно складиште полуфабриката од лима (позиција 32) лоцирано је одмах код улаза са десне стране, што омогућава лако маневрисање и допремање полуфабриката. Складиште полуфабриката од лима (32), међускладиште (15), складиште готових делова од лима (21) и складиште готових делова од бакра (23) базирани су на типу регалног складишта са полицама. Складиште вијчане робе и компонената за монтажу (22) базирано је на типу сандучастог складишта. Складиште готових производа (*SIVACON* електро-ормана – позиције 25 до 31) је палетног типа и смештено је са леве стране од улаза. Унутар тог простора предвиђено је одлагање различитих врста *SIVACON* електро-ормана, који се разликују пре свега у величини. Види се са слике 2.1.1 да је ово највеће складиште, што је и разумљиво јер се ради о складишту готових производа.

У предузећу „ХУ“ обавља се ручна монтажа ормана према унапред дефинисаној технологији. Склоп ормана се формира на палети. Пошто постоји више различитих типова ормана, постоје два радна простора на којима се врши њихова монтажа: први са четири радна места за веће ормане (позиција 20) и други са два радна места за мање ормане (позиција 19). Поред сваке од ових целина налази се радни сто за формирање подсклопова (позиција 18). Такође се види са слике 2.1.1 да су у близини радног простора за монтажу лоцирана горе наведена складишта: вијчане робе (22), полупроизвода (21) и готових делова од бакра (23), што радницима омогућава брз приступ истим и скраћује време монтаже. Монтажна целина са четири радна места (20) је у кружном распореду што омогућава оптимално искоришћење простора. У средишњем делу ове целине предвиђен је простор за пнеуматски систем монтаже.

Унутрашњи транспорт материјала обавља се помоћу два виљушкар: Дизел виљушкар носивости до 2,5 t и ручног виљушкар носивости до 1,5 t. Постојећим пројектним решењем је предвиђено да се саобраћај унутар погона одвија у једном смеру, с тим што је остављена могућност да се у некој другој фази развоја предузећа уведе и двосмерни саобраћај. У производном делу погона материјал се креће у линијском распореду

од једног до другог радног места према унапред дефинисаном технолошком поступку, док се у монтажном делу погона материјал креће кружно.

Експериментима и симулацијама потребно је анализирати увођење мобилних робота, који би уместо постојећих виљушкара вршили транспорт репроматеријала, сировина и готових делова. У том циљу у наставку се даје основна подела индустријских и мобилних робота, као и подела сензора.

2.2 Дефиниције индустријских робота и основне поставке мобилне роботике

Реч робот потиче од чешке речи „робота“ што значи присиљан рад. Да би робот могао да функционише без непосредне физичке присутности оператера мора бити способан да сам доноси одлуке на путу остварења циља. За то треба имати одређену интелигенцију. Вештачка интелигенција је створена способност сналажења робота у непознатим условима, а све је усмерено ка коначном циљу извршења задатка.

У оквиру овог дела пројекта, као увод у причу везану за мобилне роботе и њихову имплементацију у задато технолошко окружење, наводе се основне дефиниције индустријских робота.

Према [8], робот је репрограмабилни, мултифункционални манипулатор пројектован да помера материјал, делове, алате или специјалне уређаје коришћењем различитих програма при извршавању различитих задатака (Robot Institute of America - RIA).

Према [7], робот је аутоматски контролисан, репрограмабилан, вишенаменски манипулатор, који се може програмирати у три или више оса, статичан или мобилан у аутоматизованим производним процесима (ISO8737).

Према [7], робот је машина за аутоматизовано обављање послова, управљан помоћу рачунара (Cambridge online dictionary).

Све ове дефиниције на својствен начин описују робота. Посебно је занимљива једна реченица Joseph Engelberg-а, којом је он описао појам робота рекавши да не може дефинисати робота, али га може препознати када га види.

Данас је веома тешко дати потпуну дефиницију робота, али се уопштено може рећи да је робот активан вештачки систем чије је окружење физички свет са којим је у сталној интеракцији. У 21. веку се поред свих својстава робота наведених у дефиницијама могу придодати још и атрибути *интелигентан* и *аутономан* [8].

Уопштено, индустријски роботи се могу класификовати према следећим критеријумима:

- намени;
- степену универзалности;
- кинематичким, геометријским и енергетским параметрима;
- методама управљања итд.

Према JARA класификацији (Japanes Robot Assotiation) датој у [7], индустријски роботи се по типу управљања деле на следеће класе:

- ручни манипулациони уређаји (то су уређаји са неколико степени слободе којима управља човек-оператер);

- секвенцијални работи (то су манипулациони уређаји са фиксним и променљивим секвенцијалним управљањем и тешко се програмирају);
- понављајући или Play-back работи (човек као оператор, вођењем или са пулта, извршава задатак водећи робота који меморише трајекторије и касније их понавља);
- нумерички управљани работи (програмирају се текстуалним језицима);
- интелигентни работи (на бази информација са сензора и техникама вештачке интелигенције разумеју околинду и задатак и доносе одлуке у реалном времену).

Понекад се работи деле и на генерације:

- прва генерација (програмски работи, класе 1 до 4 по JARA);
- друга генерација (адаптивни работи, класе 5 по JARA);
- трећа генерација (интелигентни работи, класе 5 по JARA и работи који су у развоју).

Оно што треба истаћи је да ове генерације не наслеђују једна другу као што је случај код рачунара, већ су све присутне и биће присутне у будућности зависно од задатака који се постављају пред индустријски робот.

Са становишта функције, индустријске роботе је могуће класификовати као роботе за:

- манипулацију;
- обављање процеса;
- манипулацију и обављање процеса;
- специјалне задатке.

Са становишта примене, индустријске роботе је могуће класификовати као роботе за:

- опслуживање машине;
- палетизацију и депалетизацију;
- заваривање;
- обраду резањем;
- сечење ласером, плазмом, воденим млазом;
- роботе за монтажу;
- роботе за мерење и контролу итд.

Како су мобилни работи посебна класа индустријских робота у наставку ће бити дате њихове основне карактеристике и подела.

2.2.1 Мобилни работи

Једна од дефиниција за мобилне роботе је и следећа:

Мобилни робот је мобилан и манипулативан физички систем, који се аутономно креће кроз непознато окружење, остварујући притом интеракцију са људским бићима или аутономно обављајући неки посао уместо њих.

Једна од класификација мобилних робота може се извршити с обзиром на број додатних степени слободе и то на оне са:

- једним степеном слободе (врши линеарно кретање тзв. линеарна мобилност);
- два степена слободе (врше кретање у равни тзв. равански работи);
- три степена слободе (врше кретање у простору тзв. просторна мобилност).

Даље, мобилни роботи се могу класификовати и према уређајима помоћу којих остварују кретање и то:

- кретање помоћу ногу тзв. "legged" робота , који још могу бити хуманоидни и анималоидни;
- котрљање помоћу точкова, гусеница;
- кретање опо шинама.

Следећа подела остварена је на основу окружења у којима се мобилни роботи крећу и извршавају своје задатке.

- роботи који се крећу по подлози;
- роботи који егзистирају у ваздуху;
- роботи који егзистирају у воденом окружењу.

Постоји више критеријума на основу којих се сензори класификацију у зависности од области у оквиру које се разматрају. Од посебног значаја је и подела сензора на:

- унутрашње, који контролишу унутрашње стање робота одређено његовим унутрашњим величинама;
- спољашње, који генеришу мерне сигнале о спољашњим величинама из окружења робота.

Оправданост примене и коришћења мобилних робота:

- Већа расположивост робота, будући да немају прописано радно време
- Манипулација већим теретима и бржи рад у односу на човека(резање, фарбање, брже кретање)
- Смањење употребе скувих ресурса
- Повећање квалитета производа
- Обављање послова који су за човека заморни, досадни...

Примена мобилних робота:

- Медицинске услуге (синтеза биолошког и механичког састава је киборг)
- Опасни послови и додир са струјом
- Рударство
- Грађевински послови
- Пољопривреда
- Подводни свет
- Свемир
- Војне сврхе
- Превоз, складиштење, руковање материјалом

2.3 Поделе сензора

Што се тиче основне поделе сензора, они се могу поделити у односу на физичке величине које је потребно идентификовати. Према [4], то су:

- унутрашњи сензори или сензори који мере величине унутар самог мобилног робота, као што су број обртаја електро мотора или енергетске резерве;
- спољашњи сензори или сензори који мере величине у односу на окружење у којем се мобилни робот креће, на пример, то су тактилни сензори.

Такође, сензори се могу поделити у односу на енергетски утицај окружења и обрнуто. Према [4] су то:

- пасивни сензори или сензори који врше мерење енергетског утицаја окружења на систем мобилног робота, као што су микрофони, камере итд.;
- активни сензори или сензори који емитују енергију у окружењу и врше мерење одговора окружења на енергетски стимуланс, као што су ласерски и ултразвучни сензори.

3. Концепцијско решење пројектног задатка

Решавање пројектног задатка се састоји из више целина. Основна идеја је да се реализација пројекта врши у више фаза и да се при томе свака целина детаљно разрађује. Поступном имплементацијом сваког разрађеног дела пројектног задатка извршава се интеграција, чиме добијамо пројектно решење у целини. Претходно дефинисана идеја за решење овог пројектног задатка приказана је помоћу блокова алгоритма датог на слици 3.1.



Слика 3.1: Алгоритам концепцијског решења пројектног задатка

Детаљи концепцијских решења који обухватају модел кретања, вештачке неуронске мреже, Калманов филтер, A* алгоритам и симулацију рада технолошког система у софтверу *AnyLogic*, са реализацијом, дати су у четвртом поглављу пројекта.

4. Детаљан опис пројектних решења

4.1 Формирање конфигурације мобилног робота

4.1.1 Опис расположивих компоненти за конфигурисање мобилног робота

На почетку пројекта неопходно је било саставити конфигурацију *Lego Mindstorms NXT* мобилног робота. Поменути комплет садржи компоненте од којих је могуће изградити различита решења, па се у наставку даје опис расположивих компоненти.

Уопштено говорећи, *Lego Mindstorms NXT* робот мора да има процесор, који обрађује информације, сензоре који скупљају информације из околине у којој се налази робот, делове помоћу којих се врши кретање робота и извор енергије који неопходан за напајање компоненти робота.

За конкретне потребе овог пројектног задатка коришћени су следећи елементи:

- управљачка јединица са интерфејсима за моторе, сензоре и везу са рачунаром;
- сензор (светлосни - *light sesor*);
- погонски модул (чине га два мотора);
- носећа конструкција са точковима;
- извор енергије за напајање (батерије);

На слици 4.1.1 приказан је серво мотор који служи као актуатор за покретање робота. У сваки мотор уграђен је сензор (енкодер) помоћу кога се добија угаона вредност закретања мотора, одређује смер и брзина обртања. Пренос обртног момента од вратила мотора до излазног вратила врши се помоћу зупчастих преносника укупног преносног односа $i = 48$.



Слика 4.1.1: Серво мотор из комплета *Lego Mindstorms NXT*

На слици 4.1.2 приказан је сензор додира који омогућава роботу да региструје присуство предмета из свог окружења. Сензор додира код *Lego Mindstorms NXT* робота је у ствари један прекидач који има само два стања: притиснут (када робот додирне неку препреку) и отпуштен (када је сензор не додирује).



Слика 4.1.2: Сензор додира (*Touch sensor*)

На слици 4.1.3 приказан је сензор звука. То је уређај којим робот може да региструје и реагује на звук. Звучни сензор региструје звук тј. промене ваздушног притиска које су фреквенције до 20KHz. Микрофон детектује јачину звука у децибелима (dB) или у прилагођеним децибелима (dBA).



Слика 4.1.3: Сензор звука (*Sound sensor*)

На слици 4.1.4 приказан изглед светлосног сензора. На овај сензор посебно треба обратити пажњу. Само он је конкретно од свих наведених сензора, поред енкодера који је саставни део мотора, коришћен у пројектном задатку. Овај сензор омогућава роботу да детектује светлост и боју. Претходне генерације ових сензори имали су да распознају само нијансе сивог, а данас постоје сложенији сензори који могу да распознају и боје. Једноставна *Web* камера може да буде светлосни сензор за који је потребан процесор и софтвер да обради низ слика које снима камера и да из њих извуче баш оне информације које су роботу потребне. Ово уопште није једноставан задатак. Сналажење робота у простору на основу низа слика које у себи имају много тачака је тешко, за разлику од обраде информација која стиже са сензора додира који има само два стабилна стања (активан или неактиван). Задатак светлосног сензора у овом пројекту је био да препозна црну и белу боју. Светлосни сензор може да ради на два начина: у активном и неактивном моду. У активном начину рада активира се усмерени сноп светлости на самом сензору (*LED*), а пријемни сензор детектује количину рефлектоване светлости од неког објекта. У неактивном начину рада, извор светлости на самом сензору не ради, већ пријемни сензор детектује интензитет обојених површина на основу амбијенталног осветљења.



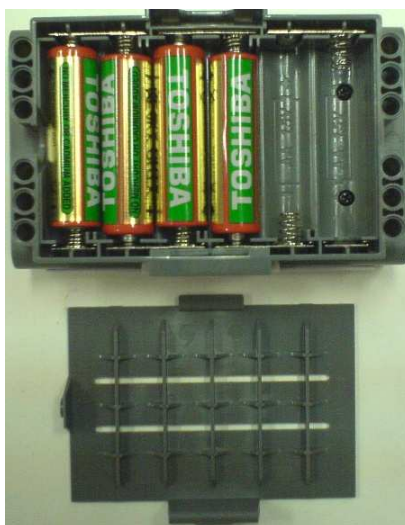
Слика 4.1.4: Светлосни сензор (*Light sensor*)

Слика 4.1.5 приказује ултразвучни сензор, уређај који мери растојање до неког предмета и реагује на покрет. Овај сензор у пракси обично има, осим пријемника ултразвука (специјални микрофон), и предајник ултразвука што није прави део сензора. Он ништа не региструје, већ само шаље ултразвучни талас, који се после одбијања о препреку у околини робота враћа ка роботу и њу прима пријемник који је прави сензор. Мерећи време које протекне од када је емитован кратак ултразвучни талас до тренутка када је, након одбијања од препреке, стигао до пријемника може се израчунати растојање до препреке.



Слика 4.1.5: Ултразвучни сензор (*Ultrasonic sensor*)

Сви делови робота захтевају напајање енергијом да би радили. Роботи су још увек електрично-механичке машине, тј. напајају се електричном енергијом која је акумулирана у хемијску енергију у батеријама. Кутија у коју се смешта извор напајања *Lego Mindstorms NXT* робота приказана је на слици 4.1.6.

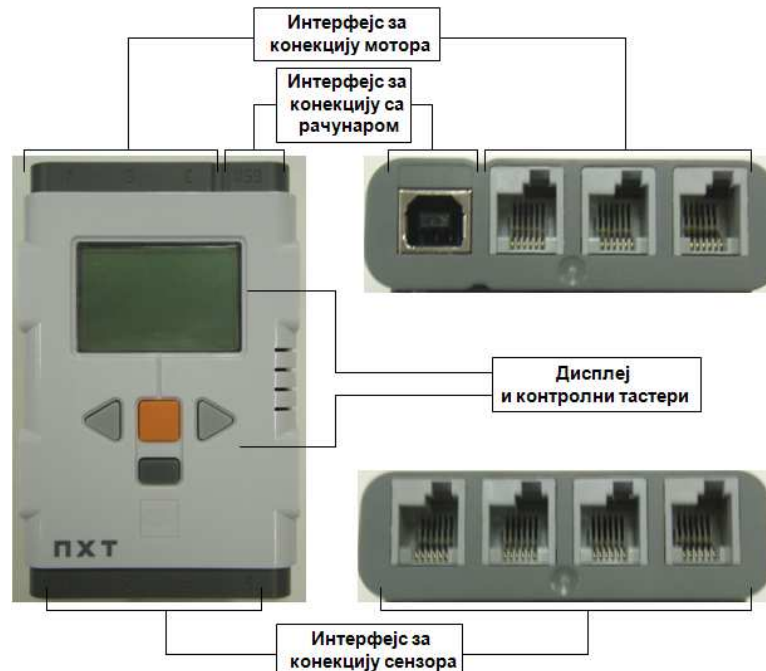


Слика 4.1.6: Кутија са батеријама мобилног робота

4.1.2 Фазе формирања конфигурације мобилног робота

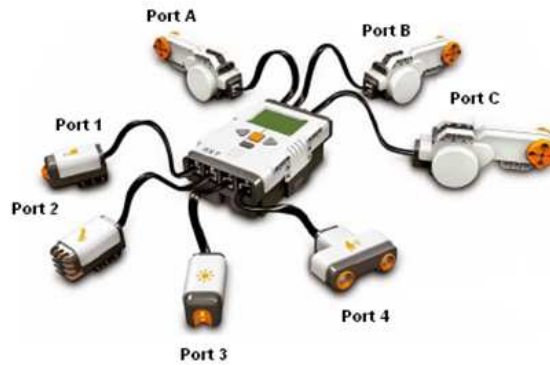
Начин на који се покреће робот може бити различит, а конкретно у овом случају је помоћу точкава. Елементи помоћу којих можемо физички реализовати *Lego Mindstorms NXT* робота да буде функционалан су *Lego Tehnics* “коцкице” помоћу којих се може конфигурисати робот добре функционалности, издржљивости и покретљивости.

На слици 4.1.7 дата је управљачка јединица или процесор *NXT*. Он прима информације од сензора и на основу стања у околини и управљачких команди одлучује о томе шта ће урадити. Као и сви други рачунарски тј. компјутерски склопови и процесор робота доноси врло сложене одлуке на основу обраде информација које он види као низове јединица и нула. То не значи да програмер мора данас да пише низове јединица и нула да би програмирао робота, јер су направљени програмски језици који штеде време у писању. Команде написане у развојном софтверу се аутоматски у процесору преводне на команде машинског језика које представљају низ јединица и нула.



Слика 4.1.7: Изглед управљачке јединице са интерфејсима

Да би робот радио исправно, сензори и мотори треба да буду повезани на одговарајуће улазне и излазне портове. На слици 4.1.8 су приказани улазно-излазни портови робота.



Слика 4.1.8: Приказ улазно-излазних портова *Lego Mindstorms NXT* робота

Улазни портови:

- | | | |
|--------------|---|--|
| PORT1 | - | Сензор додира (<i>Touch sensor</i>); |
| PORT2 | - | Звучни сензор (<i>Sound sensor</i>); |
| PORT3 | - | Светлосни сензор (<i>Light sensor</i>); |
| PORT4 | - | Ултразвучни сензор (<i>Ultrasonic sensor</i>). |

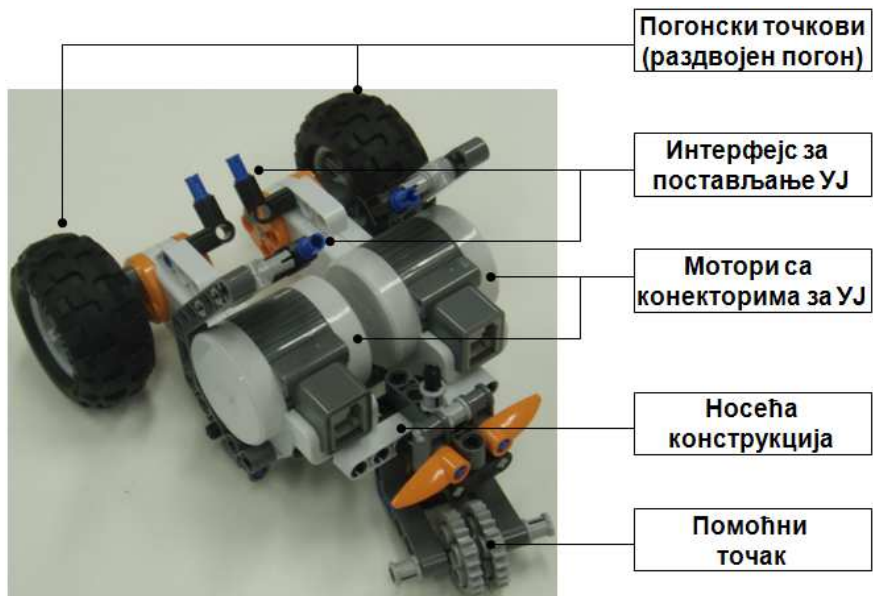
Излазни портови:

- | | | |
|---------------|---|--|
| PORT A | - | Мотор који се користи за додатне функције; |
| PORT B | - | Мотор за кретање; |
| PORT C | - | Мотор за кретање. |

У суштини, ако се користи само један сензор, а не сва четири као на слици 4.1.10, што је конкретно у овом пројекту био случај, он се може повезати на било који улазни порт 1,2,3 или 4. Што се тиче мотора, за овај пројекат су коришћена два мотора. Они се такође могу повезати на било који излазни порт, али приликом слања команди моторима, односно приликом примања информација са сензора, физички искоришћени портови морају бити дефинисани и у управљачком софтверу.

Управљачки софтвер коришћен приликом реализације пројектног решења је *MATLAB*. Функције које се користе за управљање мобилног робота су саставни део *RWTH Toolbox* – а. Помоћу ових функција се остварује директна комуникација са управљачком јединицом *LEGO* робота путем *USB* или *Bluetooth* комуникацијских протокола. *RWTH Toolbox* је развијен у оквиру студентског пројекта на универзитету у Ахену, Немачка.

На слици 4.1.9 приказана је прва фаза формирања конфигурације мобилног робота. На носећу конструкцију са помоћним ослоним тачкама постављени су погонски точкови, а након тога и мотори. Пошто сваки од мотора независно покреће по један точак, може се закључити да је реч о диференцијалном погону. Интерфејс за постављање управљачке јединице такође се види на слици 4.1.9.



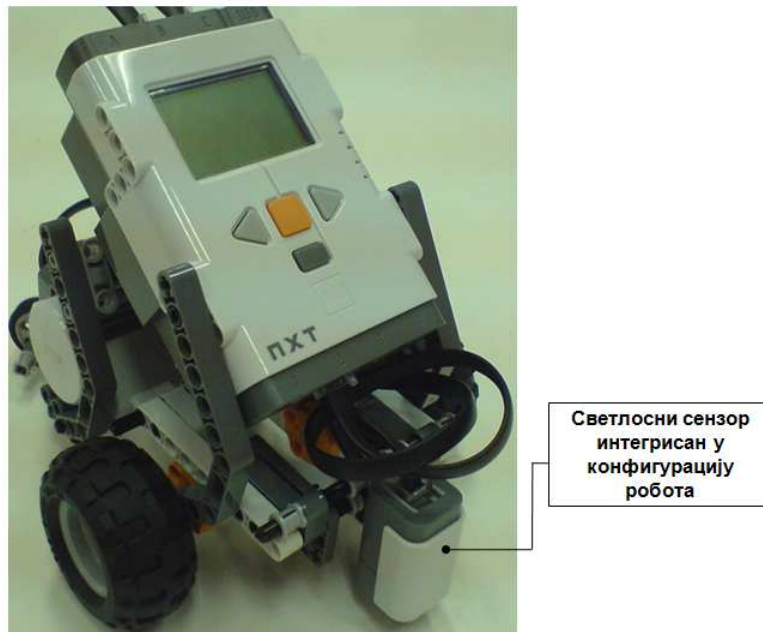
Слика 4.1.9: Прва фаза конфигурисања мобилног робота

Централно место у конфигурацији робота заузима процесорска *NXT* јединица. На слици 4.1.10 приказана је друга фаза конфигурисања мобилног робота, тачније постављање управљачке јединице *NXT* и повезивање са моторима.



Слика 4.1.10: Друга фаза конфигурисања мобилног робота

У трећој фази формирана је коначна конфигурација мобилног робота додавањем светлосног сензора као што се види на слици 4.1.11.



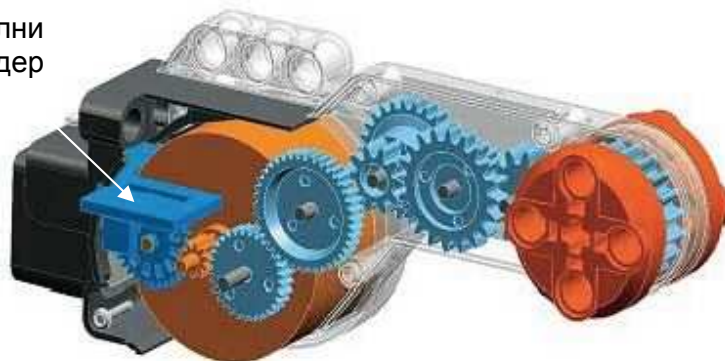
Слика 4.1.11: Изглед коначне конфигурације мобилног робота

4.1.3 Детаљан опис коришћених сензора

Сензори које садржи конфигурисани *LEGO Mindstorms NXT* мобилни робот су енкодер мотора и светлосни сензор. Принципи рада и детаљне карактеристике коришћених сензора дате су у тексту који следи.

Енкодери представљају класу дигиталних сензорских система који се користе за мерења линијских и угаоних помераја [10]. У већини случајева изводе се коришћењем оптичких и магнетних електромеханичких претварача и могу се класификовати у две групе: апсолутни енкодери и инкрементални енкодер. Апсолутни енкодери мере апсолутну позицију, која је дефинисана конструктивним решењем склопа у оквиру кога функционишу, а инкрементални енкодери мере релативни положај у односу на неку унапред познату иницијалну координату (угаону или линијску)[10]. Према спецификацији, *NXT LEGO* серво мотор садржи у себи имплементиран инкрементални обртни оптички енкодер, као што је приказано на 4.1.12.

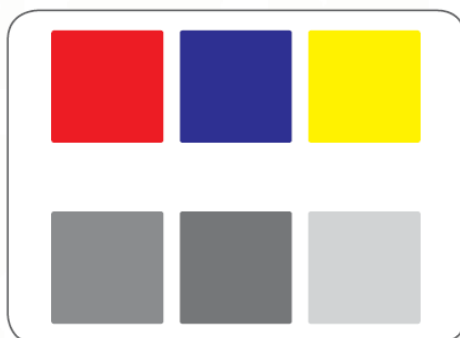
Инкрементални енкодер



Слика 4.1.12: Приказ отвореног кућишта мотора са позицијом инкременталног енкодера

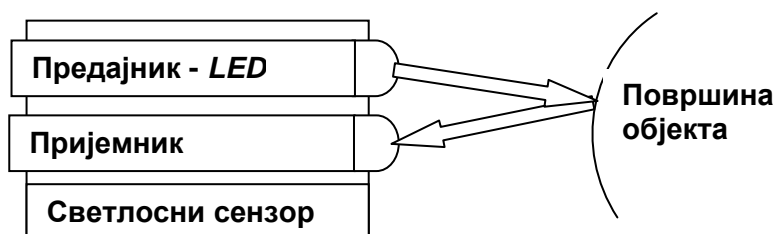
Инкрементални енкодер ради на принципу генерисања поворке импулса чијим се бројањем израчунава угаони померај па се на основу тога може назвати и бројачки енкодер [10]. Овај енкодер спада у групу оптичких енкодера који поседују оптичке елементе, где спадају фото диоде и фото транзистори, од којих је један извор а други пријемник. Поред оптичких елемената енкодер садржи и маску на којој се налазе прозирна и непрозирна поља. Величина ових поља је у директној сразмери са механичком резолуцијом енкодера. Приликом обртања оптичког енкодера поља маске наизменично прекидају оптички пут између извора и пријемника. Прекид оптичког пута региструје фотоелектрични сензор који тада генерише сигнал који се након обраде уобличава и претвара у дискретну форму погодну за прорачун угаоног помераја. Резолуција инкременталног енкодера који се користи код *NXT LEGO* серво мотора је 1° .

Светлосни сензор је класа сензора чији је задатак да препознају интензитет рефлексије светлости неке, задате или обојене, површине. Зона осетљивости диктира конструкцију односно тачност сензора па тако за осетљивије имамо комплекснију конструкцију, квалитетнији одзив и већу тачност сензора и обрнуто за оне са мањом зоном осетљивошћу. Због своје несавршености ови сензори су подложни утицају шума из околине као и слабој осетљивости на објекте са лошом рефлексијом. Препознавање боја помоћу овог типа сензора манифестује се различитим нијансама сиве боје, како је то приказано на слици 4.1.13, односно нијансним прелазом између црне и беле.



Слика 4.1.13: Приказ препознавања боја светлосним сензором

Оптички сензор се састоји од предајника (емитора) који најчешће чини ласерска диода или у овом конкретном случају *LED* диода као извор светлости и пријемника. Светлосни зрак који емитује предајник (*LED* диода) пада на површину у коју је сензор уперен, одбија се и враћа на пријемник, обично фототранзистор, како је приказано на слици 4.1.14, у коме се примљена светлосна побуда преводи у електрични сигнал који се даље шаље на обраду.



Слика 4.1.14: Упрошћени шематски приказ светлосног сензора

Како је већ поменуто светлосни сензор има два начина рада и то активан и неактиван начин рада. У активном начину ради пали се усмерени зрак светлости на самом сензору (*LED*), а пријемни сензор детектује количину рефлектоване светлости од неког објекта, док у неактивном начину рада, извор светлости на самом сензору не ради, већ пријемни сензор

детектује интензитет обојених површина. Случај активног рада је искоришћен у пројекту на начин како ће већ бити описано у наставку.

У задато технолошко окружење, на основу теоријских знања стечених на предавањима из предмета Интелигентни технолошки системи, потребно је имплементирати интелигентног мобилног агента, који ће извршавати опслуживање машина и обављати унутрашњи транспорт материјала о чему ће у наставку бити више речи.

4.2 Имплементација модела кретања мобилног робота

Да би се успоставио унутрашњи транспорт материјала, сировина и готових делова у оквиру дефинисаног технолошког окружења, што је и циљ овог пројектног задатка, мора се најпре извршити математичко моделирање овог система. Тај математички модел треба да омогући познавање позиције и оријентације мобилног робота у сваком временском тренутку.

Вектор стања система се означава са x_t и дат је једначином:

$$x_t = (x, y, z, \theta, \psi, \varphi) \quad (1)$$

У једначини (1) x , y , z представљају компоненте вектора стања, које дефинишу позицију мобилног робота. Детаљније објашњење ових компоненти је следеће:

- x - позиција мобилног робота у координатном систему дуж осе x ;
- y - позиција мобилног робота у координатном систему дуж осе y ;
- z - позиција мобилног робота у координатном систему дуж осе z ;

Такође, у једначини (1) фигуришу и Ојлерови углови θ , ψ , φ , који одређују оријентацију мобилног робота, а прецизније објашњење ових углова је следеће:

- θ - пропињање, тј. обртање око y осе;
- ψ - скретање, тј. обртање око x осе;
- φ - ваљање, тј. обртање око z осе;

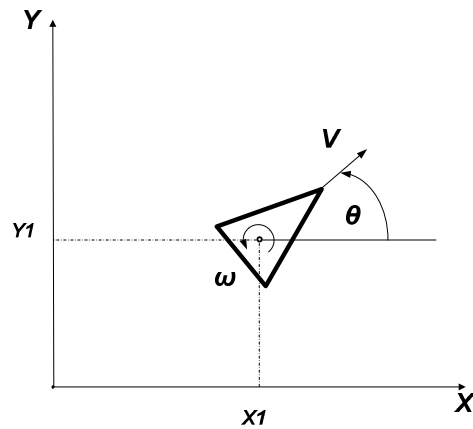
Како би се мобилни робот оспособио да врши задаке који се од њега захтевају, он мора у сваком тренутку да зна где се налази, тачније своју позицију и оријентацију. Да би робот знао своју позицију и оријентацију, неопходно је развити математички модел.

Пошто су компоненте вектора стања система наведене у претходном делу текста, треба дати објашњење шта, конкретно, позиција и оријентација подразумевају и како се оне тачно дефинишу. Сходно томе, позицијом се могу одредити координате тачака у Декартовом координатном систему, а оријентација се одређује углом који заклапа оса робота у односу на осу x координатног система.

Робот који је коришћен у овом пројектном задатку поседује три степена слободе, а као што је познато из теорије, од броја степени слободе зависи и број компоненти вектора стања система. Зато је за раванско кретање вектор стања изражен као:

$$x_t = (x, y, \theta) \quad (2)$$

Вектор стања изражен једначином (2) говори о томе да се мобилни робот може кретати само у x и y правцу, тачније у равни x - y , и да се може окретати за угао око осе вертикалне-ортогоналне на раван кретања. Позиција и оријентација мобилног робота у Декартовом координатном систему приказани су на слици 4.2.1.

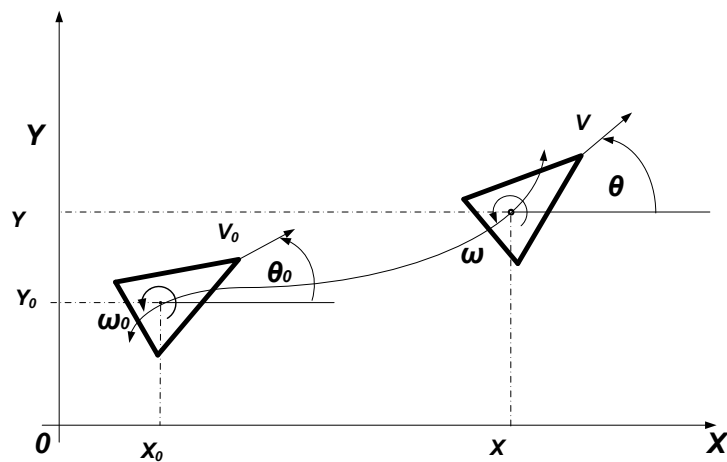


Слика 4.2.1: Позиција и оријентација мобилног робота у равни x - y

Модели кретања који се могу користити и имплементирати за одређивање положаја мобилног робота су:

- модел кретања на основу брзине;
- модел кретања на основу пређеног пута.

Брзински модел кретања базира се промени брзине центра маса и угаоне брзине (velocity based motion model). У овом моделу транслаторна и угаона брзина мобилног робота представљају вектор управљања $u(t)$, која одређују промену позиције и оријентације током експлоатације.



Слика 4.2.2: Брзински модел кретања мобилног робота у равни x - y

За потребе овог пројектног задатка изабран је модел кретања на основу пређеног пута (odometry). Овим моделом управљање $u(t)$ је дефинисано пређеним путем тачкова мобилног робота током тачно дефинисаног временског интервала.

4.2.1 Модел кретања на основу пређеног пута

Модел кретања на основу пређеног пута представља модел помоћу кога се може вршити управљање мобилног робота на основу податка о пређеном путу тачкова.

Математичка формулација модела кретања на основу пређеног пута је следећа:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta / 2) \\ \Delta s \sin(\theta + \Delta\theta / 2) \\ \Delta\theta \end{bmatrix} \quad (3)$$

Ако се узме у обзир да је прираштај лучне координате:

$$\Delta s = \frac{\Delta s_d + \Delta s_l}{2} \quad (4)$$

и прираштај угла $\Delta\theta$:

$$\Delta\theta = \frac{\Delta s_d - \Delta s_l}{2b} \quad (5)$$

онда се једначине кретања могу представити на следећи начин:

$$x' = x + \frac{\Delta s_d + \Delta s_l}{2} \cdot \cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \quad (6)$$

$$y' = y + \frac{\Delta s_d + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \quad (7)$$

$$\theta' = \theta + \frac{\Delta s_d - \Delta s_l}{2b} \quad (8)$$

4.3 Развој и имплементација опсервационог (сензорског) модела мобилног робота применом система вештачких неуронских мрежа

Како је основни задатак пројекта да се начин транспорта у разматраном технолошком окружењу од конвенционалног решења, које се огледа у примени ручног и електричног виљушкарa, замени применом интелигентног мобилног робота, неопходно је обезбедити интеракцију мобилног робота са његовим окружењем. У конкретном случају, на нивоу на коме је вршена имплементација мобилног робота у технолошко окружење (кретање робота у оквиру окружења), неопходно је обезбедити одговарајућу позицију и оријентацију робота током кретања задатом путањом, што се постиже командама за погон мотора. Међутим, пошто се ради о погону (моторима) који због своје несавршености задате команде у општем случају не остварују у потпуности, потребно је имплементирати одговарајући сензорски систем како би се грешка, која се приликом кретања прави, компензовала. Сензорски систем, према потребама у конкретном случају, састоји се од енкодера и светлосног

сензора. Енкодери мере остварену угаону позицију вратила мотора, а светлосни сензор врши детекцију контролних тачака у радном окружењу.

Услед несавршености погонског и сензорског система који су осетљиви на промене из њихове околине (температура, амбијентално осветљење), извршавање команди односно мерење неке величине (остварен угаони померај вратила мотора или измерени ниво светлости одбијене од подлоге) је често различито и за случајеве када је задата команда или објекат мерења номинално исти. Сходно овоме у систем је неопходно уградити вештачку интелигенцију која ће обезбедити да се непоуздана сензорска мерења пресликају у поуздане информације које ће представљати основу за наредну акцију мобилног робота. Поменута интелигенција у систем је уграђена применом вештачких неуронских мрежа.

4.3.1 Основе примењеног модела вештачких неуронских мрежа

Уопштено посматрано, развој интелигентних технолошких система одвијао се уз развој вештачке интелигенције. Тако да данашњи тренд развоја интелигентних технолошких система помера тежиште ка адаптивном процесирању информација [9], чији највиши ниво представљају вештачке неуронске мреже. У основи свих интелигентних формализованих методологија се налази мања или већа способност система да учи [9].

Када при моделирању и управљању системом, због његове комплексности није могуће успоставити општи, аналитички модел, тада се на основу анализе процеса користе неуронске мреже као универзални апроксиматори. Оне се могу користити и као софистицирани сензори, са задатком да изврше естимацију вредности сигнала или променљивих у процесу, уз пажљив избор скупа стабилних, квалитетних нумеричких података који верно описују сам процес [9]. Неуронске мреже имају способност адаптивног поншања према променама, кроз учење улазног узорка, што значи да могу да уче пресликавања између улазног и излазног простора и да при томе синтетизују асоцијативну меморију, која омогућава налажење одговарајућег излаза [9].

У општем случају, неуронска мрежа или конективни модел има три основне компоненте: неурон, топологију мреже и алгоритам учења. Додатне компоненте су: величина мреже (број нивоа-слојева, број неурона у слоју), функционалност неурона (улазни оператор неурона $f(w,x)$, функција преноса $h(s)$, активациона функција $f_i(\text{net}_i)$), обучавање/валидност (величина обучавајућег узорка, формат података, итд.), имплементација/реализација (аналогна, дискретна, софтверска, итд.) [9].

Најраспрострањенији развијени модели вештачких неуронских мрежа, према [9], су:

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

Од наведених модела овде је, при развоју пројекта, коришћена *Backpropagation* (BP) неуронска мрежа. BP неуронска мрежа, за разлику од *Perceptron*-а, има могућност да реши проблем нелинеарног пресликавања из улазног простора у излазни простор. BP је, како је дефинисано у [9], неуронска мрежа са простирањем сигнала унапред, која реализује супервизорски вид учења (улазна и излазна стања се могу одредити у било којој ситуацији јер су успостављене детерминисане релације између њих), са различитим активационим функцијама и алгоритмима учења. BP мрежа користи градијентни поступак при обучавању. Затим се примењује генералисано делта правило да би се утврдила грешка на излазу, коју

простирањем у назад преко скривеног слоја, мрежа користи за полако модификовање сваког тежинског односа, што се понавља за сваки нови узорак. Генерализовано делта правило обезбеђује конвергенцију процеса учења до задатог нивоа тачности кроз итеративни процес адаптације тежинских односа, чиме се остварују основне карактеристике мреже, а то је генерализација и нелинеарно раздвајање.

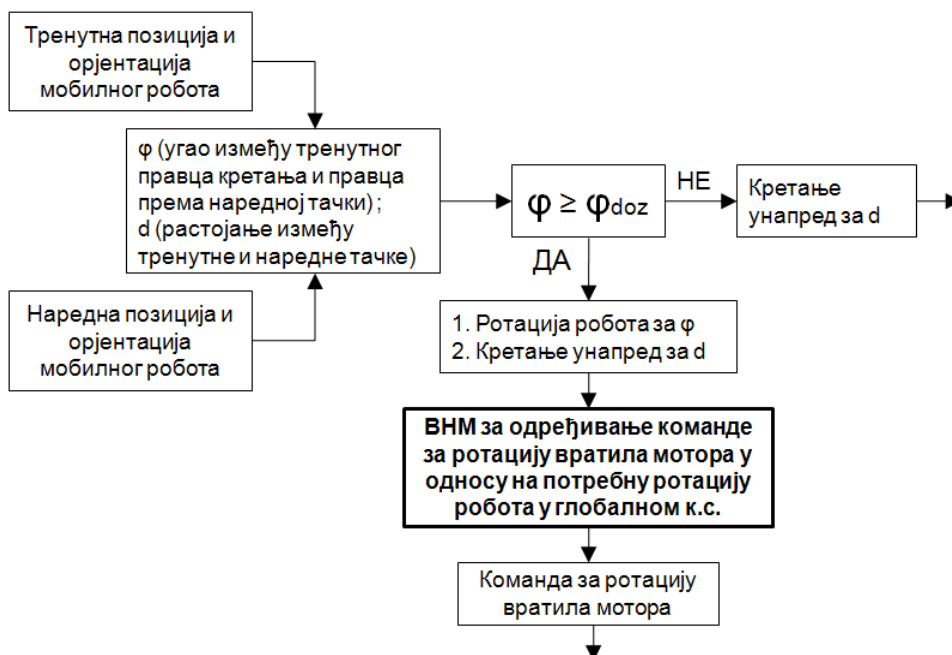
Елементарна архитектура ВР мреже има три слоја који су потпуно повезани [9]. Међутим оптималан број скривених слојева, као и неурона, није унапред одређен, већ зависи од конкретне области примене и проблема који се решава. За ВР мрежу је карактеристично да може да поседује и екстра додатни улаз (неурон који је увек активан и има излазно стање 1 – константна активација), познат као *bias*. Он је повезан са свим неуронима у скривеном и излазном слоју. *Bias* неурон обезбеђује константан члан у тежинским сумама неурона наредног слоја, што резултира побољшањем конвергентне карактеристике ВР мреже [9].

Поступак обучавања ВР неуронске мреже, при коришћењу генерализованог 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 [9]. Оптимална вредност параметра учења зависи од конкретног проблема који се решава и одређује се експерименталним путем.

Развој вештачких неуронских мрежа за реализацију пројектног задатка извршен је у софтверском окружењу *MATLAB – Neural Network Toolbox*, применом модела ВР вештачке неуронске мреже којом софтвер располаже, под називом *newff*.

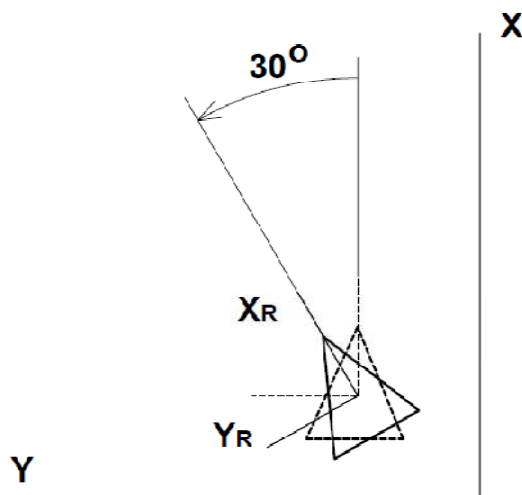
4.3.2 Вештачка неуронска мрежа за одређивање команде ротације вратила мотора у односу на потребну ротацију робота у глобалном координатном систему

Како су једни од основних параметара при развоју ВНМ подаци који се доводе на улаз, као и излази који представљају резултат симулације ВНМ, на слици 4.3.1 се приказује положај развијене неуронске мреже у току информација кроз управљачки систем.



Слика 4.3.1: Ток информација кроз развијени неуронски модел за одређивање команде за ротацију вратила мотора, у реалном времену

Прва фаза у реализацији вештачке неуронске мреже за потребе одређивања команде за потребну ротацију вратила мотора, како би се остварила жељена ротација (орјентација) мобилног робота у глобалном координатном систему, је формирање скупа обучавајућих парова. Овај скуп је формиран у експерименталном поступку задавања команди за кретање робота (у овом случају ротација робота у месту, слика 4.3.2), у циљу одређивања вредности које је потребно задати како би се робот ротирао за жељени угао. Скуп обучавајућих парова дат је у табели 4.3.1.



Слика 4.3.2: Пример ротације робота у месту за 30°

Табела 4.3.1: Скуп обучавајућих парова ВМ за ротацију робота током кретања у технолошком окружењу												
Редни број обучавајућег пара	1	2	3	4	5	6	7	8	9	10	11	12
Улаз (угао ротације робота)	15	-15	30	-30	45	-45	60	-60	75	-75	90	-90
Изназ (команда за ротацију мотора)	40	50	70	70	100	100	130	160	180	180	210	220
Редни број обучавајућег пара	13	14	15	16	17	18	19	20	21	22	23	24
Улаз (угао ротације робота)	105	-105	120	-120	135	-135	150	-150	165	-165	180	-180
Изназ (команда за ротацију мотора)	240	260	250	300	300	350	320	390	350	420	400	450

Анализирајући вредности дате у табели 4.3.1 може уочити да пораст задате вредности (команде) за ротацију вратила мотора није пропорционалан порасту ротације робота која се том приликом остварује. Такође, разликују се и апсолутне вредности које је потребно задати да би се робот ротирао за исти угао у позитивном и негативном математичком смеру. Применом вештачких неуронских мрежа ове нелинеарности би требало да буду превазиђене.

Како би се у поступку развоја, пре имплементације, дошло до вештачке неуронске мреже која ће давати што боље резултате у току коришћења (одговарајуће команде за ротацију мотора у реалном времену) одлучено је да се процес обучавања спроведе за више различитих мрежа. Величине које су овом приликом вариране су број скривених слојева и број неурона у скривеним слојевима, као и параметар учења. Да би постојао одређени ред у варирању ових параметара установљен је план експеримента који је дат у табели 4.3.2.

Табела 4.3.2: План експеримента обучавања ВНМ за ротацију робота током кретања у технолошком окружењу				
Број скривених слојева	Број неурона у првом скривеном слоју	Број неурона у другом скривеном слоју	Параметар учења	Редни број експеримента
1	10	-	0.05	1
			0.3	2
			0.7	3
1	16	-	0.05	4
			0.3	5
			0.7	6
2	8	6	0.05	7
			0.3	8
			0.7	9
2	12	6	0.05	10
			0.3	11
			0.7	12

Поред величина које се у овом случају варирају, у оквиру *Neural Network Toolbox*-а *MATLAB*-а, дефинишу се још неки параметри који се овом приликом нису мењали. Тако се приликом дефинисања величине ВНМ, дефинишу још активационе функције за сваки од скривених слојева. У овом случају то је сигмоидна активациона функција (*tansig*). Затим се још дефинише примена Левенберг-Маркеоовог алгорита учења (*trainlm*), као и поступак минималног градијента (*learngdm*).

Такође, *MATLAB*-ов *Neural Network Toolbox* омогућује дефинисање одређеног скупа параметара који се односе на сам процес обучавања и овом приликом нису вариран. То су следећи параметри:

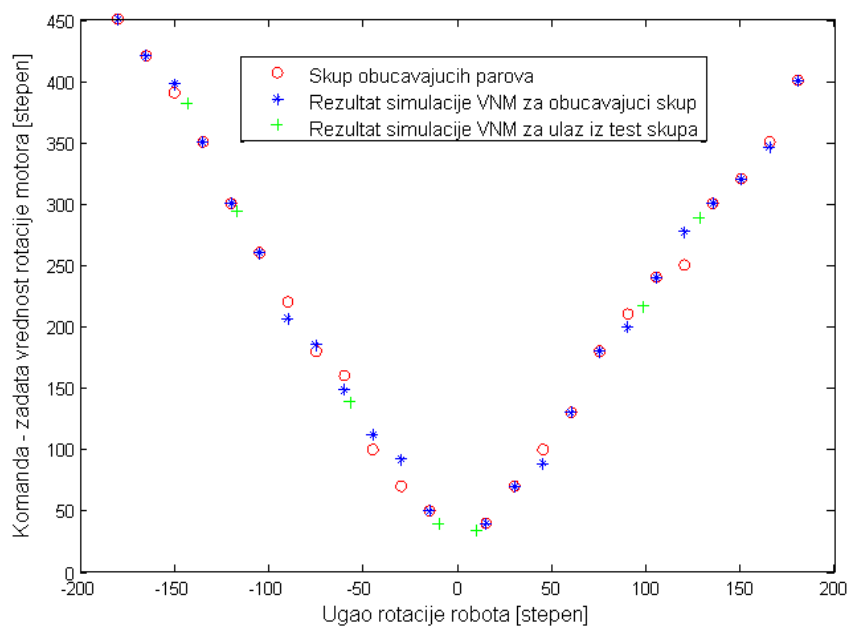
- *show=50* – број итерација након кога се приказује промена,
- *mc=0.9* и *mi=0.001* – константе момента,
- *epochs=1000* – број итерација након кога се процес зауставља, иако поступак обучавања није завршен,
- *goal=10⁻⁵* – захтевана вредност грешке између излаза који је остварила мрежа и захтеваног излаза,
- *max_fail=10* – колико пута се дозвољава да ВНМ падне на тесту приликом валидације.

Оно што је такође важно, усвојен је вектор улазних вредности за тестирање вештачке неуронске мреже одмах након завршеног поступка обучавања. Тај вектор садржи жељене вредности ротације робота у глобалном координатном систему, а које при томе нису садржане у обучавајућем скупу. Скуп углова ротације робота за тестирање сваке од обучаваних ВНМ дат је у табели 4.3.3.

Табела 4.3.3: Скуп углова ротације робота за тестирање обучаваних ВНМ-а							
Редни број тест улаза	1	2	3	4	5	6	7
Тест улази (угао ротације робота)	-143	128	-117	-10	10	98	-57

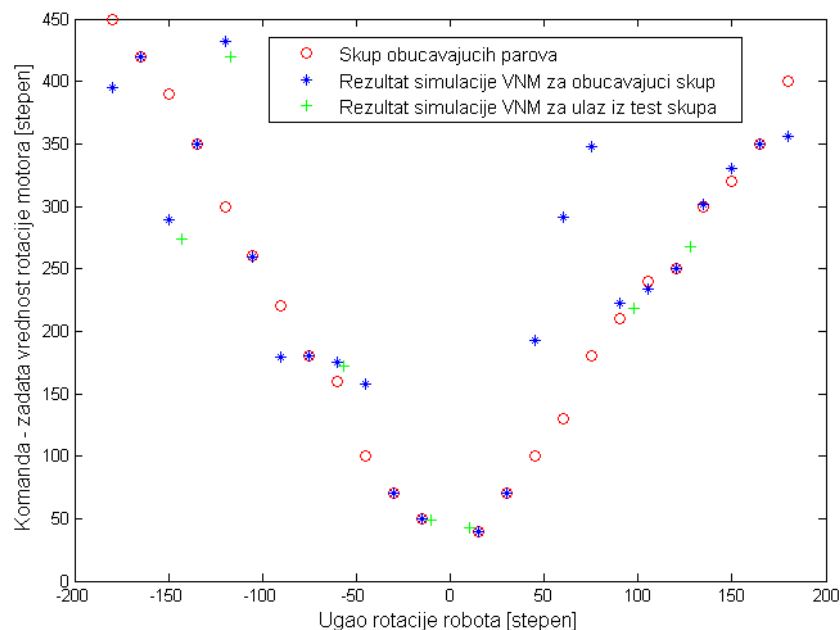
Као резултат процеса обучавања вештачких неуронских мрежа за дефинисан скуп обучавајућих парова, тест улазе и дефинисане параметре обучавања, пратећи установљени план експеримента, поред резултата који карактеришу успешност обучавања и које на крају тог процеса даје *Neural Network Toolbox*, даје се и упоредни графички приказ. Тај графички приказ се формира исцртавањем скупа обучавајућих парова (црвени кругови), резултата симулације ВММ за улаз из обучавајућег скупа (плаве звезде) и резултата симулације ВММ за скуп тест улаза (зелени плусеви).

Како је сужавање избора вештачке неуронске мреже за ротацију робота приликом кретања у технолошком окружењу, која ће се имплементирати у управљачки код мобилног робота, у првој фази спроведено упоређујући претходно дефинисане графичке приказе резултата симулације за сваку од обучавањих ВММ, у наставку је приказан по један пример за ВММ која након обучавања и симулације даје добре резултате (слика 4.3.3), и лоше резултате (слика 4.3.4).



Слика 4.3.3: Пример графичког приказа симулације ВММ која даје добре резултате (ВММ под редним бројем 2)

Са слике 4.3.3 уочава се, за мрежу која даје добре резултате, да је преклапање вредности из обучавајућег скупа (црвени кругови) са вредностима које даје ВММ након симулације (плаве звезде и зелени плусеви) у великој мери остварена. За разлику од добрих резултата, за мрежу која даје лоше резултате (слика 4.3.4) уочава се да је преклапање поменутих вредности знатно мање.



Слика 4.3.4: Пример графичког приказа симулације ВНМ која даје лоше резултате (ВНМ под редним бројем 6)

Приликом упоређивања графичког приказа резултата симулације за сваку од обучаваних неуронских мрежа, визуелно је вршено утврђивање степена поклапања резултата симулације ВНМ за улаз из обучавајућег скупа (плаве звезде) са вредностима из обучавајућег скупа (црвени кругови). Визуелни увид, у оно што мрежа може да оствари након обучавања, употпуњују резултати симулације ВНМ за улаз из тест скупа (зелени плусеви). На овај начин издвојене су четири ВНМ у ужи избор за имплементацију, а то су ВНМ под редним бројевима експеримента: 1, 2, 7 и 10.

Приказ резултата које даје *Neural Network Toolbox* дат је у табели 4.3.4, за све обучаване ВНМ према плану експеримента. Посебно су означене мреже које су издвојене визуелним упоређивањем графичких приказа резултата симулације ВНМ.

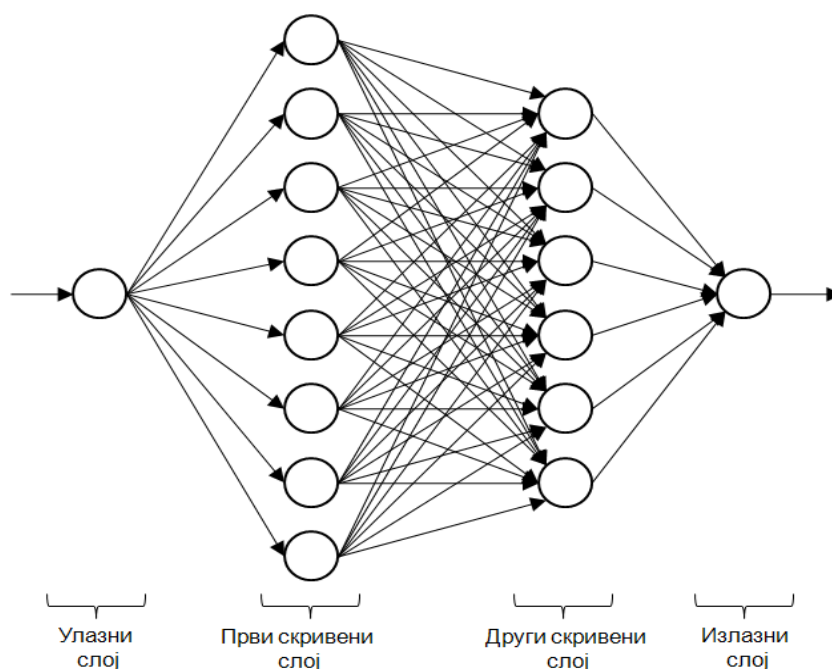
Табела 4.3.4: Резултати обучавања ВНМ-а за ротацију робота током кретања у технолошком окружењу					
Редни број експеримента	Број итерација	Завршен процес минимизације грешке	Остварена минимална грешка	Остварена вредност градијента	Број негативних валидација мреже
1	7	Не	$2.15 \cdot 10^{-1}$	0.00243	2
2	4	Да	$4.77 \cdot 10^{-9}$	0.0315	0
3	7	Не	$1.65 \cdot 10^3$	0.000393	4
4	4	Да	$1.34 \cdot 10^{-9}$	0.0177	0
5	3	Да	$1.78 \cdot 10^{-6}$	0.614	0
6	7	Не	$0.15 \cdot 10^3$	0.00768	5
7	6	Да	$2.35 \cdot 10^{-12}$	0.00046	0
8	6	Не	$0.646 \cdot 10^3$	1.09	4
9	8	Не	$647 \cdot 10^{-2}$	1	2
10	5	Да	$2.96 \cdot 10^{-6}$	0.872	0
11	6	Не	$193 \cdot 10^{-1}$	0.68	2
12	6	Не	$0.514 \cdot 10^3$	0.00336	3

Коначна одлука о избору ВММ за имплементацију у управљачки систем донесена је на основу анализе резултата датих у табели 4.3.4, и то за издвојене четири ВММ на основу визуелне анализе графичких приказа резултат. За прву по реду (1), од четири изабране ВММ, одмах се уочава да процес минимизације грешке није завршен закључно са завршетком поступка обучавања, за разлику од остале три. Такође прва мрежа има две негативне валидације у току поступка обучавања, па се из ових разлога избор даље сужава на преостале три.

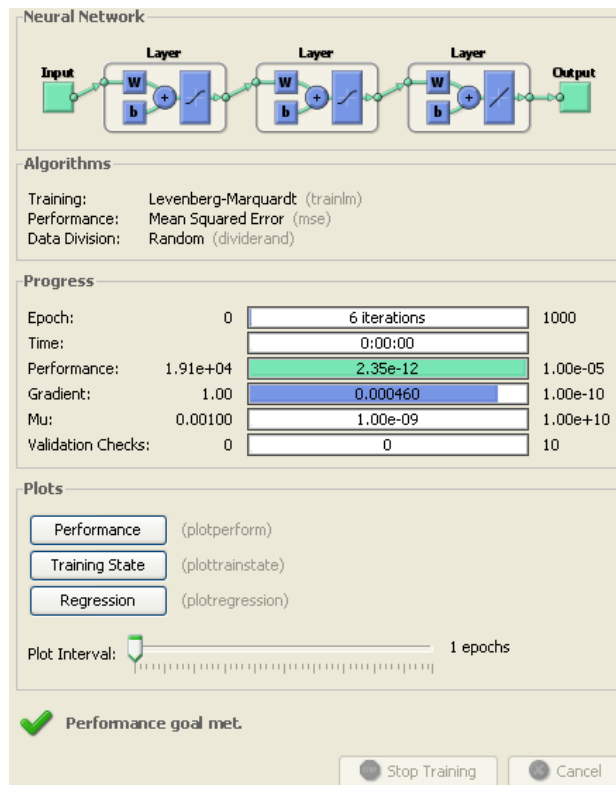
Како преостале три ВММ не карактеришу недостаци који су приписани првој, следи једноставно упоређивање остварене минималне грешке и постигнуте вредности градијента. Из резултата датих у табели 4.3.4 закључује се да најнижу вредност грешке и градијента има ВММ редним бројем 7. Из табеле 4.3.2 се може уочити да је то вештачка неуронска мрежа са два скривена слоја, са 8 у првом и 6 неурона у другом скривеном слоју, као и да је параметар учења $\eta=0.05$.

Како у општем случају број неурона у улазном и излазном слоју неуронске мреже зависи од димензија улазног, односно излазног вектора, тако у овом случају имамо по један неурон у улазном и излазном слоју. Разлог овоме је у томе што је неуронска мрежа развијена тако да, без обзира колики је улазни скуп, врши симулацију за по један улаз и као резултат даје по један излаз.

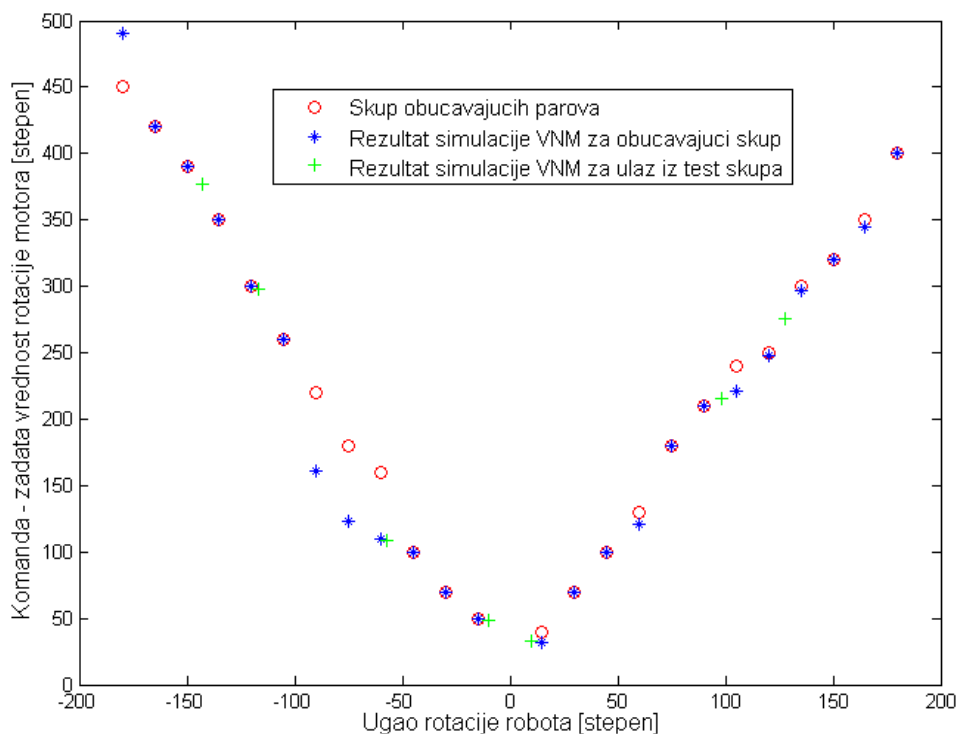
На слици 4.3.5 даје се топологија изабране ВММ, на слици 4.3.6 командни прозор који је у *Neural Network Toolbox*-у активан у току обучавања, тј. са кога по завршетку процеса обучавања читавају резултати тог процеса, док је на слици 4.3.7 дат графички приказ симулације, након процеса обучавања, изабране ВММ.



Слика 4.3.5: Топологија изабране ВММ за одређивање команде за ротацију вратила мотора у односу на потребну ротацију робота у глобалном координатном систему



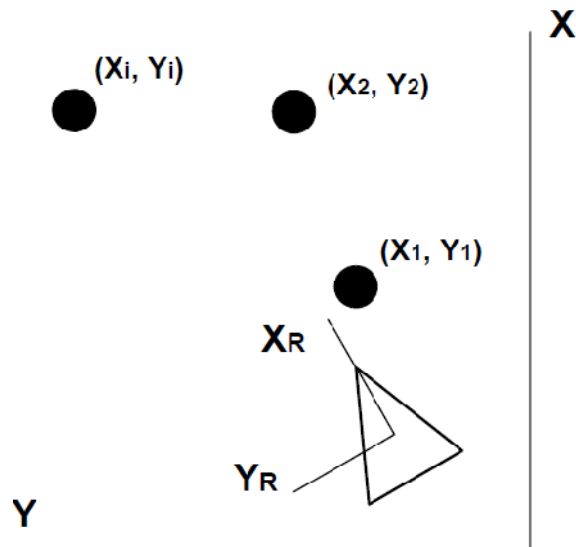
Слика 4.3.6: Приказ резултата обучавања из *Neural Network Toolbox*-а изабране ВНМ за одређивање команде за ротацију вратила мотора у односу на потребну ротацију робота у глобалном координатном систему



Слика 4.3.7: Пример графичког приказа резултата симулације ВНМ која изабрана (ВНМ под редним бројем 7)

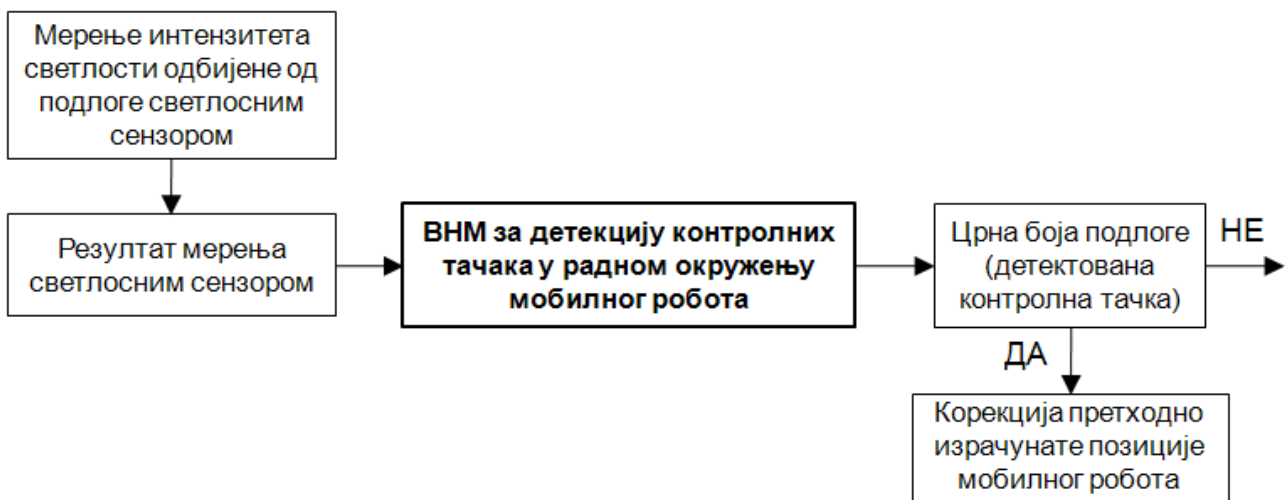
4.3.3 Вештачка неуронска мрежа за детекцију контролних тачака у радном окружењу мобилног робота

Конкретан задатак о коме се овде ради односи се на детекцију контролних тачака у радном окружењу које су црне, за разлику од експерименталног окружења које је беле боје. За сваку од контролних тачака познате су њене координате у глобалном координатном систему (слика 4.3.8).



Слика 4.3.8: Пример произвољне диспозиције контролних тачака у радном окружењу

Заправо, када се на основу мерења које врши сензор, а затим ВМ класификује, установи да се робот налази изнад контролне тачке, потребно је ту информацију проследити управљачком систему. Управљачки систем даље, на основу те информације, врши корекцију положаја мобилног робота применом Калмановог филтера. Ток информација кроз ВМ, за потребе детекције контролних тачака дат је на слици 4.3.9. Проблематика корекције положаја мобилног робота у радном окружењу није од значаја за развој вештачке неуронске мреже, па се у овом делу пројекта даље не разматра.

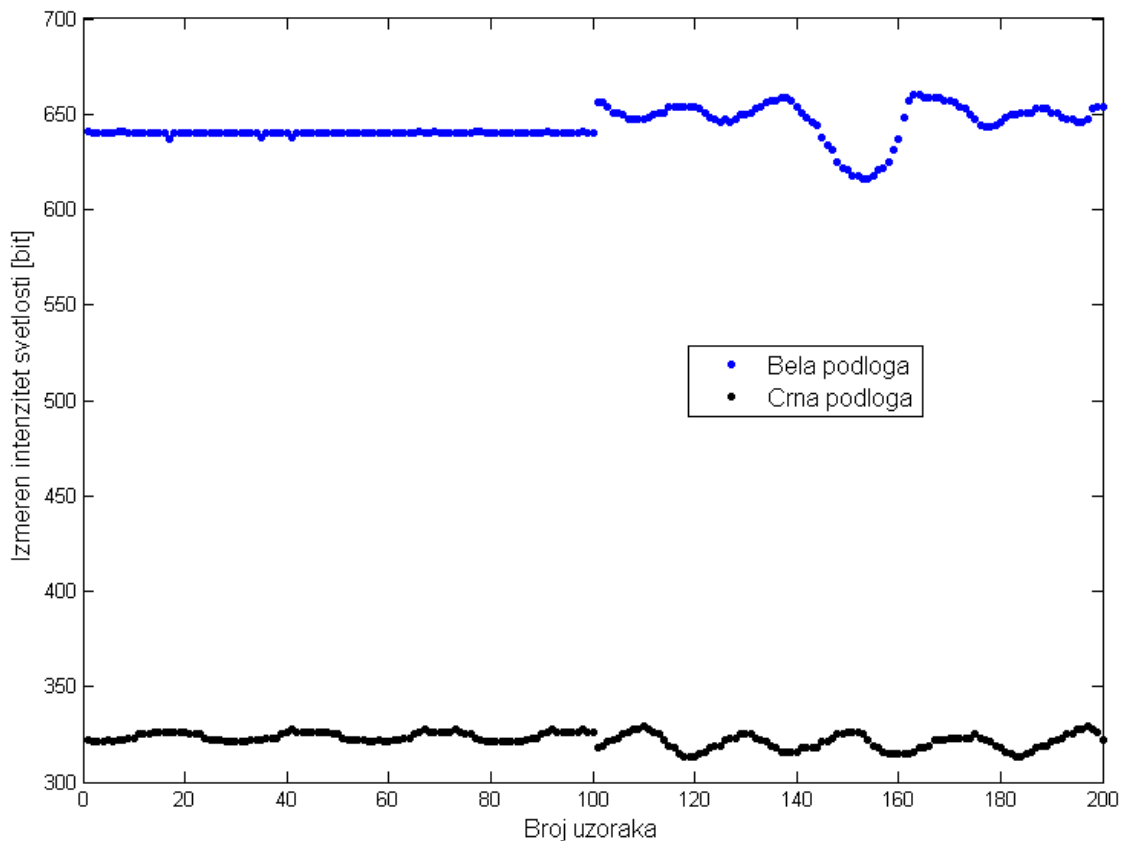


Слика 4.3.9: Ток информација кроз развијени неуронски модел за детекцију контролних тачака у радном окружењу робота

Проблем који ова ВНМ треба да реши, након имплементације у управљачки систем, јавља се при узорковању интензитета светлости одбијене од подлогу, односно при одређивању боје подлоге применом светлосног сензора. Наиме, услед нагле промене амбијенталног осветљења, појаве сенки или затамњења услед уласка робота у угао радног окружења, мерење које расположиви сензор може да оствари не даје јасно разграничење између резултата који карактеришу подлогу беле и црне боје. Генерално, може се закључити да сензор не поседује потребан ниво робусности, тако да се применом ВНМ жели постићи мултиплицирање његових могућности, тј. да се и при граничним (критичним) мерењима одреди поуздана информација за даље акције.

Важно је напоменути да се интензитет светлости одбијене од подлогу, као резултат мерења светлосног сензора, добија у виду 1024 *bit*-ног податка. Овај податак узима вредност 0 уколико нема светлости одбијене од подлогу, односно вредност 1023 уколико је интензитет светлости одбијене од подлогу максималан који сензор може да измери.

Сходно изложеном проблему потребно је формирати такав обучавајући скуп да га карактеришу све сметње које су претходно наведене. Такав обучавајући скуп формиран је експериментално, уз разне промене осветљења у околини сензора, односно места мерења, и то у току кретања мобилног робота, тј. сензора. Укупно је извршено по пет мерења изнад беле подлоге окружења и црних контролних тачака, при чему свако од мерења садржи по 100 узорака. За улазни обучавајући скуп изабрано је по два репрезентативна мерења. Из разлога прегледности приликом увида у резултате мерења, исти се не приказују као скуп бројева, већ графички, исцртавањем на дијаграму, слика 4.3.10.



Слика 4.3.10: Графички приказ изабраног репрезентативног резултата мерења сензора

Са слике 4.3.10 се уочава првих 100 узорака који се односе на мерење са тренутним расположивим амбијенталним осветљењем, тј. без стварања додатних сметњи у окружењу, док се других 100 узорака односи на мерење са стварањем сметњи током мерења (симулирање различитих сенки заклањањем извора амбијенталног осветљења). Као

последица сметњи могу се уочити далеко веће флукуације измереног интензитета светлости него у случају мерења без сметњи. Такође се може установити да су уобичајене вредности мерења за белу подлогу (боју) ~640 *bit*, а за црну ~320 *bit*

Даљи ток формирања улазног обучавајућег скупа своди се на нормализацију изабраних репрезентативних резултата, приказаних на слици 4.3.10. Овим поступком се добијају вредности x које су у границама $-1 \leq X_i \leq 0$ за резултате који се односе на мерење када је подлога црна, и у границама $0 \leq X_i \leq 1$ за резултате који се односе на мерење када је подлога бела. Овим се постиже да максимална вредност резултата мерења узима вредност 1 за белу, тј. -1 за црну боју, док остале вредности узорака мерења узимају вредности пропорционалне максималној у дефинисаним границама. Превођење резултата мерења у претходно дефинисане вредности врши се на основу израза (9), где је x скуп узорака (резултата мерења).

$$x_i = \frac{x - (\max(x) + \min(x)) / 2}{(\max(x) - \min(x)) / 2} \quad (9)$$

Превођењем резултата мерења, у вредности које су претходно дефинисане, постиже се формирање улазног обучавајућег скупа за који ће процес обучавања вештачке неуронске мреже бити неупоредиво бржи и успешнији, а самим тим обучена ВММ биће веома робусна. Постигнута робусност вештачке неуронске мреже је веома значајна у ситуацијама када су узорци који су добијени мерењем близу границе која одговара разграничењу беле и црне боје.

Након формирања улазног обучавајућег скупа неопходно је дефинисати излазни обучавајући скуп, односно извршити комплетирање скупа обучавајућих парова. У овом случају одлучено је да се то изведе на веома једноставан начин – излаз који одговара белој боји је 1, а излаз који одговара црној боји је -1. Уочава се да се овакве вредности добијају и након превођења резултата мерења у скуп x_i па се може посумњати у потребу за вештачком неуронском мрежом. Међутим када се, као што је претходно наведено, добију резултати мерења који су испод уобичајених вредности за белу, тј. изнад уобичајених вредности за црну боју, тада вредности које се добијају након превођења у скуп x_i теже ка нули. Робусна ВММ и у тим случајевима, за потребе доношења одлуке о којој се боји ради, ефикасно класификује резултате мерења у белу, односно црну боју.

Као и при развоју ВММ за ротацију робота током кретања у оквиру модела технолошког окружења, потребно је и овде установити план експеримента по коме ће се кроз процес обучавања утврдити одговарајућа структура мреже и вредност параметра учења. План експеримента дат је у табели 4.3.5.

Табела 4.3.5: План експеримента обучавања ВММ за детекцију контролних тачака		
Број неурона у скривеном слоју	Параметар учења	Редни број експеримента
2	0.05	1
	0.2	2
	0.6	3
4	0.05	4
	0.2	5
	0.6	6
6	0.05	7
	0.2	8
	0.6	9

За сваку од ВНМ дефинисаних планом експеримента задати су и параметри који се односе на сам процес обучавања и овом приликом нису варирано. То су следећи параметри:

- show=50 – број итерација након кога се приказује промена,
- mc=0.9 i mu=0.001 – константе момента,
- epochs=1000 – број итерација након кога се процес зауставља, иако поступак обучавања није завршен,
- goal=10⁻⁵ – захтевана вредност грешке између излаза који је остварила мрежа и захтеваног излаза,
- max_fail=10 – колико пута се дозвољава да ВНМ падне на тесту приликом валидације.

Посебна анализа претходно дефинисаних параметара није извршена ни током развоја ВНМ за детекцију тачака, као ни при развоју ВНМ за ротацију мобилног робота током кретања. Генерално се може закључити да је нпр. број итерација након кога се процес зауставља могао бити веома нижи сходно броју итерација колико су трајали процеси обучавања мрежа. Такође је и број дозвољених негативних валидација могао бити мањи, пошто са дефинисаном вредношћу (10) ниједан процес обучавања ВНМ није из тог разлога заустављен, односно ниједна ВНМ није искључена из разматрања из тог разлога. Практично ове вредности, овако дефинисане, нису биле ограничавајући фактор, тј. нису имале тежину приликом одлучивања која ће се од обучаваних ВНМ имплементирати.

Такође, за сваку од ВНМ из плана експеримента дефинисана је сигмоидна активациона функција (*tansig*) за скривени слој, затим се још дефинише примена Левенберг-Маркеоовог алгоритма учења (*trainlm*),

Имајући у виду да се у овом случају ради о релативно једноставнијем проблему него приликом развоја ВНМ која се односи на ротацију мобилног робота, овде се не дају додатни графички прикази резултата симулације сваке од обучаваних мрежа, већ се у табели 4.3.6 приказују резултати које, као резултат обучавања мреже, даје *MATLAB*-ов *Neural Network Toolbox*.

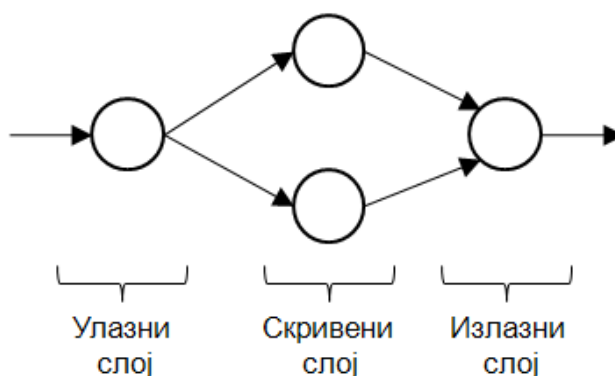
Табела 4.3.6: Резултати обучавања ВНМ-а за детекцију контролних тачака у радном окружењу мобилног робота					
Редни број експеримента	Број итерација	Завршен процес минимизације грешке	Остварена минимална грешка	Остварена вредност градијента	Број негативних валидација мреже
1	3	Да	6.21·10 ⁻⁹	0.0000408	0
2	3	Да	1.17·10 ⁻⁶	0.00212	0
3	20	Да	1.32·10 ⁻⁶	0.00252	0
4	2	Да	1.67·10 ⁻⁶	0.00109	0
5	2	Да	3.3·10 ⁻⁷	0.000138	0
6	2	Да	1.5·10 ⁻⁶	0.0024	0
7	2	Да	1.03·10 ⁻⁶	0.00223	0
8	2	Да	2.05·10 ⁻⁶	0.00309	0
9	2	Да	1.1·10 ⁻⁷	0.00101	0

Анализирајући приказане резултате у табели 4.3.6 најпре се уочава да за све обучаване вештачке неуронске мреже важи да је процес минимизације грешке завршен, да

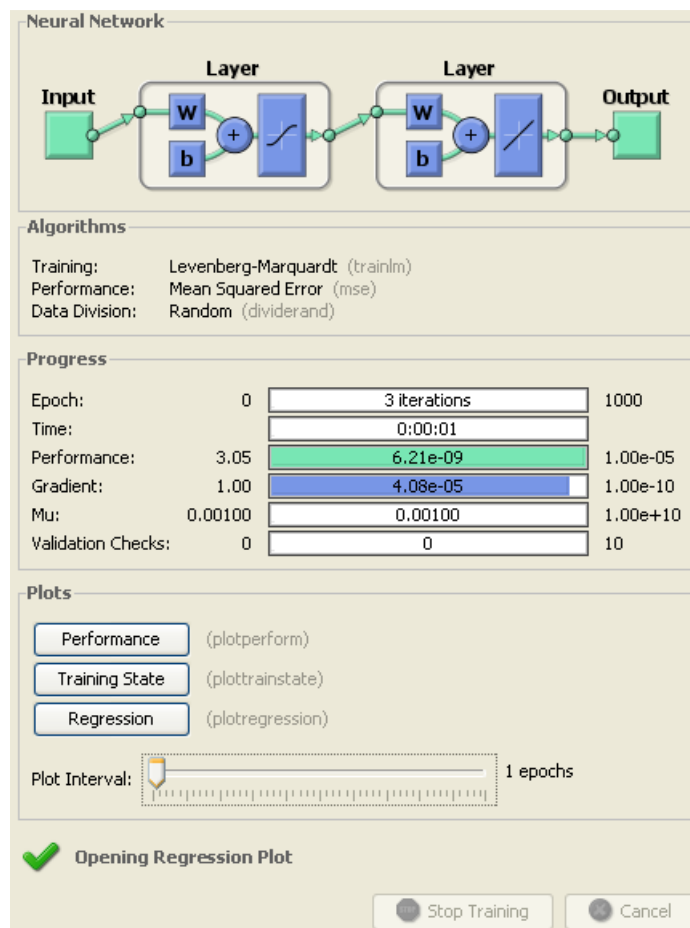
није достигнута минимална вредност градијента и да ниједна мрежа није имала негативну валидацију током процеса обучавања.

Као преостали параметри за избор ВММ за детекцију контролних тачака у радном окружењу мобилног робота, која ће се имплементирати у управљачки систем, су остварене вредности минималне грешке и градијента. Упоредјујући ове вредности долази се до закључака да су најниже за мрежу под редним бројем 1, па се тиме завршава и избор ВММ за детекцију контролних тачака у радном окружењу мобилног робота. Из табеле 4.3.5 се може уочити да је то вештачка неуронска мрежа два неурона у скривеном слоју, као и да је параметар учења $\eta=0.05$. Ова ВММ је развијана као и претходна, односно симулацију врши за по један улаз и даје по један излаз, тако да је и у овом случају по један неурон у улазном и излазном слоју.

У наставку се даје топологија изабране ВММ (слика 4.3.11) и командни прозор који је у *Neural Network Toolbox*-у активан у току обучавања, тј. са кога по завршетку процеса обучавања читавају резултати симулације (слика 4.3.12)



Слика 4.3.11: Топологија изабране ВММ за детекцију контролних тачака у радном окружењу мобилног робота



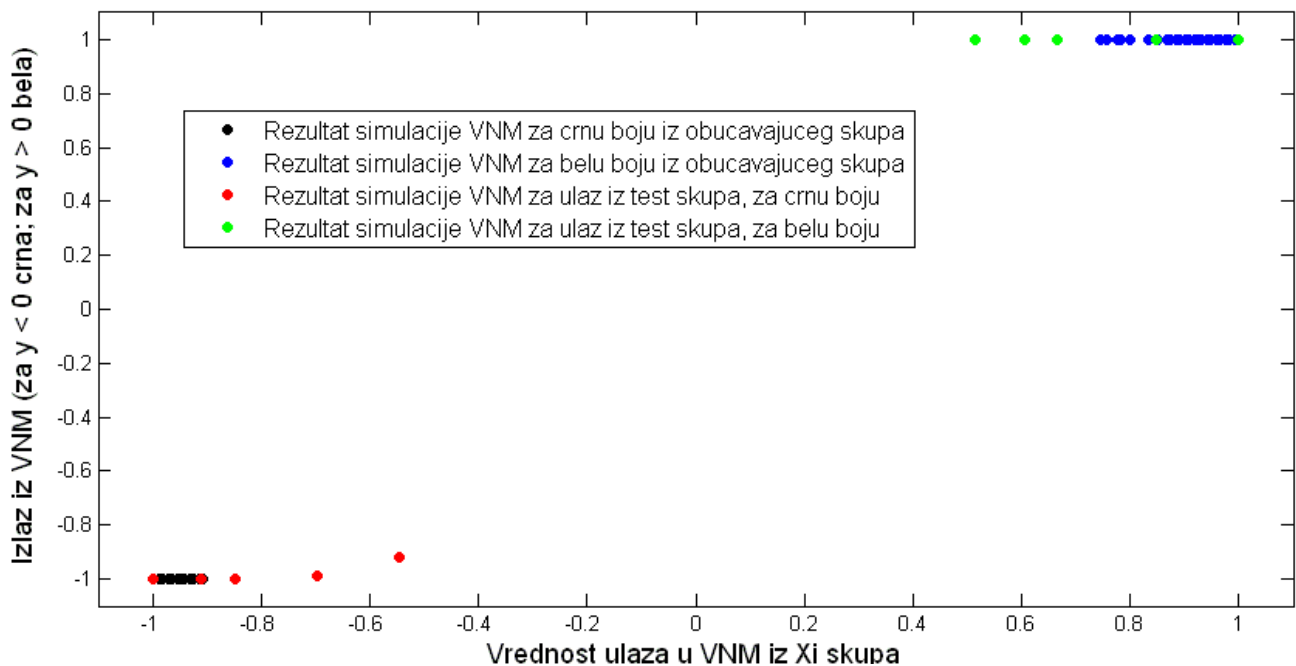
Слика 4.3.12: Приказ резултата обучавања из *Neural Network Toolbox*-а изабране ВММ за детекцију контролних тачака у радном окружењу мобилног робота

Као још једна провера изабране вештачке неуронске мреже формиран је улазни тест скуп са вредностима које нису садржане у обучавајућем скупу. Поред тога, вредности у тест скупу су такве да теже ка граничним вредностима мерења, односно вредности дефинисане као резултат мерења за црну боју су изнад уобичајених вредности ($\sim 320 \text{ bit}$), а за белу боју испод уобичајених вредности ($\sim 640 \text{ bit}$). Улазни тест скуп дат је у табели 4.3.7.

Табела 4.3.7: Скуп улазних вредности за проверу изабране ВММ за детекцију контролних тачака у радном окружењу мобилног робота

Редни број тест улаза	1	2	3	4	5
Улазни тест скуп за белу подлогу	500	511	498	506	495
Улазни тест скуп за црну подлогу	450	445	448	455	460

Подразумева се да се улазни скуп дат табелом 4.3.7, пре симулације ВММ, преводи у скуп x_i поступком који је претходно детаљно објашњен, применом израза (18). Резултат ове симулације дат је на слици 4.3.13 упоредо са резултатима симулације ВММ за вредности из обучавајућег улазног скупа.



Слика 4.3.13: Резултат симулације изабране ВНМ за улазе из обучавајућег и тест скупа

Након спроведене симулације ВНМ и резултата приказних на слици 4.3.13, може се закључити да изабрана ВНМ за детекцију контролних тачака у радном окружењу мобилног робота поседује високу робусност, што се огледа у томе да и вредности из скупа x_i које теже нули, односно граничним вредностима, јасно класификује у црну (~ -1) и белу боју (~ 1).

4.4 Примена алгоритма Калмановог филтера у циљу одређивања положаја мобилног робота у окружењу

У циљу одређивања положаја мобилног робота у задатом технолошком окружењу, неопходно је најпре дефинисати теоријске основе, а потом дати једначине Калмановог филтера, који се користити за решавање проблема локализације.

4.4.1 Основне теоријске поставке Калмановог филтера

За увођење Калмановог филтера заслужан је Рудолф Е. Калман, по коме је филтер и добио име. Данас се широко користи у области теорије оптималног управљања различитих система, у области обраде сигнала и у навигацији. Пример таквих система су и мобилни роботи, а циљ овог пројектног задатка је анализа могућности њихове имплементације (примене) у различитим технолошким системима, ради успостављања унутрашњег транспорта материјала.

Наиме, примена Калмановог филтера за естимацију положаја робота у технолошком окружењу темељи се на примени модела кретања на основу пређеног пута. Из овог модела, као што је већ објашњено, добијају се вредности унутрашњих координата (број обртаја вратила мотора) за задате вредности спољашњих координата, односно, за задат положај и оријентацију мобилног робота. Једначине које описују зависност спољашњих од унутрашњих координата добијају се решавањем инверзног кинематичког проблема.

Естимација се може дефинисати као процена вредности неке величине до које се долази посредно путем мерења. По правилу, мерења су нетачна, несигурна и непрецизна, али се на основу њих, захваљујући естимацији, може предвидети стање неког система. Одређивање параметара математичког модела једна је од широког спектра примене екстимације. Сврха екстимације у овом пројекту је одређивање положаја робота, односно позиције и оријентације.

У општем случају, сигнали добијени са сензора састоје се из две величине и могу се према [7] представити на следећи начин:

$$y(t) = x_1(t) + x_2(t) \quad (10)$$

Где су:

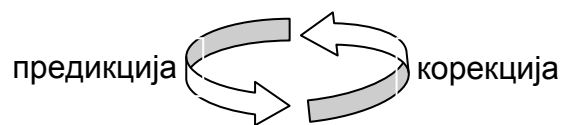
- $y(t)$ - сензорска информација, која представља збир правог сигнала и шума;
- $x_1(t)$ - вредност сигнала за пређени пут који се читава са сензора;
- $x_2(t)$ - шум у сензорској информацији;

Процес одређивања стварне вредности сигнала $x_1(t)$ и вредности шума $x_2(t)$ у сензорској информацији $y(t)$ назива се *филтрација*.

Како сензори генеришу континуалан и временски променљив сигнал (аналогни сигнал), одређивање дискретних вредности жељеног сигнала у одређеном временском интервалу, на пример од тренутка $t=0$ закључно са временским тренутком t , може послужити за предвиђање стања система, односно неке величине тог система у временском тренутку t_1 . Овај вид примене Калмановог филтера назива се *предикција*.

Поред филтрације и предикције, позната је још једна категорија за екстимацију параметара стања неког система. Назива се *интерполација* и користи се за прикупљање валидних информација о неком систему.

На тај начин, омогућена је имплементација Калмановог филтера као естиматора на бази предикције и корекције, који је симболично приказан сликом 4.4.1 и детаљно ће бити разматран приликом појашњавања пројектног решења.



Слика 4.4.1: Симболички приказ естиматора за предикцију и корекцију

4.4.2 Основне једначине Калмановог филтера

Из теорије аутоматског управљања позната је подела система на статичке и динамичке у односу на величине потребне за математичко описивање излаза, односно, одзива ових система [8]. У складу са претходно наведеним, статички системи се дефинишу као системи чији је излаз једнозначно одређен улазом, док је излаз динамичких система једнозначно одређен и стањем система и улазом. Тако се доводи до појма стања система, као и неопходности постојања једначина за описивање стања система. Познавање стања система у тренутку t , као и познавање промене улаза од тренутка t па надаље, једино једнозначно одређују даљу промену излаза и стање самог система.

Стање система се према моделу општег просторног кретања дефинише преко положаја и оријентације робота и то као вектор $x_t = (x, y, z, \theta, \psi, \varphi)$, док се за раванско кретање вектор стања система дефинише са три компоненте $x_t = (x, y, \theta)$.

Где су:

- x, y, z - компоненте вектора стања које одређују позицију мобилног робота;
- θ, ψ, φ - компоненте вектора стања које одређују оријентацију мобилног робота;

Једначина стања система се, сходно основној формулацији Калмановог филтера, може представити у следећем облику:

$$x_{k+1} = A_k x_k + B_k u_k + w_k \quad (11)$$

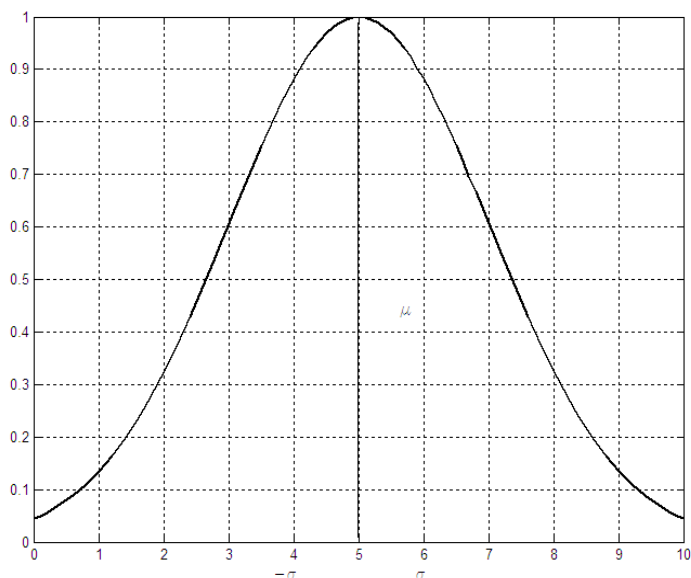
$$y_k = C_k x_k + v_k \quad (12)$$

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 се дефинише како систем прелази из стања x_k у стање x_{k+1} ;
- w_k - шум система; неодређеност у систему; случајни вектор тј. случајна променљива помоћу које моделујемо понашање система; грешка; поремећај који се не може контролисати јер је систем стохастички- није детерминистички;
- y_k - једначина излаза тј. једначина мерења;
- C_k - матрица која успоставља везу између тренутног стања и тренутног мерења;
- v_k - шум мерења;

Једначина (2) представља линеарну стохастичку диференцијалну једначину стања, док једначина (3) представља једначину мерења.

Како је систем стохастички, што значи функционише под дејством случајних догађаја и да се о његовом будућем стању може говорити само у домену одређене вероватноће (не може се аналитички описати), то се величине w_k и v_k моделирају тако да подлежу закону Гаусове расподеле. Гаусова „нормална” расподела за скаларну случајну променљиву приказана је на слици 4.4.2, при чему су μ и σ^2 очекивана вредност и варијанса, респективно.



Слика 4.4.2: Графички приказ Гаусове нормалне расподеле

Математички приказано, Гаусова расподела за векторску случајну променљиву x , се може представити на следећи начин:

$$p(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\} \quad (13)$$

Где су са μ и Σ означене матрице очекиване вредности и матрице коваријансе, респективно, а d представља број димензија вектора. Пошто је експоненцијална функција увек позитивна и растућа, то именилац једначине (22) обезбеђује да вероватноћа, према аксиому теорије вероватноће, има вредност $0 \leq p(x) \leq 1$.

Након овако дефинисане Гаусове расподеле, филтер се заснива на две претпоставке. Оне се односе на величине w_k и v_k и могу се представити на следећи начин:

$$p(w_k) \sim N(0, Q) \quad (14)$$

$$p(v_k) \sim N(0, R) \quad (15)$$

Једначина (14) представља претпоставку да шум (грешка) система подлеже Гаусовој нормалној расподели, при чему је очекивана вредност нула, а коваријанса Q . Аналогно, за шум мерења се према нормалној расподели претпоставља очекивана вредност нула и коваријанса R , као што је приказано једначином (15).

Даље, модел промене стања система приказан је једначином (16).

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \quad (16)$$

Треба напоменути и да се често вектор стања система у тренуту t не може директно мерити, па се из тог разлога сензорима региструје вредност Z_t која се представља линеарном комбинацијом променљивих стања и шума мерења, што је и представљено једначином (17).

$$z_t = C_t x_t + \delta_t \quad (17)$$

где су основне величине које фигуришу у једначинама (25) и (26):

- x_t - вектор стања система у тренутку t ;
- A_t - матрица која успоставља везу између стања претходног и тренутног корака без утицаја управљања и шума;
- x_{t-1} - вектор стања система у тренутку $t-1$;
- B_t - матрица која повезује управљање и стање система у тренуцима t и $t-1$;
- u_t - вектор управљања помоћу које се дефинише како систем прелази из стања x_k у стање x_{k+1} ;
- ε_t - шум система; неодређеност у једначини система; случајни вектор тј. случајна променљива помоћу које моделујемо понашање система; грешка коју моделирамо; поремећај који се не може контролисати јер је систем стохастички- није детерминистички;
- Z_t - вектор мерења у тренутку t ;
- C_t - матрица која успоставља везу између тренутног стања и тренутног мерења;
- δ_k - случајна променљива која описује шум мерења; неодређеност у мерењу;

Из једначина (16) и (17) се види да постоје два извора информација. На основу модела можемо да претпоставимо каква ће бити промена вредности вектора стања између тренутака $t-1$ и t . Као што је објашњено, ова процена се назива предикција. На основу мерења врши се корекција и на тај начин се затвара циклус Калмановог филтера приказан на слици 4.4.1.

4.4.3 Алгоритам Калмановог и линеаризованог Калмановог филтера

Оптимална оцена, односно филтрирање, представља процес естимације (предвиђања) стања или параметара система, на основу вредности добијених мерењем. На овом принципима се заснива и темељи Бајесов филтер, који је општији од Калмановог. Када је систем линеаран, шум адитиван и подлеже Гаусовој расподели из Бајесовог филтера можемо да изведемо Калманов филтер. Зато се може рећи да је Калманов филтер ефикасно рекурзивно решење, односно, рекурзивни алгоритам за проблем дискретног линеарног филтрирања. За разлику од нерекурзивне естимације (предвиђања) 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 се временом повећава.

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

Табела 4.4.1: Алгоритам Калмановог филтера			
1.	улазне вредности	АЛГОРИТАМ Калманов_филтер($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)	предикција
2.		$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$	
3.		$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$	
4.		$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$	корекција
5.		$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$	
6.		$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$	
7.	излазне вредности	μ_t, Σ_t	

Величине које се користе за алгоритам Калмановог филтера су:

- μ_{t-1} - средња вредност у тренутку t-1;
- Σ_{t-1} - матрица коваријансе у тренутку t-1, тј. колико смо сигурни у μ_t ;
- u_t - управљање у тренутку t;
- z_t - мерење у тренутку t – читавања 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 тренуцима t и t-1;
- $\bar{\mu}_t$ - предикција стања система;
- $C_t \bar{\mu}_t$ - предикција мерења;
- Σ_t - коваријанса;
- Q_t - матрица коваријанси шума мерења; несигурност у мерењу;

Користећи термине аутоматског управљања, Калманов филтер представља линеарни регулатор, који поседује таква појачања која минимизирају варијансу грешке.

Да би се Калманов филтер применио на нелинеарним моделима, неопходно је модел линеаризовати 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 (18). Такође, и опсервациони модел је нелинеарна функција стања система што је приказано једначином (19) и дато у [4]. Функција $g(u_t, x_{t-1})$ показује како се мењају x, y, θ у функцији управљања и средње вредности.

$$x_t = g(u_t, x_{t-1}) \quad (18)$$

$$z_t = h(x_t) \quad (19)$$

Да би се дата функција представила у околини тачке по избору, користи се развијање функције у Тејлоров ред. Задржавајући се само на прва два члана Тејлоровог реда, изводи функције $g(u_t, x_{t-1})$ за изабрану тачку μ_{t-1} и функције $h(x_t)$ у изабраној тачки $\bar{\mu}_t$ имају следећи запис:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1}) \quad (20)$$

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1}) \quad (21)$$

$$h(x_t) \approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t} (x_t - \bar{\mu}_t) \quad (22)$$

$$h(x_t) \approx h(\bar{\mu}_t) + H_t (x_t - \bar{\mu}_t) \quad (23)$$

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, односно, баждарење сензора. Оба алгоритма важе уколико модел подлеже Гаусовој расподели. Ако то није случај, резултати добијени применом филтера нису довољно тачни.

На основу свега дефинисаног, алгоритам линеаризованог Калмановог филтера је приказан у табели 4.4.2.

Табела 4.4.2: Алгоритам линеаризованог Калмановог филтера			
1.	улазне вредности	АЛГОРИТАМ Линеаризовани_Калманов_филтер($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)	предикција
2.		$\bar{\mu}_t = g(u_t, \mu_{t-1})$	
3.		$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$	
4.		$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$	корекција
5.		$\mu_t = \bar{\mu}_t + K_t (z_t - h_t(\bar{\mu}_t))$	
6.		$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$	
7.	излазне вредности	μ_t, Σ_t	

Закључци:

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 коваријанса повећава, расте несигурност у систем, па расте и вредност Калмановог појачања; Такође, уколико опада вредност коваријансе грешке мерења (мерење прецизно), вредност Калмановог појачања се повећава;
- Алналогно, уколико се коваријанса 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 $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$, која се односи на корекцију (ажурирање стања), може се закључити да део једначине представљен са $z_t - C_t \bar{\mu}_t$ односи на координату из мапе окружења. Ова разлика представља иновацију.

Вредност z_t се добија мерењем, док је $C_t \bar{\mu}$ вредност која се добија предвиђањем. Уколико је вредност грешке сензора велика, вредност Калмановог појачања ће бити веома мала, па ће и та грешка, помножена са малом вредности Калмановог појачања такође имати малу вредност и неће утицати на вредност жељеног сигнала са енкодера.

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

Локализација применом линеаризованог Калмановог филтера - корак предикције:

Табела 4.4.3: Алгоритам линеаризованог Калмановог филтера

Рб.	Математичка формулација	Опис
1.	ЛКФ_локализација ($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m$)	
2.	$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,z}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,z}} \\ \frac{\partial z'}{\partial \mu_{t-1,x}} & \frac{\partial z'}{\partial \mu_{t-1,y}} & \frac{\partial z'}{\partial \mu_{t-1,z}} \end{pmatrix}$	Јакобијан функције g у односу на положај робота
3.	$V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix} \frac{\partial x'}{\partial u_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial \omega_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial z'}{\partial \omega_t} & \frac{\partial z'}{\partial \omega_t} \end{pmatrix}$	Јакобијан функције g у односу на управљање
4.	$M_t = \begin{pmatrix} (\alpha_1 u_t) + (\alpha_2 \omega_t)^2 & 0 \\ 0 & (\alpha_3 u_t) + (\alpha_4 \omega_t)^2 \end{pmatrix}$	Шум, грешка управљања
5.	$\bar{\mu}_t = g(u_t, \mu_{t-1})$	Предвиђена очекивана вредност
6.	$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T$	Предвиђена матрица коваријанси

Јакобијан функције g у односу на положај робота дефинисан је једначином (24).

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \mu_{t-1,\theta}} \end{pmatrix} \quad (24)$$

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

$$\frac{\partial x'}{\partial x} = 1; \quad \frac{\partial x'}{\partial y} = 0; \quad \frac{\partial x'}{\partial \theta} = \frac{\Delta s_d + \Delta s_l}{2} \cdot \left(-\sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right)\right) \cdot 1 \quad (25)$$

$$\frac{\partial y'}{\partial x} = 0; \quad \frac{\partial y'}{\partial y} = 1; \quad \frac{\partial y'}{\partial \theta} = \frac{\Delta s_d + \Delta s_l}{2} \cdot \cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \cdot 1 \quad (26)$$

$$\frac{\partial \theta'}{\partial x} = 0; \quad \frac{\partial \theta'}{\partial y} = 0; \quad \frac{\partial \theta'}{\partial \theta} = 1 \quad (27)$$

Јакобијан функције g у односу на управљање робота се дефинише преко следећих парцијалних извода, што је представљено једначинама (28) – (32):

$$\frac{\partial x'}{\partial \Delta s_d} = \frac{1}{2} \left(\cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) + \frac{\Delta s_d + \Delta s_l}{2} \cdot \left(-\sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right)\right) \cdot \left(\frac{1}{2b}\right) \right) \quad (28)$$

$$\frac{\partial x'}{\partial \Delta s_l} = \frac{1}{2} \left(\cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) + \frac{\Delta s_d + \Delta s_l}{2} \cdot \left(-\sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right)\right) \cdot \left(-\frac{1}{2b}\right) \right) \quad (29)$$

$$\frac{\partial y'}{\partial \Delta s_d} = \frac{1}{2} \cdot \sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) + \frac{\Delta s_d + \Delta s_l}{2} \cdot \cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \cdot \frac{1}{2b} \quad (30)$$

$$\frac{\partial y'}{\partial \Delta s_l} = \frac{1}{2} \cdot \sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) + \frac{\Delta s_d + \Delta s_l}{2} \cdot \cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \cdot \left(-\frac{1}{2b}\right) \quad (31)$$

$$\frac{\partial \theta'}{\partial \Delta s_d} = \frac{1}{2b}; \quad \frac{\partial \theta'}{\partial \Delta s_l} = -\frac{1}{2b}; \quad (32)$$

Локализација применом линеаризованог Калмановог филтера - корак корекције:

Табела 4.4.4: Алгоритам линеаризованог Калмановог филтера		
Рб.	Математичка формулација	Опис
1.	ЛКФ_локализација ($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m$);	
2.	$\hat{z}_t = \begin{pmatrix} \sqrt{(m_x - \bar{\mu}_{t,x})^2 + (m_y - \bar{\mu}_{t,y})^2} \\ a \tan 2(m_y - \bar{\mu}_{t,y}, m_x - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{pmatrix}$	Предвиђена очекивана вредност мерења
3.	$H_t = \frac{\partial h(\bar{\mu}_t, m)}{\partial x_t} = \begin{pmatrix} \frac{\partial r_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,z}} \\ \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,z}} \end{pmatrix}$	Јакобијан функције h у односу на положај робота
4.	$Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{pmatrix}$	Шум мерења
5.	$S_t = H_t \bar{\Sigma}_t H_t^T + Q_t$	Предвиђена матрица варијанси мерења
6.	$K_t = \bar{\Sigma}_t H_t^T S_t^{-1}$	Каламаново појачање
7.	$\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$	Нова очекивана вредност
8.	$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$	Нова матрица коваријанси

Једначинама (33) и (34) представљају се координате положаја светлосног сензора у глобалном координатном систему.

$$x_s = x_r + p \cdot \cos(\theta_r) \quad (33)$$

$$y_s = y_r + p \cdot \sin(\theta_r) \quad (34)$$

при чему p представља растојање између сензора и тежишта робота.

Јакобијан функције h у односу на положај робота добија се парцијалним изводима једначина (33) и (34) по вектору стања.

$$\frac{\partial x_s}{\partial x_r} = \frac{\partial}{\partial x_r} (x_r + p \cdot \cos(\theta_r)) = 1 \quad (35)$$

$$\frac{\partial x_s}{\partial y_r} = \frac{\partial}{\partial y_r} (x_r + p \cdot \cos(\theta_r)) = 0 \quad (36)$$

$$\frac{\partial x_s}{\partial \theta_r} = \frac{\partial}{\partial \theta_r} (x_r + p \cdot \cos(\theta_r)) = -p \cdot \sin(\theta_r) \quad (37)$$

$$\frac{\partial y_s}{\partial x_r} = \frac{\partial}{\partial x_r} (y_r + p \cdot \sin(\theta_r)) = 0 \quad (38)$$

$$\frac{\partial y_s}{\partial y_r} = \frac{\partial}{\partial y_r} (y_r + p \cdot \sin(\theta_r)) = 1 \quad (49)$$

$$\frac{\partial y_s}{\partial \theta_r} = \frac{\partial}{\partial \theta_r} (y_r + p \cdot \sin(\theta_r)) = p \cdot \cos(\theta_r) \quad (40)$$

Коначно, матрица Јакобијана функције h има следећи запис:

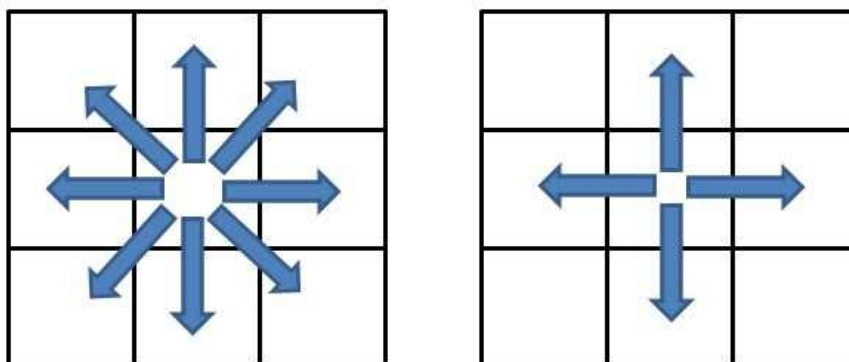
$$H_t = \begin{pmatrix} 1 & 0 & -p \cdot \sin(\theta_r) \\ 0 & 1 & p \cdot \cos(\theta_r) \end{pmatrix} \quad (41)$$

4.5 A* алгоритам

4.5.1 Теоријске поставке и уопштена објашњења A* алгоритма за претраживање

A* алгоритам је један од алгоритама претраживања који проналази најкраћу путању између почетне и циљне тачке у неком окружењу. Почетна и циљна тачка, као и окружење у коме се налазе је предходно дефинисано и задато. Овај алгоритам је веома флексибилан и као такав има широку употребу. Поред тога, као и други граф алгоритми, има ту особину да може да претражи велики простор задатог окружења за кратко време. Овај алгоритам има најопштији облик, односно комбинује предности *Dijkstra* и *BestFirstSearch(BFS)* алгоритма. Ова два поменута алгоритма нису потребна у наставку и нису предмет разматрања, тако да ће се њихов опис занемарити. Користећи хеуристички прихватљиве процене алгоритам поступно пролази кроз задато окружење бирајући при томе најоптималнију путању. Хеуристички прихватљиве процене су процене засноване на искуству на основу којих вршимо брзи одабир оптималног (или приближно оптималног) следећег корака.

Задато окружење је потребно поједноставити односно дискретизовати, представити пикселима дводимензионално (преко x и y координате). Величина и облик пиксела су опциони и бирају се према потреби и жељеној тачности, па тако имамо да што је површина пиксела мања путања је прецизнија, али је време обраде тог подручја (окружења) спорије и обрнуто, за случај већег пиксела прецизност је мања а време обраде је брже. Најчешће коришћен облик пиксела је квадрат, а величина у зависности од диманзија целог окружења може бити изражена у метрима, центриметрима, милиметрима и сл. Кретање је могуће по четвороспојивом окружењу, где су дозвољена померања по вертикали и хоризонтални, и по осмоспојивом окружењу, где је поред наведених дозвољено и дијагонално кретање, а посматра се у односу на центар пиксела (тзв. чвор). На слици 4.5.1 дат је приказ могућег кретања по окружењу.



Слика 4.5.1: Приказ осмоспојивог и четвороспојивог окружења пиксела

Сваки чвор у мрежи садржи оцену (процену) најкраћег пута мерено од задатог циља - $h(n)$, као што је приказано у [4], где је (n) означава позицију чвора у мрежи (двостручнодимензионалној матрици). Један од начина на који се може израчунати оцена пута мерено од задатог циља је коришћење Менхетн норме (Менхетн растојања). Менхетн растојање представља растојање између две тачке, мерено дуж оса под правим углом [10]. Примењујући предходну дефиницију, овом нормом се рачуна укупан број пиксела, крећући се искључиво вертикално и хоризонтално, потребан да се стигне од тренутне позиције до задатог циља, занемарујући могуће постојање препрека. Под препреком се подразумева пиксел који сигнализира постојање физичке препреке у описаном окружењу.

Математички записано, израчунавање Менхетн нормом реализује се на следећи начин:

$$h(n) = |n_{x \text{ тренутно}} - n_{x \text{ циља}}| + |n_{y \text{ тренутно}} - n_{y \text{ циља}}| \quad (42)$$

На слици 4.4.2 дат је пример реализације Менхетн норме, односно, примене једначине (42) у окружењу димензија 6x6. Пиксел је величине 1x1 (јединица мере занемарена јер у примеру није неопходна), облика квадрата. Пиксел обојен зеленом бојом представља стартну, црвеном обојен представља циљну позицију, док сивом обојен представља место на којем се налази препрека у окружењу.

$h(1,1) = 8$	$h(1,2) = 7$	$h(1,3) = 6$	$h(1,4) = 5$	$h(1,5) = 4$	$h(1,6) = 5$
$h(2,1) = 7$	$h(2,2) = 6$	$h(2,3) = 5$	$h(2,4) = 4$	$h(2,5) = 3$	$h(2,6) = 4$
$h(3,1) = 6$	$h(3,2) = 5$	$h(3,3) = 4$	$h(3,4) = 3$	$h(3,5) = 2$	$h(3,6) = 3$
$h(4,1) = 5$	$h(4,2) = 4$	$h(4,3) = 3$	$h(4,4) = 2$	$h(4,5) = 1$	$h(4,6) = 2$
$h(5,1) = 4$	$h(5,2) = 3$	$h(5,3) = 2$	$h(5,4) = 1$	$h(5,5) = 0$	$h(5,6) = 1$
$h(6,1) = 5$	$h(6,2) = 4$	$h(6,3) = 3$	$h(6,4) = 2$	$h(6,5) = 1$	$h(6,6) = 2$

Слика 4.5.2: Приказ оцене најкраћег пута за чворове у окружењу применом Менхетн норме

Ограничење у кретању, односно искључивање могућности кретања дијагонално је један од највећих недостатака ове норме. Поменуто ограничење се елиминише коришћењем Еуклидске норме (Еуклидско растојање). Еуклидско растојање представља праволинијско растојање између две тачке [10]. За разлику од предходне норме, овом нормом је поред вертикалног и хоризонталног кретања омогућено и дијагонално кретање (померање), што за последицу има тачнију оцену поља, у односу на Менхетн норму, која ће проузроковати касније и скраћење пута. О самом кретању и прорачуну пута биће више речи у наставку. Треба напоменути да и у овом случају, као и у предходном, посматрамо кретање у односу на чворове пиксела. Математички запис ове норме приказан је једначином (43):

$$h(n) = \sqrt{(n_x \text{ тренутно} - n_x \text{ циља})^2 + (n_y \text{ тренутно} - n_y \text{ циља})^2} \quad (43)$$

У наставку на слици 4.5.3 је приказана израчунавања Еуклидске норме тј. примена једначине (43) на истом примеру на ком је приказана и Менхетн норма. Напомиње се још једном да је усвојена величина пиксела 1x1. Идеја је да се на основу добијених оцена чворова окружења у овом случају направи поређење са предходном Менхетн нормом, односно оценама добијеним у том случају.

$h(1,1) = 5.6569$	$h(1,2) = 5$	$h(1,3) = 4.4721$	$h(1,4) = 4.1231$	$h(1,5) = 4$	$h(1,6) = 4.1231$
$h(2,1) = 5$	$h(2,2) = 4.2426$	$h(2,3) = 3.6056$	$h(2,4) = 3.1623$	$h(2,5) = 3$	$h(2,6) = 3.1623$
$h(3,1) = 4.4721$	$h(3,2) = 3.6056$	$h(3,3) = 2.8284$	$h(3,4) = 2.2361$	$h(3,5) = 2$	$h(3,6) = 2.2361$
$h(4,1) = 4.1231$	$h(4,2) = 3.1623$	$h(4,3) = 2.2361$	$h(4,4) = 1.4142$	$h(4,5) = 1$	$h(4,6) = 1.4142$
$h(5,1) = 4$	$h(5,2) = 3$	$h(5,3) = 2$	$h(5,4) = 1$	$h(5,5) = 0$	$h(5,6) = 1$
$h(6,1) = 4.1231$	$h(6,2) = 3.1623$	$h(6,3) = 2.2361$	$h(6,4) = 1.4142$	$h(6,5) = 1$	$h(6,6) = 1.4142$

Слика 4.5.3: Приказ оцене најкраћег пута за чворове у окружењу применом Еуклидске норме

Наведене норме су најчешће коришћене приликом реализације A^* алгоритма, и њихова употреба заправо представља хеуристичку процену, на основу које се између осталог бира оптимална путања (у наставку ће детаљно бити објашњен одабир путање).

Следећа карактеристика коју поседује сваки чвор у мрежи је цена помераја $g(n)$ [4]. Цена помераја уствари представља пређени пут од стартног до n -тог чвора у мрежи. Рачуна се за околину чвора у коме се тренутно налазимо (локални прорачун који се понавља за сваки нови чвор), а вредности које се додељују зависе од димензија пиксела и могу бити произвољне. Додељене вредности за вертикално и хоризонтално померање се разликују од оних које се додељују у случају дијагоналног померања. На местима где је декларисана препрека практикује се да додељена вредност цене помераја буде велика, како би била искључена приликом одабира путање.

На слици 4.5.4 дат је пример доделе вредности цене помераја. Окружење приказаног чвора је преузето из предходног примера. У примеру је задржана оцена најкраћег пута у односу на циљни чвор добијена Еуклидском нормом. У случају вертикалног (хоризонталног)

кретања додељена је цена помераја $g(n)=1$. За вредност цене помераја у случају кретања у дијагоналним правцима додељена је вредност $g(n)=1.4141$. Ова вредност је заправо резултат стварне дистанце по дијагонали јер је пиксел квадратног облика и димензија 1×1 (примена Питагорине теореме). На местима где је лоцирана препрека употребљене су исте вредности као у случају да препреке нема и додатно помножене са 100 (произвољно одабрана вредност која ће обезбедити велику цену помераја).

$h(1,1) = 5.6569$ $g(1,1) = 1.4141$	$h(1,2) = 5$ $g(1,2) = 1$	$h(1,3) = 4.4721$ $g(1,3) = 1.4141$
$h(2,1) = 5$ $g(2,1) = 1$	$h(2,2) = 4.2426$ $g(2,2) = 0$	$h(2,3) = 3.6056$ $g(2,3) = 1$
$h(3,1) = 4.4721$ $g(3,1) = 1.4141$	$h(3,2) = 3.6056$ $g(3,2) = 100$	$h(3,3) = 2.8284$ $g(3,3) = 141.41$

Слика 4.5.4: Додељене вредности цене помераја за окружење чвора на позицији (2,2)

На основу ове две карактеристике функционише алгоритам A^* . Збир цене помераја и оцене најкраћег пута у односу на задати циљ означава се са $f(n)$ према [4] и назива се оцена најкраћег пута од стартног до циљног чвора. Поменути збир чини још један сабирак – пређени пут. Пређени пут представља збир цене помераја од стартног до тренутног чвора, односно збир цене помераја свих чворова који ће се налазити у оптималној путањи између оних који су задати. Једначина која описује предходно наведено има облик:

$$f(n) = h(n) + g(n) + \text{predjeni_put} \quad (44)$$

Како је оцена најкраћег пута између задатих чворова различита и карактеристична за сваки чвор може се узети као меродавна при сврставању у скуп у коме су садржани сви они кроз које треба проћи на путу од стартног до циљног чвора (пиксела). Алгоритам у тренутном окружењу тражи чвор са најмањом оценом најкраћег пута који ће усвојити за локални минимум. Одабрани чвор ће бити сврстан у поменути скуп, а у наставку ће бити испитана његова околина у потрази за новим локалним минимумом. Процедура се понавља све док се не достигне циљни чвор. Оцена пређеног пута од стартног па до циљног чвора за циљни чвор једнака је само пређеном путу јер су остале две карактеристике једнаке нули. У тренутку када се за локални минимум усвоји вредност циљног чвора претрага је завршена, а решење тј. оптимална путања представљена је преко чворова који се налазе у скупу.

Следећом сликом 4.5.5 биће приказан пример који ће демонстрирати описани поступак. Приказано окружење је димензија 6×6 . Облик пиксела је квадрат димензија 1×1 при чему се јединице не узимају у обзир. Пиксел обојен зеленом бојом представља стартну, црвеном обојен представља циљну позицију, док сивом обојен представља место на којем се налази препрека у окружењу. За оцену најкраћег пута мереног од задатог циља за чворове окружења примењена је Еуклидска норма. Цена помераја је произвољно одабрана и за вертикално (хоризонтално) померање износи $g(n)=0.5$, за дијагонално померање износи $g(n)=0.7072$, а на местима где се налази препрека цена помераја се добија тако што се већ усвојена вредност увећа за 100. Као што се може видети на слици 4.5.5, за неке чворове постоји само оцена најкраћег пута, док за друге унутар пиксела се налазе различите вредности. Те вредности представљају (гледајући одозго на доле) оцена најкраћег пута мереног од задатог циља, цену помераја, пређени пут и оцену најкраћег пута од стартног до циљног чвора респективно. Стрелицама је описано померање са чвора на чвор. Примећујемо да су за околинду сваког изабраног чвора израчунати наведени параметри.

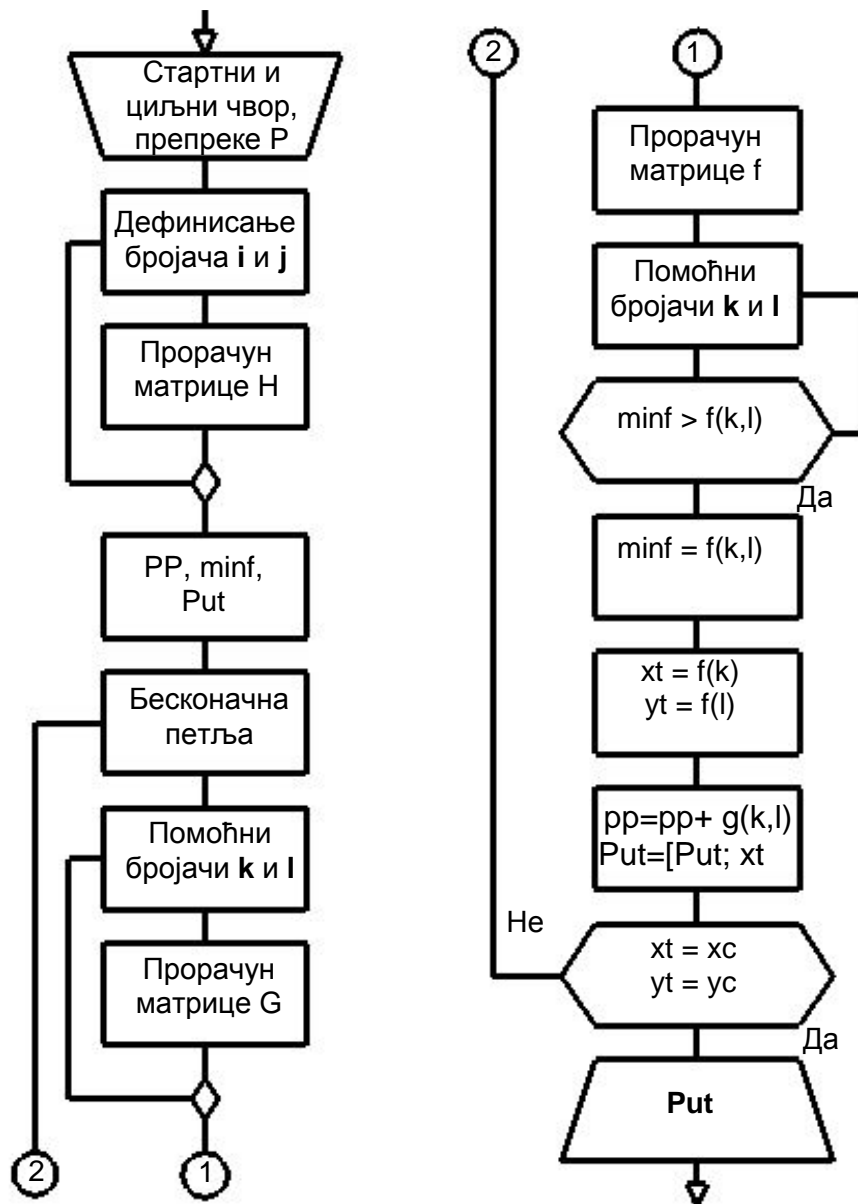
5.6569 0.7072 0 6.3641	5 0.5 0 5.5	4.4721 0.7072 0 5.1793	4.1231 0.7072 0.5 5.3303	$h(1,5)=$ 4	$h(1,6)=$ 4.1231
5 0.5 0 5.5	4.2426 0 0 4.2426	3.6056 0.5 0 4.1056	3.1623 100.5 0.5 104.16	3 0.7072 1.2072 4.9144	$h(2,6)=$ 3.1623
4.4721 0.7072 0 5.1793	3.6056 100.5 0 104.11	2.8284 100.7072 0 103.54	2.2361 0.7072 0.5 3.4433	2 0.5 1.2072 3.7072	2.2361 0.7072 1.9144 4.8577
$h(4,1)=$ 4.1231	$h(4,2)=$ 3.1623	2.2361 0.7072 1.2072 4.1505	1.4142 0.5 1.2072 3.1214	1 0.5 1.4144 2.9144	1.4142 0.5 1.9144 3.8286
$h(5,1)=$ 4	$h(5,2)=$ 3	$h(5,3)=$ 2	1 0.7072 1.9144 3.6216	0 0 2.4144 2.4144	1 0.7072 1.9144 3.6216
$h(6,1)=$ 4.1231	$h(6,2)=$ 3.1623	$h(6,3)=$ 2.2361	$h(6,4)=$ 1.4142	$h(6,5)=$ 1	$h(6,6)=$ 1.4142

Слика 4.5.5: Приказ путање од почетног до циљног чвора са прорачуном неопходних вредности за одређене чворове

Према већ описаном алгоритму, потребно је пронаћи тренутни локални минимум, који се у нашем случају за окружење око стартног пиксела налази на чвору који према координатама има позицију (2,3). У наставку из окружење тог чвора проналази се нови локални минимум на позицији (3,4). Даље за окружење чвора на позицији (3,4) усваја се локални минимум на позицији (4,5) и на крају циљни чвор (5,5). Могуће је приметити да се оцена најкраћег пута од стартног до циљног чвора $f(n)$ и оцена најкраћег пута мереног од задатог циља $h(n)$ смањује како смо ближи задатом циљном чвору, док се пређени пут повећава преласком са чвора на чвор.

4.5.2. Реализација и имплементација A^* алгоритма у моделу кретања мобилног робота LEGO Mindstorms NXT.

Како је већ предходно речено, A^* алгоритам има широку употребу, што нас води до закључка да је у зависности од потреба развијен одговарајући алгоритам а на основу њега и програм у неком програмском окружењу. У наставку на слици 5.5.6, биће приказан алгоритам пропраћен адекватним коментарима који је реализовала Група1 на предмету Интелигентни технолошки системи, модула Производно машинство. Програм је реализован у софтверу *MATLAB*. Обзиром да софтвер *MATLAB* оперише са матрицама, неопходно је било сва решења, уколико је то могуће, представити матрично или у низу, како би се омогућило обављање задатих команди у складу са алгоритмом.



Слика 4.5.6: Приказ алгоритма према ком је урађен програм у софтверу *MATLAB*

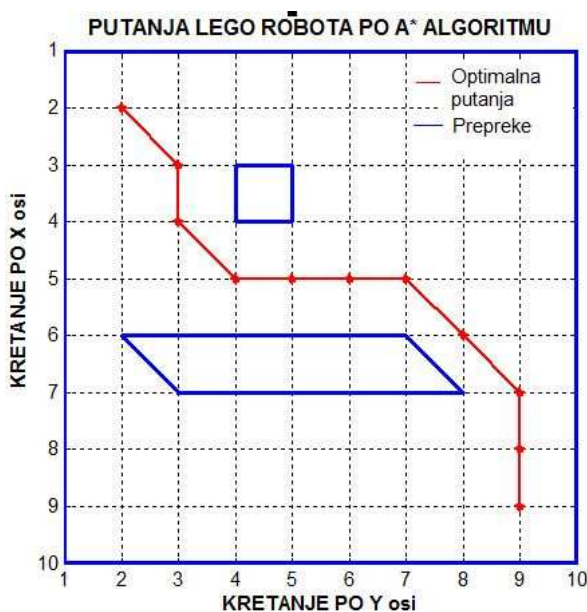
У наставку на основу слике 4.5.6, биће поступно објашњени сви делови алгоритма. Као што можемо видети, информације са којима располажемо на почетку су координате стартног и циљног чвора, као и места на којима су лоциране препреке. Матрица P је матрица у коју се смештају подаци о постојању препрека. Ова матрица је већа од дефинисаног окружење чиме се обезбеђује да ће алгоритам спроводити претрагу све док је испуњен услов да се стартни и жељни чвор налазе унутар ове матрице. Након тога потребно је израчунати и приказати оцене најкраћег пута мерене од задатог циља за све чворове у окружењу које је у предходном кораку дискретизовано. Израчунате оцене најкраћег пута се налазе у матрици H . Она се рачуна само једном а потребна нам је у наставку. Даље, дефинише се почетна (јер мора да има неку вредност у односу на коју ће се позивати) и тренутна (проласком кроз петљу ова вредност се мења) вредност параметра $minf$ који ће нам послужити у једном тренутку за проверу и избор следећег (наредног) чвора који ће бити посећен. У наставку потребно је подесити вредност пређеног пута (PP), као и прве чланове матрице Put . Матрица Put је матрица коју добијамо као излаз из датог алгоритма. Чланови ове матрице су чворови кроз које је неопходно проћи крећући се од почетног (стартног) ка циљном (жељеном) чвору. У почетном тренутку једини чланови ове матрице су координате стартног чвора. Бесконачном петљом обезбеђујемо да алгоритам проверава све задате тачке до

тренутка када дође до координата циљног чвора. У наставку алгоритма долазимо до тренутка када треба да добијемо матрицу цене помераја G . Она у себи садржи цене помераја за све посећене чворове и њихову околину. При сваком рачуну цене помераја за одабрани чвор и његову околину постоје три односно пет чворова чију цену помераја знамо на основу предходног прорачуна (изузима се случај стартног чвора). Стога треба напоменути да је чворовима могуће само једном доделити односно израчунати вредност цене помераја. У случају да се ова напомена занемари и дозволи рачун цене помераја више пута за последицу има да добијена путања на крају неће бити најкраћа односно оптимална. Разлог за ту констатацију је утицај на главни збир на основу ког се бира наредни чвор (о поменутом збиру биће више речи у наставку). Параметар који нам је неопходан за наставак алгоритма (програма) је оцена најкраћег пута од стартног до циљног чвора $f(n)$. Ова оцена се добија према једначини (44). У овом случају имамо малу промену, тј сабирак више. Тај сабирак је настао раздвајањем цене помераја и могуће препреке која се налази на том чвору. Разлог за раздвајање се проналази у лакшој реализацији програма. Проширена једначина (44) дата је следећом једначином (45):

$$f(k,l) = h(k,l) + g(k,l) + pp + P(k,l) \quad (45)$$

Добијене вредности поступком сабирања се чувају у матрици f на основу чијих ће вредности бити вршен избор наредног чвора. Долазимо до дела алгоритма у коме се бира наредни посећени чвор. Упоредивањем вредности сваког чвора из окружења са декларисаним минимумом ($minf$) тражимо ону вредност која је мања од тренутне вредности минимума. Вредности са којима се $minf$ пореди су из матрице f . Када се појави вредност чвора која је мања од тренутне проглашавамо је за тренутни (локални) минимум, прелазимо у тај чвор, а његове координате уписујемо у матрицу Put . Пређени пут се повећава за вредност цене помераја чвора у коме смо предходно били. Процедура се понавља све док се не стигне до жељене циљне тачке. Када се то деси услов за престанак рада алгоритма (излазак из бесконачне петље) је испуњен а као решење се добијају координате чворова који чине путању између задатих чворова (пиксела) и графички приказ најоптималније путање.

На слици 4.5.7 приказана је верификација описаног програма на произвољно одабраном примеру. Величина окружења је 10×10 . Пиксел је облика квадрата димензија 1×1 , не узимајући у обзир јединице. Стартни чвор је на позицији $[2,2]$ у окружењу а циљни на позицији $[9,9]$ у окружењу. Места препрека су на позицијама $[3,4]$, $[3,5]$, $[4,4]$, $[4,5]$, $[6,2]$, $[6,3]$, $[6,4]$, $[6,5]$, $[6,6]$, $[6,7]$, $[7,3]$, $[7,4]$, $[7,5]$, $[7,6]$, $[7,7]$, $[7,8]$.



Слика 4.5.7: Графички приказ оптималне путање у софтверу MATLAB

Обраћајући пажњу на описан алгоритам и узимајући у обзир правила и синтаксу коју условљава софтвер *MATLAB* приступа се писању програма. Обзиром на захтеве пројектног задатка написан програм нема никакву функцију када се посматра самостално, па га је из тог разлога неопходно имплементирати у главни модел кретања мобилног робота *LEGO Mindstorms NXT*. A^* алгоритам није убачен у главни програм директно, већ се позива као функција. Ова минимална измена извршена је због прегледности главног програма, као и због чињенице да нас у овом тренутку занима само излаз који генерише A^* алгоритам. Почетна као и циљне тачке (чворови) су задати у главном програму. Напомиње се да је почетна тачка увек иста док се циљне тачке мењају у зависности од технолошког поступка. Алгоритам A^* , како је већ познато, у тренутку позивања из главног програма генерише путању, односно координате чворова кроз које ће робот проћи како би дошао до циља. Поменуте координате се чувају у матрици *Put*, која ће бити дата на излазу заједно са графичким приказом путање (налик ономе са слике 4.5.7). Након извршења A^* алгоритма наставља се даље извршење главног програма.

4.6 Развој симулационог модела у AnyLogic софтверском окружењу

У овој фази пројектног задатка потребно је развити симулациони модел постојећег диспозиционог плана погона за производњу и монтажу *SIVACON* електро ормана у предузећу "XY", а након тога, на основу резултата симулације, предложити нови диспозициони план технолошког окружења.

4.6.1 Теоријске поставке симулације

Симулација је имитација рада реалног процеса или система током времена и омогућава генерисање вештачке историје и посматрање те вештачке историје у циљу извођења закључака о радним карактеристикама система. Понашање система које се мења током времена проучава се развијањем симулационог модела [2].

Модел представља концептуални оквир који описује систем и најчешће се јавља у облику скупа математичких и логичких претпоставки или кроз скуп симболичких релација између ентитета [2].

Систем се дефинише као група објеката која је уједињена како би кроз одређену интеракцију испунила одређени задатак [2].

Циљ моделирања и симулације према [2]:

- На моделу се може испитати широк спектар "шта – ако" питања о реалном систему;
- Могу се симулирати потенцијалне измене система и предвидети њихов утицај на систем;
- Могу се пронаћи адекватни параметри система пре имплементације;
- Проучавање модела уместо реалног система је најчешће много лакше, брже, јефтиније, сигурније.

Симулација се према [2] може користити као:

- Аналитички алат за предвиђање ефеката промена;
- Пројектантски алат за предвиђање перформанси новог система;

Разлози за примену симулације према [2]:

- Симулација омогућује проучавање интеракција унутар комплексног система;
- Могу се симулирати и изучавати промене информационе структуре, организационе промене и промене окружења;
- Симулациони модел подстиче унапређење знања о систему;
- Проналажење важних улазних параметара кроз промену симулационих улаза;
- Експериментисање са новим пројектима и стратегијама пре имплементације;
- Симулирање различитих могућности машина ради одређивање потреба;
- Симулациони модели за обуку омогућавају учење без већих трошкова;
- План се може визуелизовати кроз анимирани приказ;
- Савремени системи (фабрике, постројења, сервиси, итд.) су толико комплексни да се интеракције унутар система могу обрађивати само кроз симулацију.

4.6.2 Симулација рада технолошког система

Симулација тока материјала кроз производни погон обављена је у програмском пакету *AnyLogic*. У симулацији су обухваћена четири карактеристична дела која се производе у овом предузећу са нормираним припремним временима и временима обраде која су дефинисана технолошким поступком. У табели 4.6.1 дато је трајање технолошких операција за репрезентативне делове.

Табела 4.6.1 : Трајање технолошких операција (у секундама) за репрезентативне делове				
Делови	Transport fuse (део 1)	Mainbusbar support (део 2)	Suport d800 (део3)	Busbar 2 L1 (део 4)
Машина				
Маказе за сечење	-	-	-	-
CNC машина за пробијање и просецање	6+2	-	-	-
CNC хидраулична апкант преса	2+2	-	4+4+4+2+2	-
Машина за исецање профила	2,5	-	32+37+6+3,5+18	-
Стубна бушилица	-	10+6	-	-
Кружна тестера	-	12	-	-
Оштрилица алата	-	-	-	-
Линија за обраду делова од бакра		-	-	2+23+4+7+8+5

Узимајући у обзир претходну таблицу, извршена је симулација неколико модела. У првом моделу који је симулиран није узето у обзир време транспорта материјала између складишта и машина, а такође ни време транспорта делова између самих машина (не постоје конвејери), тако да овај модел обухвата време потребно за припрему делова за обраду, само време обраде, и време потребно да се део склони са машине.

Као што се из таблице 4.6.1 види, део 1 и део 3 се обрађују на истим машинама а то су машине М3 и М4. Овом симулацијом није обухваћено да се више делова обрађује на истим машинама јер су постојали одређени проблеми који су се огледали у томе да није могуће подесити да се два различита дела обрађују различитим временима на једној машини. Када

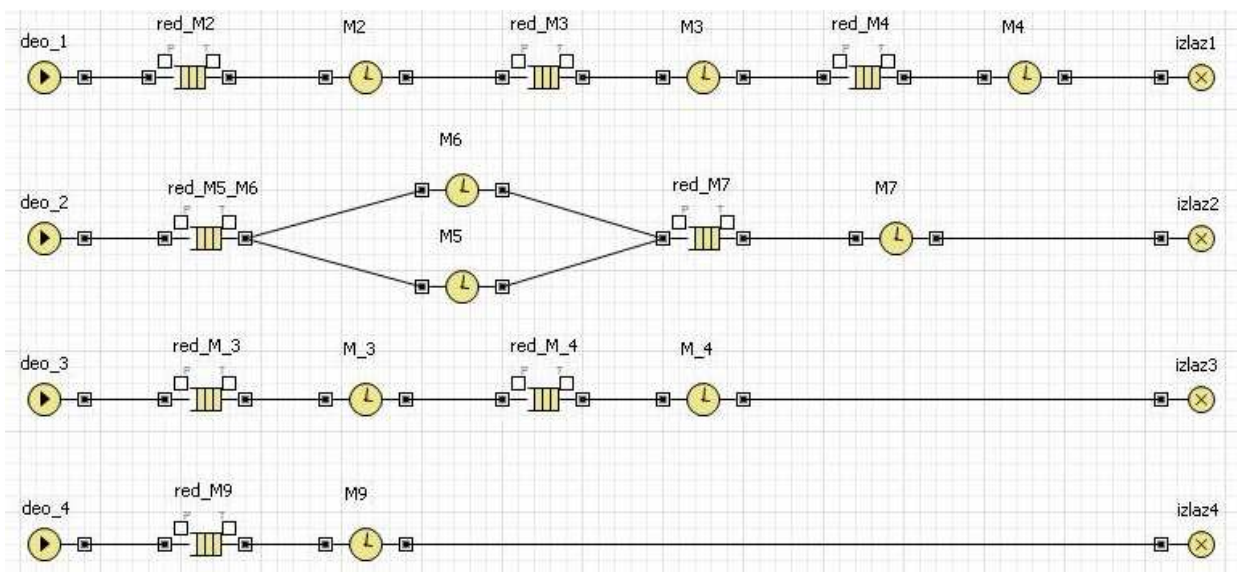
је, дакле, покушано да се два различита дела обрађују различитим временима, како налаже технолошки поступак, симулациони програм је делове обрађивао увек истим временима, тако да симулација и није толико веродостојна, али опет, и поред ових недостатака, могу се извући одређени закључци о понашању система.

Да би овај проблем у одређеној мери био решен, у симулацију су убачене машине M3 и M4 на два места (слика 4.6.1), односно на линије за производњу дела 1 и дела 3. Треба опет напоменути да, како је и претходно речено, у погону не постоје по две машине M3 и M4, али се за потребе симулације оне уводе јер се ток материјала симулира за сваки део посебно.

Од ентитета, у симулацији су коришћени следећи:

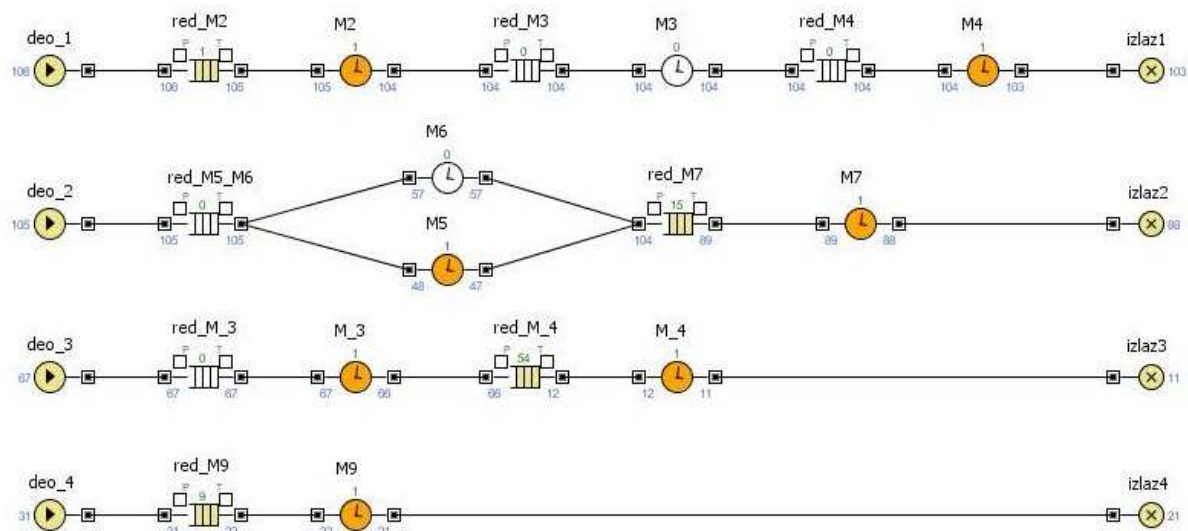
- *Source* – представља улаз, у нашем случају делове који пристижу у одређеном временском периоду;
- *Delay* – ентитет који је у пројекту коришћен као машина са дефинисаним временима припреме и обраде материјала;
- *Queue* – представља ред делова испред машине који чекају на обраду;
- *Sink* – представља излаз, у нашем случају отпремање материјала до неких од складишта.

4.6.2.1 Симулациони модел без анализе транспортних токова



Слика 4.6.1: Симулациони модел без анализе транспортних токова

Симулација је тако подешена да симулира рад једне смене која траје 8 часова (28.800 sec.). Као што се са слике 4.6.1 види, испред сваке машине налази се ред чекања за ту машину. Махине M5 и M6 (Стубне бушилице) везане су паралелно јер су то у ствари исте машине, тако да део може да се нађе на било којој од њих.



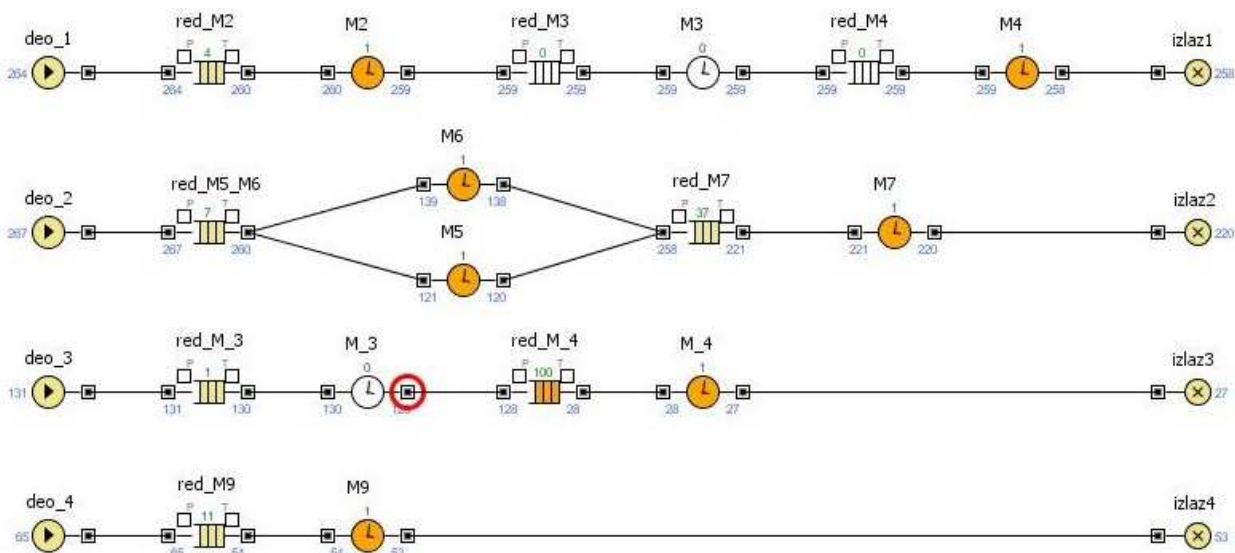
Слика 4.6.2: Симулациони модел у току симулације

На слици 4.6.2 приказан је ток симулације кретања делова кроз погон на половини симулације. Види се да је део 1 направљен у 103 примерака. На линији за производњу дела 1, нема редова чекања, јер је време израде дела 1 релативно кратко на све три машине у овој линији. Укупно је из улазног складишта полуфабриката (поз. 32) слика 4.6.2 испоручено 106 делова 1.

Део 2 израђен је у 88 примерака. Маchine M5 и M6 представљају исте машине (стубне бушилице) па су везане паралелно, тако да се део 2 из реда чекања за машине M5 и M6 наизменично распоређује на неку од ове две машине, у зависности од тога која је тренутно слободна. Симулација је показала да је на машини M5 обрађено 47 делова, с тим да се један део тренутно обрађује, док је на машини M6 обрађено 57 делова. У реду чекања за машину M7 налази се 15 делова. Овај ред је створен због тога што се делови са машина M5 и M6 сустичу у ред чекања за машину M7. На машини M7 обрађено је 88 делова, док се један део тренутно обрађује. Делови се допремају из улазног складишта полуфабриката од лима (поз. 32) слика 4.6.2

Део 3 израђен је у 11 примерака. Овако мали број произведених делова и велики ред чекања на машини M4 (54 дела) доводи се у везу са великим припремним и производним временима на машини M4. Време долазака делова 3 је смањено баш због ових великих времена потребних за припрему дела и обраду на машини. Као што се види са слике 4.6.2, на улазу је из складишта полуфабриката од лима (поз. 32) слика 4.6.2 доведено 67 делова. На машини M3 обрађено је 66 делова, док се један део тренутно обрађује, а редова на машини M3 нема.

Део 4 израђен је у 21-ом примерку. Као и код дела 3, и део 4 је израђен у релативно малом броју примерака, што је, опет, условљено великим временом обраде на машини M9. У реду за машину M9 налази се 9 делова, један део се тренутно обрађује, а на улаз је из складишта полуфабриката од лима (поз. 32) слика 4.6.2 доведен 31 део.



Слика 4.6.3: Симулациони модел након завршене симулације

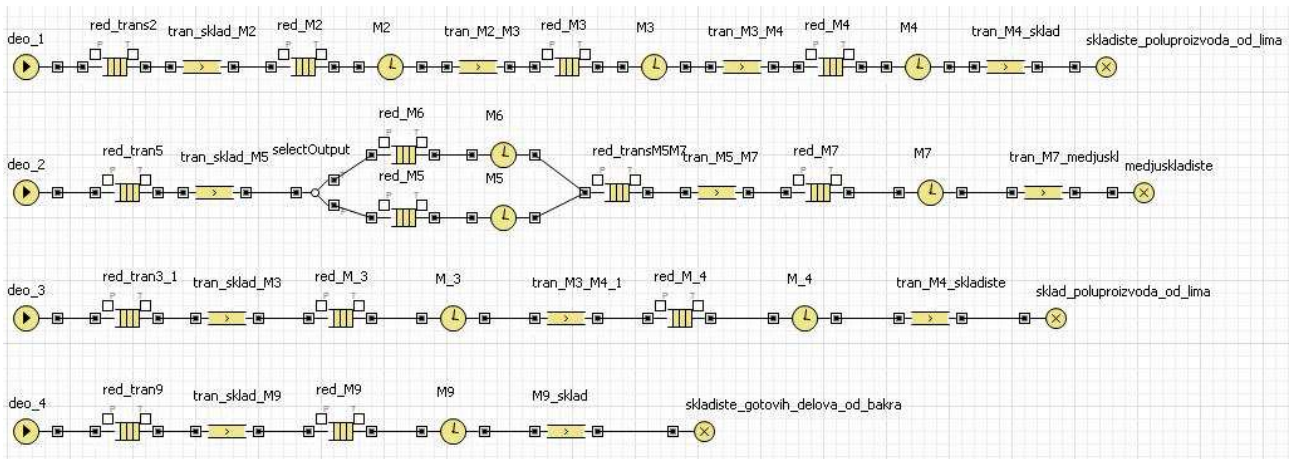
На слици 4.6.3 приказан је модел добијен на крају симулације, односно након 2689 сек. од почетка рада симулације. Симулација је након истека овог времена аутоматски прекинута, јер је дошло до загушења на реду чекања за машину М4 у линији за производњу дела 3. Видело се још на половини симулације да овај ред чекања има тенденцију да се препуни због горе у тексту наведених разлога (велико време обраде дела 3 на машини М4).

На осталим линијама за производњу делова нису се појавили слични проблеми у симулацији, тако да се може извући следећи закључак:

- Део 1 направљен је у 258 примерака са веома малим редом чекања на машини М2 (4 дела);
- Део 2 направљен је у 220 примерака, с тим да се на машини М7 јавио ред чекања од 37 делова. Делови се на машине М5 и М6 релативно подједнако распоређују, а ред чекања на овим машинама износи 7 делова;
- Део 3 направљен је у 27 примерака, док је на реду чекања за машину М7 дошло до загушења делова;
- Део 4 направљен је у 53 примерака, ред чекања на машини М9 је релативно мали и износи 11 делова у реду.

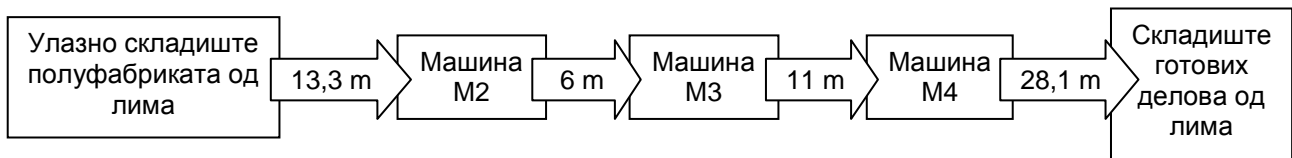
4.6.2.2 Симулациони модел са анализом транспортних токова

Други случај симулације приказан је на слици 4.6.4. У овом случају у обзир су узети транспортни токови. Транспорт међу објектима у симулацији представљен је конвејером а његови параметри се постављају тако што се дефинише раздаљина између објеката између којих се врши транспорт, и брзина кретања између ових објеката. Раздаљина међу објектима одређена је са цртежа *Layout*-а производно-монтажног погона предузећа „ХУ“ (слика 2.1.1), док је за брзину транспорта дефинисана брзина кретања ручног виљушкара која износи око 1 m/s. Дефинисањем ове две вредности добија се време које је потребно да се део транспортује између машина или између машина и складишта.

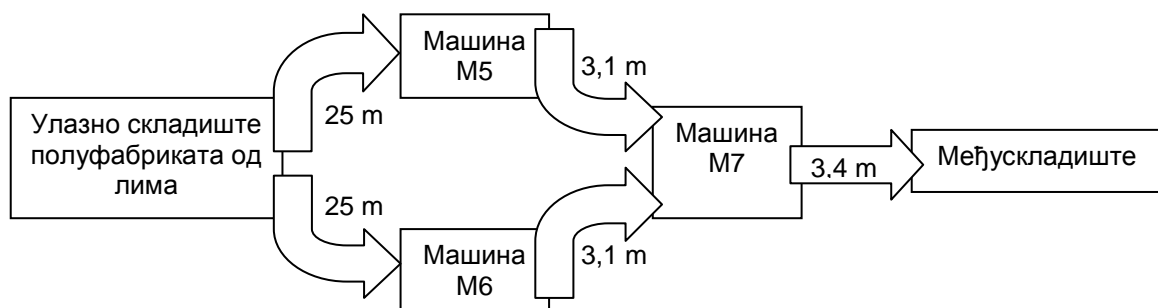


Слика 4.6.4: Симулациони модел са анализом транспортних токова

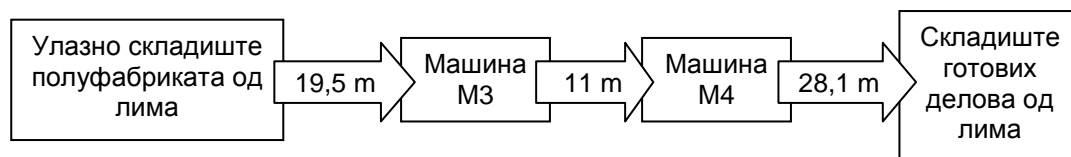
На сликама које следе (4.6.5 – 4.6.8) графички је приказано кретање делова кроз погон, са тачним назнакама путање делова и раздаљинама које делови остварују да би доспели до одређене машине или одређеног складишта.



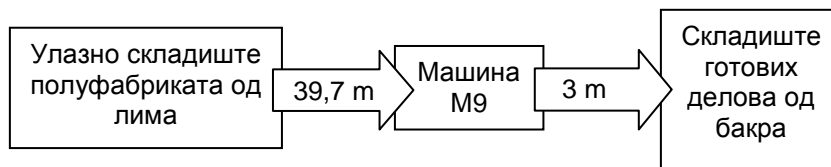
Слика 4.6.5: Кретање дела 1 кроз погон и удаљености између датих објеката



Слика 4.6.6: Кретање дела 2 кроз погон и удаљености између датих објеката

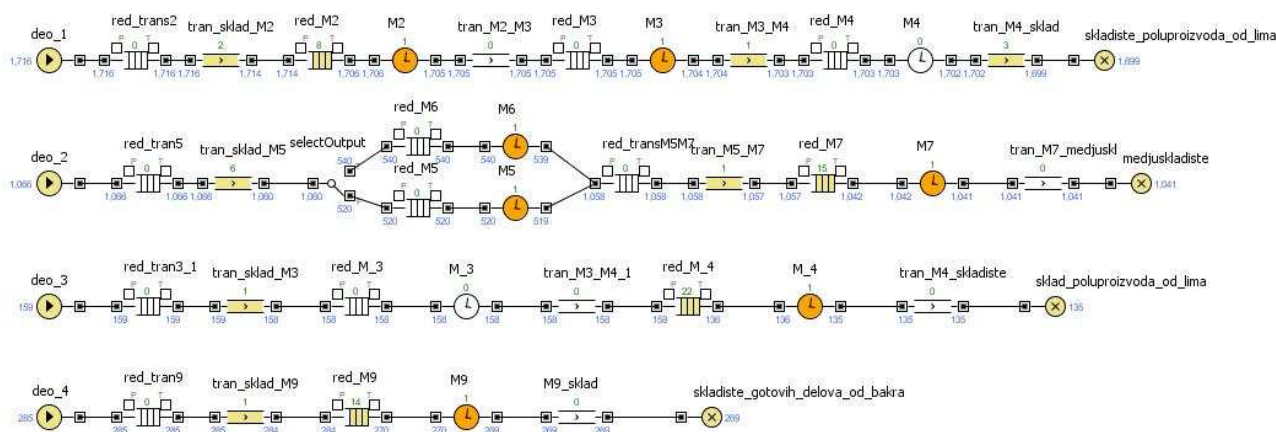


Слика 4.6.7: Кретање дела 3 кроз погон и удаљености између датих објеката



Слика 4.6.8: Кретање дела 4 кроз погон и удаљености између датих објеката

Узимајући у обзир горе наведене параметре, на следећој слици 4.6.9 приказан је ток симулације кретања делова кроз погон који је затечен у 14.000-ом секунду симулације (на половини радног времена).



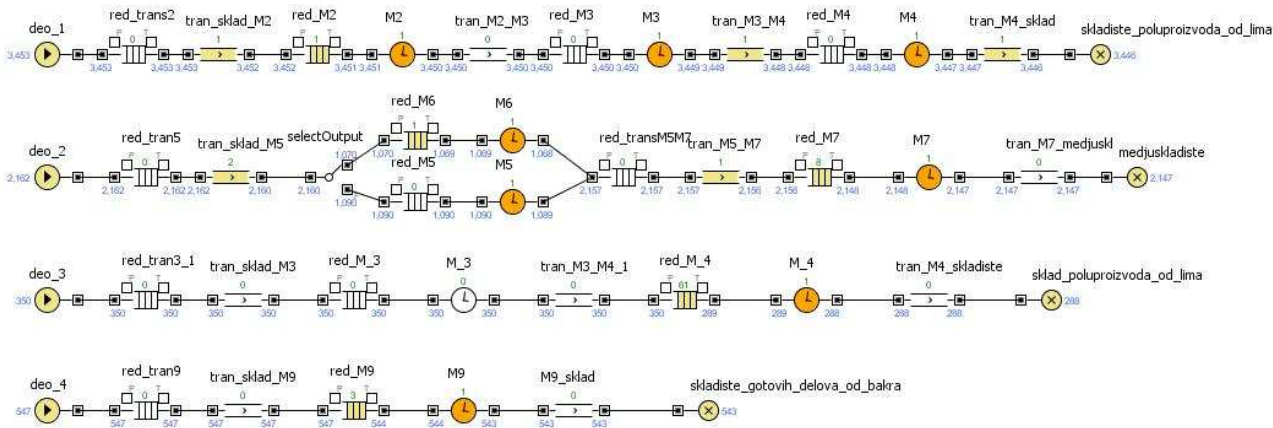
Слика 4.6.9: Симулациони модел са анализом транспортних токова (у току симулације)

Са слике 4.6.9 се види да је део 1 израђен у 1.699 примерака. На линији за производњу дела 1 ред чекања се јавио на машини М2 и износи 8 делова. На осталим машинама у овој линији нема редова чекања.

Део 2 израђен је у 1.041 примерака и види се да овде постоји ред чекања на машини М7 (15 делова) из истих оних разлога који су наведени у првој симулацији. Делови се релативно подједнако распоређују на машине М5 и М6.

Део 3 израђен је у 135 примерака што се доводи у везу са великим временом припреме и обраде дела 3 на машини М4. Из улазног складишта полуфабриката испоручено је 159 делова. Ред чекања на машини М4 износи 22 дела, док редова чекања на машини М3 нема.

Део 4 израђен је у 269 примерака. На линији за производњу дела 4 у реду чекања на машини М9 чека 14 делова.



Слика 4.6.10: Симулациони модел са анализом транспортних токова (на крају симулације)

На слици 4.6.10 приказан је симулациони модел са анализом транспортних токова на крају симулације која је трајала 8 часова (28.800 секунди).

На линији за производњу дела 1 произведено је 3.446 делова. Из улазног складишта полуфабриката послато је 3.453 дела, што значи да се у систему тренутно налази 7 делова који се или обрађују или налазе у редовима чекања, што после оволико пређеног времена симулације не представља велики број делова. На овој линији нема загушења и уских грла.

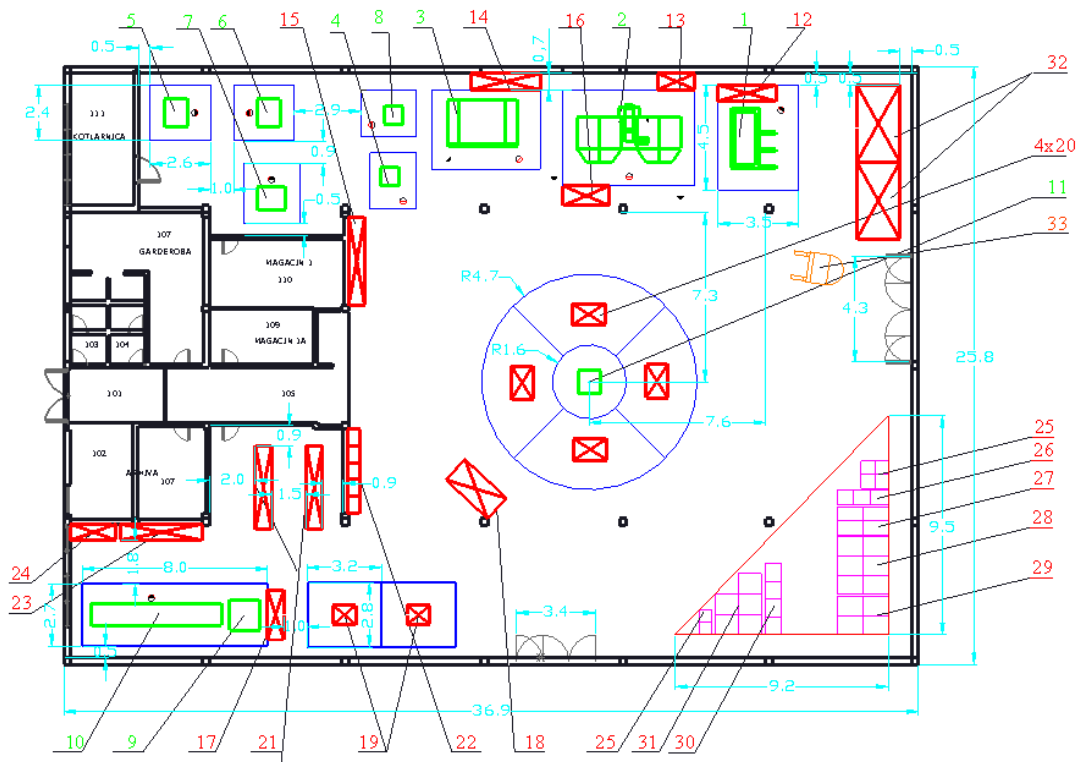
Део 2 произведен је у 2.147 примерака. Из улазног складишта полуфабриката послато је 2.162 дела. На овој линији, ред чекања се појавио на машини М7 и износи 8 делова у реду. Машина М6 обрадила је 1.068 делова а М5 1.089 дела. На ове две машине нема редова чекања.

Део 3 направљен је у 288 примерка. Из улазног складишта полуфабриката испоручено је 350 делова, што значи да се тренутно у систему налази 62 дела. У реду чекања за машину М4 налази се 61 део. Машина 4 представља потенцијално уско грло у овом производном систему. На машини М3 нема редова чекања.

Део 4 направљен је у 543 примерака. На машини М9 постоје 3 дела у реду чекања. Обрада се изводи релативно брзо, тако да на овој линији нема уских грла.

4.6.2.3 Пројектовање новог диспозиционог плана технолошког окружења

На основу претходних симулација и анализе технолошког окружења, предложен је нови диспозициони план предузећа, с тим да су постојећи машински капацитети заджани. Пошто технолошки поступак производње делова мора да буде испоштован, главни критеријум за израду новог диспозиционог плана био је смањење транспортних путева међу машинама, као и између складишта и машина. Транспорт се у овом случају уместо ручним виђушкарком обађа мобилном роботом. На слици 4.6.10 дат је *layout* новог технолошког окружења.



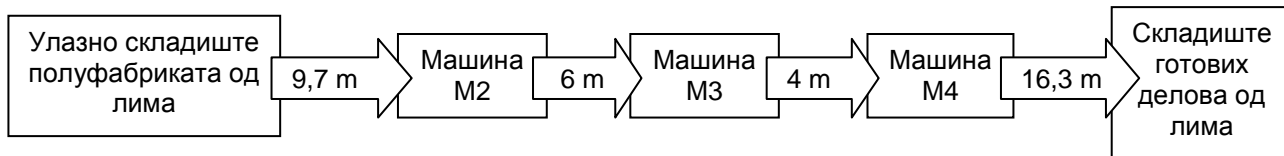
Слика 4.6.11: Layout новог технолошког окружења предузећа “XY”

Упоредивањем слика 2.1.1 и 4.6.11, види се да је на место машина М4 и М8, постављена машина М2, на место машине М2 постављена је машина М3, а на место машине М3 постављене су машине М4 и М8. Овим новим распоредом машина добијено је:

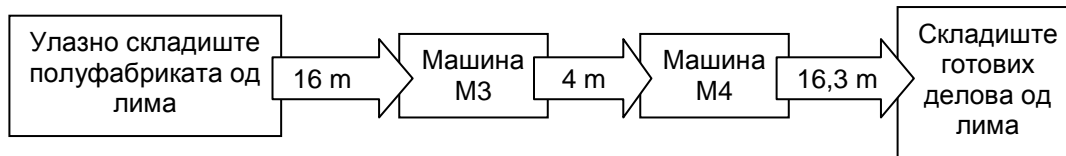
- Део 1 прелази мањи пут, јер је скраћена раздаљина између машине М2 и улазног складишта полуфабриката, а смањена је и раздаљина између машине М4 (последња машина на линији за производњу дела 1) и складишта готових делова од лима.
- Код претходног модела, део 1 је морао са машине М3 да се врати на машину М4 јер ове машине нису биле једна до друге (слика 2.1.1), док је код новог распореда то избегнуто. Део 1 се кроз погон креће редом, јер су у предложеном решењу машине М3 и М4 једна до друге. Самим тим, и удаљеност између машина М3 и М4 је смањена.
- Смањено је растојање између улазног складишта полуфабриката и машине М3, што имплицира да ће транспортно време дела 3 такође бити мање.

На сликама које следе (4.6.12 и 4.6.13) графички је приказано кретање делова кроз погон, са тачним назнакама путање делова и новим раздаљинама које делови остварују да би доспели до одређене машине или одређеног складишта.

Предложеним решењем није промењено стање на линијама за производњу дела 2 и дела 4, јер машине на којима се обрађују ови делови нису мењале своју позицију у оквиру технолошког окружења.



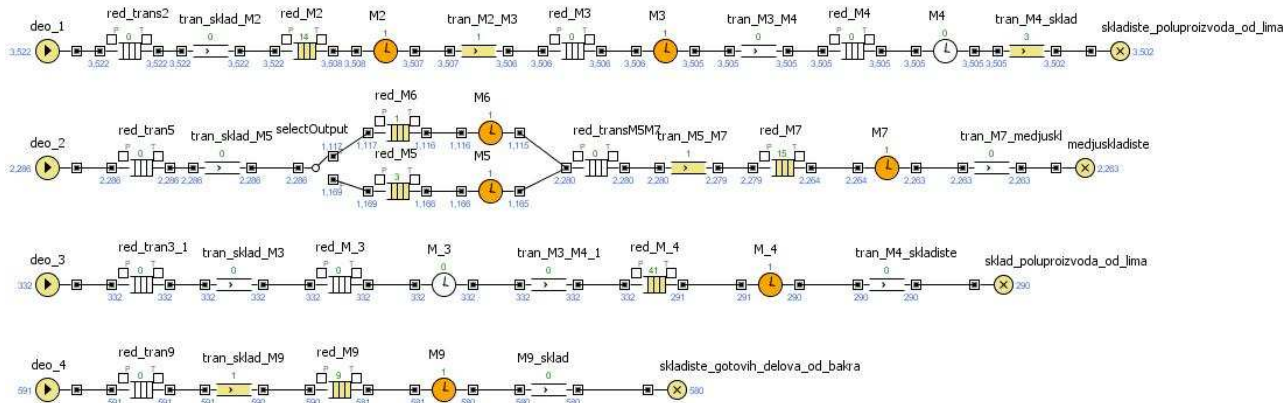
Слика 4.6.12: Кретање дела 1 и удаљености између датих објеката (нови распоред)



Слика 4.6.13: Кретање дела 3 и удаљености између датих објеката (нови распоред)

4.6.2.4 Симулација новог диспозиционог плана технолошког окружења

Као и у претходном случају, симулиран је рад једне смене која траје 8 сати. Симулациони модел је потпуно исти као и у претходној симулацији, с тим што су обзир узете нове раздаљине између машина и складишта. На слици 4.6.14 приказан је симулациони модел новог решења диспозиционог плана на крају симулације.



Слика 4.6.14: Симулациони модел новог решења диспозиционог плана на крају симулације

Са слике 4.6.14 види се да је из улазног складишта полуфабриката испоручено 3.522 делова 1. Направљено је 3.502 дела, што значи да се у систему налази 20 делова. У реду чекања за машину М2 налази се 14 делова док на машини М3 и М4 нема редова чекања.

Део 2 направљен је у 2.263 примерака. Из улазног складишта полуфабриката испоручено је 2.286 делова. У реду чекања на машини М7 налази се 15 делова, док се у редовима чекања на машинама М5 и М6 налази укупно 4 дела.

Део 3 направљен је у 290 примерака. Укупно је из улазног складишта полуфабриката испоручено 332 дела, што значи да се у систему налази 42 дела. У реду чекања на машини М4 налази се 41 део.

На линији за производњу дела 4 произведено је 580 делова. Из улазног складишта испоручен је 591 део, што значи да се у систему налази 11 делова. У реду чекања на машини М9 налази се 9 делова.

На основу изведених симулација може се констатовати да се предложеним диспозиционим планом скраћује време које делови проводе у транспорту од складишта до машина. Такође, смањен је и транспортни пут између самих машина, што је довело до веће продуктивности производног погона.

5. Експериментални резултати реализованог пројектног решења

У претходној тачки детаљно су приказана решења која се односе на појединачне делове пројектног задатка. Као таква имплементирана су у једну функционалну целину која представља главни управљачки код за реализацију основног задатка (кретање мобилног робота унутар експерименталног технолошког окружења), у реалном времену. Овако формиран главни управљачки код на почетку рада захтева дефинисање следећих параметара:

- конструкционе карактеристике робота (растојење између тачкова (b) и полупречник тачкова (r));
- координате контролних тачка (m_{apa});
- координате почетне, међутачака и крајње тачке (x , y);
- дозвољена грешка при позиционрању мобилног робота у задату тачку (d_{max});
- угао између тренутног правца кретања и правца ка наредној задатој тачки за који је неопходно извршити ротацију пре кретања унапред (f_{imax}).

Функционисање управљачког кода, односно реализација кретања мобилног робота према постављеном задатку у реалном времену, биће приказано на једноставном примеру. Мобилном роботу је постављен задатак да изврши кретање од почетне до крајње тачке, са једном међутачком.

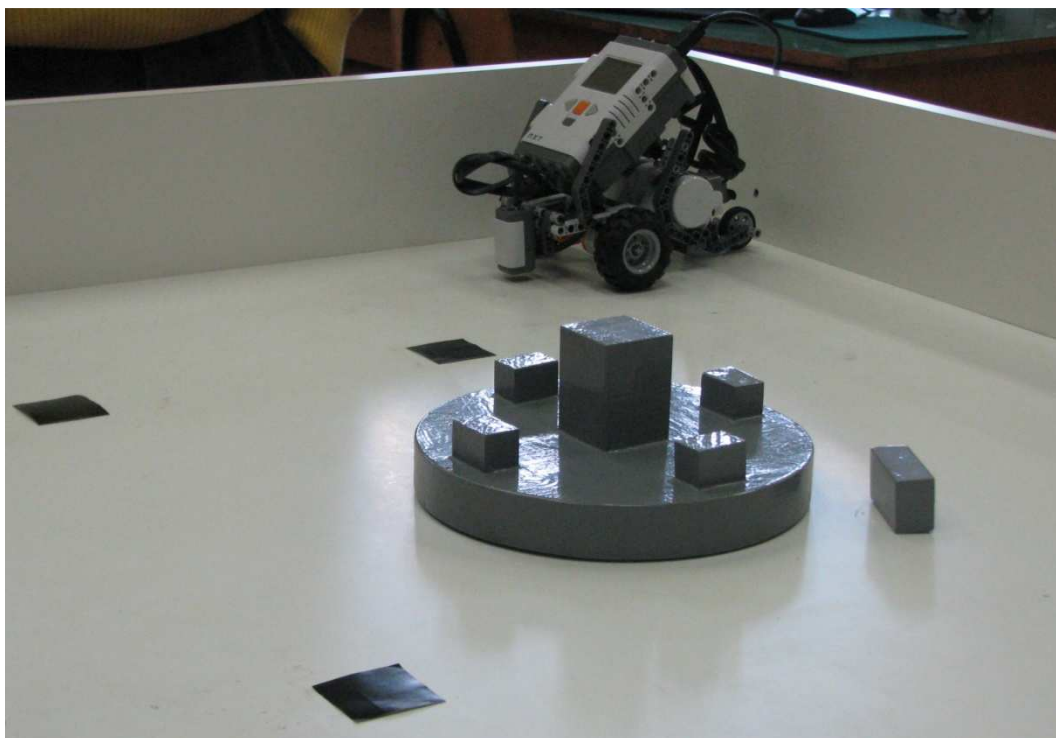
У примеру кретања робота параметри који су претходно дефинисани су:

- $b=9$ [cm];
- $r=2.8$ [cm];
- $m_{apa}=[20\ 14; 42.5\ 14; 42.5\ 56]$ у [cm];
- $x=[2\ 7\ 7]$;
- $y=[2\ 3\ 6]$;
- $d_{max}=5$ [cm];
- $f_{imax}=15^\circ$.

Из задатих параметара уочава се да су на почетку координате тачака задате као вредности врста и колона матрице. Разлог овоме је што се у оквиру A^* алгоритма врши дискретизација експерименталног технолошког окружења у формат квадратне матрице жељених димензија. Такође извршава се и скалирање димензија матрице према димензијама окружења, у конкретном случају померај по x оси има вредност $8,5$ [cm], а померај по y оси 14 [cm]. Пошто су дата и ограничења кретања робота као границе радног окружења, онда позиција у матрици са координатама $x=2$ и $y=2$ практично представља тачку са координатама $x=0$ и $y=0$. На основу наведеног, координате почетне, међутачке и крајње тачке у конкретном примеру су:

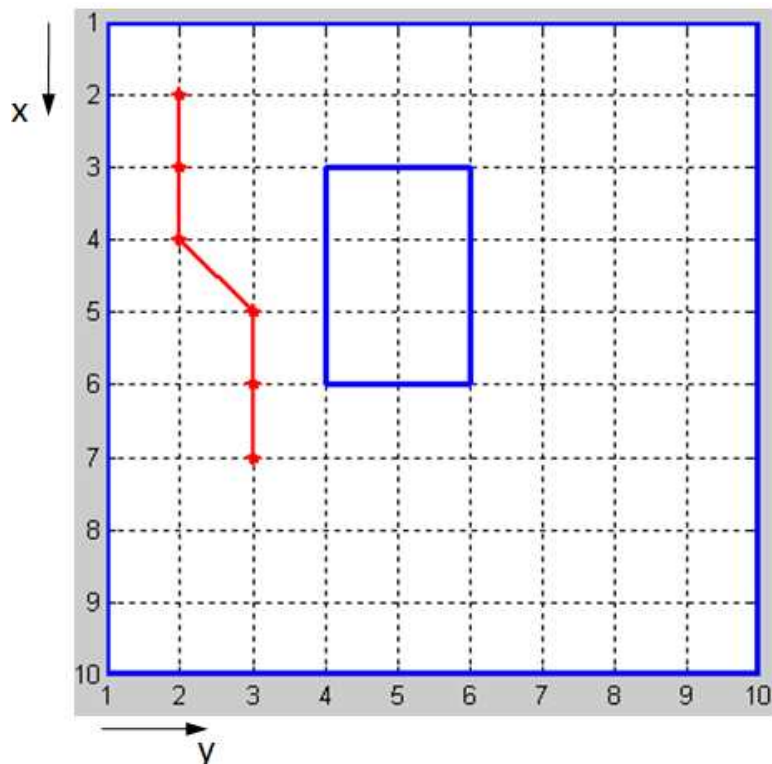
- $x=[0\ 42.5\ 42.5]$ у [cm];
- $y=[0\ 14\ 56]$ у [cm].

Положај робота у почетном тренутку, непосредно пре покретања главног програма, дат је на слици 5.1.1. На овој слици уочавају се и црне квадратне контролне тачке, као и место вршења монтаже готовог производа. Место вршења монтаже узето је у обзир као препрека у A^* алгоритму.



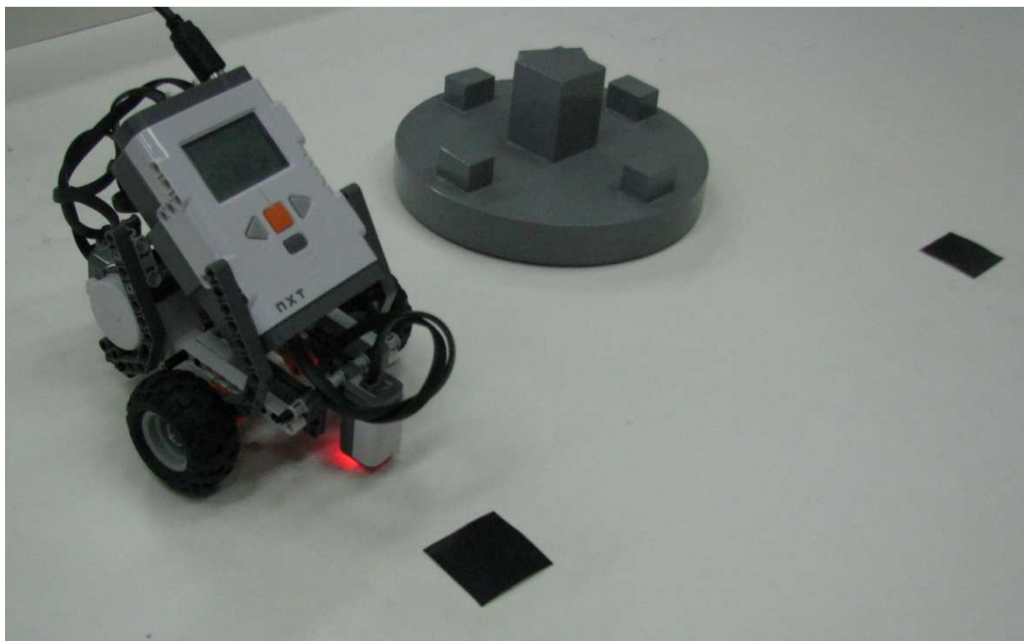
Слика 5.1.1: Положај робота у почетном тренутку

Према претходно дефинисаним улазним подацима, прво се применом алгоритма претраживања A^* одређује оптимална путања од почетне тачке до међутачке, што је приказано на слици 5.1.2.



Слика 5.1.2: Оптимална путања од почетне тачке до међутачке

Одређивањем оптималне путање генеришу се и додатне међутачке (слика 5.1.2) на основу којих се врши кретање. За сваку од тих међутачака, упоређујући тренутну позицију и оријентацију робота са наредном тачком, израчунава се растојање и потребна оријентација робота. Уколико се тренутна оријентација, тј. правац кретања, разликује од потребне за више од fit_{max} тада је потребно извршити ротацију робота пре кретања унапред. У том случају команду за ротацију вратила мотора одређује вештачка неуронска мрежа која је развијена за те потребе. Након спроведене ротације, уколико је за њом било потребе, врши се кретање у напред све док се растојање до наредне тачке не смањи на вредност d_{max} , или мању од ње. На слици 5.1.3 дат је мобилни робот у ток кретања по путањи са слике 5.1.2

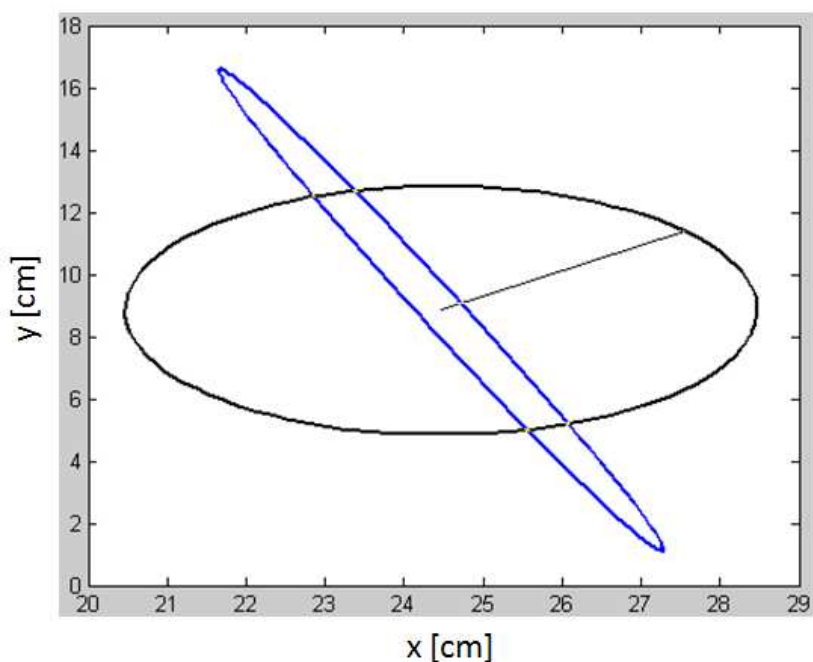


Слика 5.1.3: Мобилни робот у току кретања између почетне и крајње тачке

У току кретања стално се врши израчунавање тренутне позиције (X , Y) и оријентације (θ) у глобалном координатном систему, тј. координатном систему експерименталног технолошког окружења. Ово израчунавање врши се према имплементираном моделу кретања на основу резултата мерења угаоног помераја вратла мотора које врше енкодери. Као што је у пројекту већ више пута напоменуто, мотори не извршавају задате команде идеално нити сензори (енкодери) врше идеално тачна мерења, па се у општем случају израчунати положај разликује од физички оствареног. Из овог разлога у току кретања потребно је вршити корекцију израчунатог положаја робота.

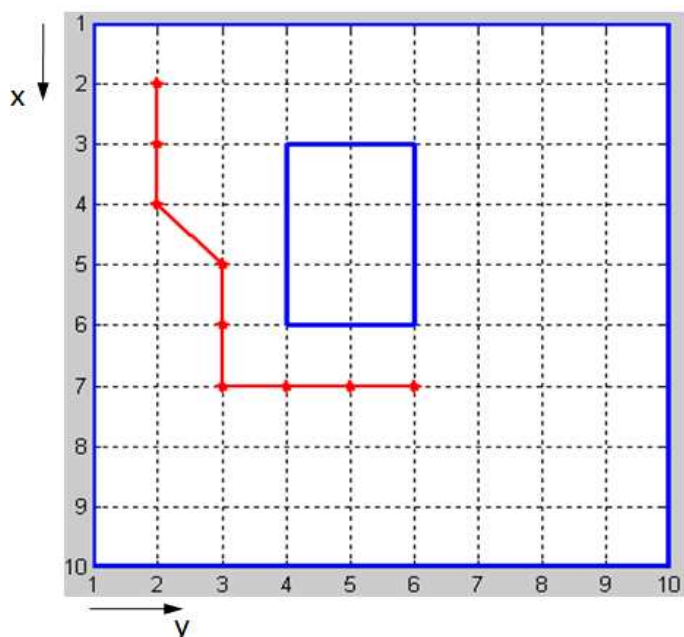
Корекција израчунатог положаја робота врши одређивањем боје подлоге, на основу резултата мерења светлосног сензора, помоћу вештачке неуронске мреже за детекцију контролних тачака у радном окружењу мобилног робота. Важно је напоменути да се мерења помоћу светлосног сензора врше искључиво у тренутку када мобилни робот стоји, пре извршавања наредног кретања. Наиме када се помоћу ВНМ утврди да се робот налази изнад контролне (црне) тачке, тада се на основу израчунатог положаја и координата свих контролних тачака утврди која од њих има највећу вероватноћу да је детектована. Након тога израчунати положај узима координате контролне тачке са највећом вероватноћом. Овај поступак спроводи се помоћу Калмановог филтера и представља локализацију мобилног робота у глобалном координатном систему.

Као резултат овог поступка, у току експлоатације робота, стално се има увид у графички приказ сигурности да се он налази у положају који је израчунат. Ова сигурност опада у току кретања све док се не изврши локлизација. Тај графички приказ за случај када робот није довољно сигуран да се налази у положају који је израчунат дат је на слици 5.1.4, и односи се на тренутак кретања који је приказан слико 5.1.3.



Слика 5.1.4: Графички приказ сигурности мобилног робота за случај када није довољно сигуран да се налази у израчунатом положају

Након што је робот извршио сва потребна кретања према претходно дефинисаном поступку, и нашао се у међутачки која је дефинисана поставком овог једноставног примера, поново се активира A^* алгоритам и врши одређивање оптималне путње до крајње тачке. Целокупна путања одређена A^* алгоритмом дата је на слици 5.1.5.



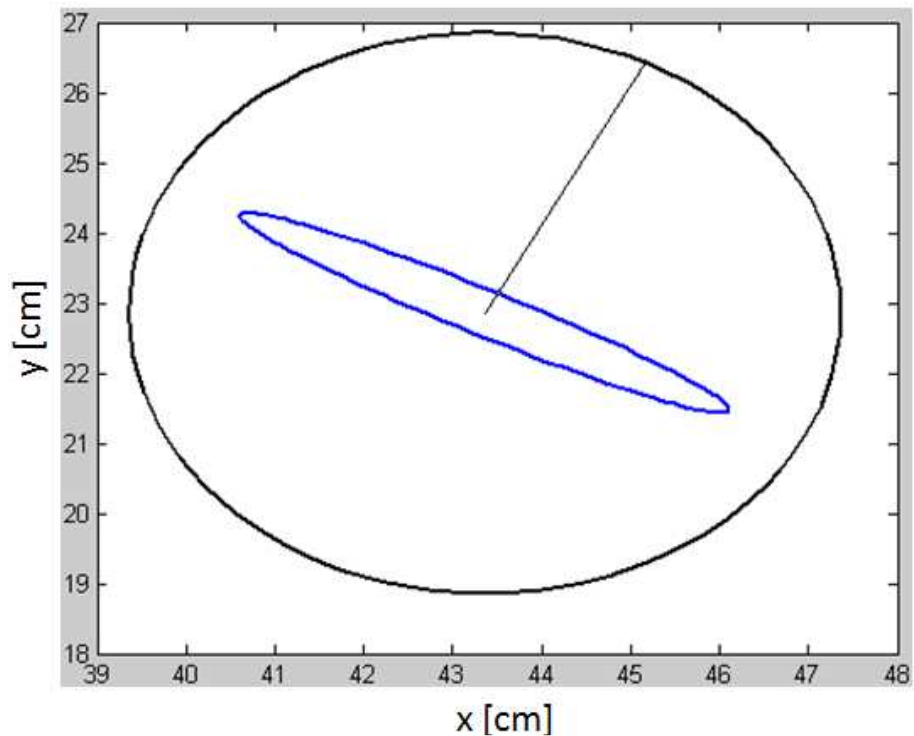
Слика 5.1.5: Целокупна путања мобилног робота према датом примеру

По завршетку одређивања оптималне путање робота од међутачке до крајње тачке наставља се кретање робота ка циљу. Наставак кретања робота од почетне до крајње тачке дат је на слици 5.1.6



Слика 5.1.6: Мобилни робот у фази кретања од међутачке до крајње тачке

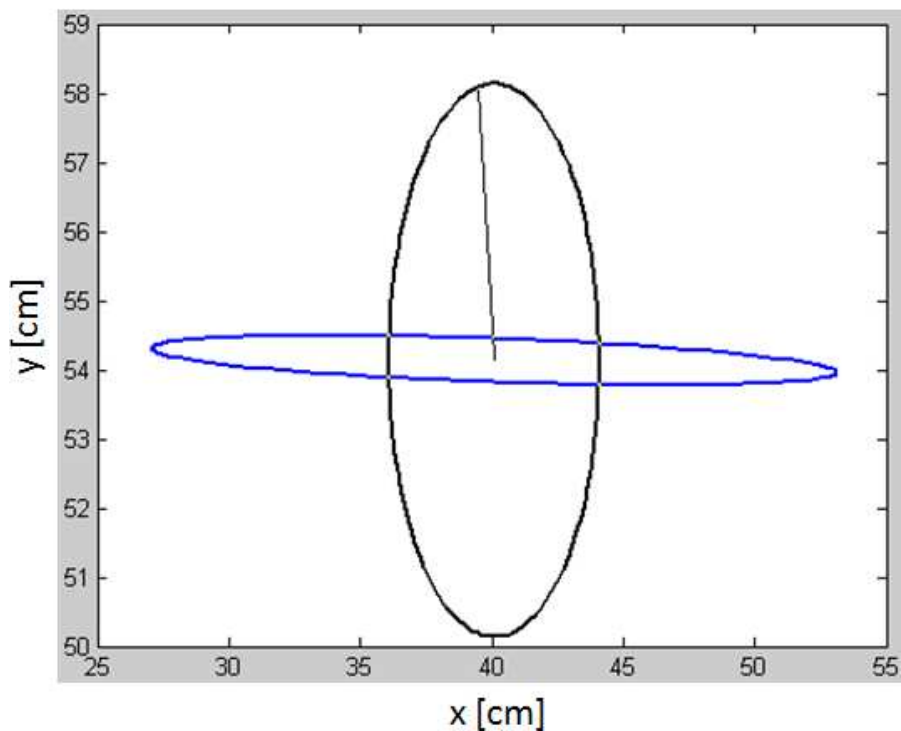
Приликом преласка мобилног робота из првог дела путање у други, детектована је контролна тачка и извршена је његова локализација. Резултат локализације дат је на слици 5.1.7.



Слика 5.1.7: Графички приказ сигурности мобилног робота да се налази у израчунатом положају након извршене локализације

Са преходне слике се може уочити да је знатно смањена плава елипса у односу на случај са слике 5.1.4, тј. повећана је сигурност мобилног робота да се налази баш у положају који је тренутно израчунат.

Након што је извршена локализација, мобилни робот је наставио кретање ка крајњој тачки и успешно га завршио. На слици 5.1.8 да је приказ сигурности мобилног робота да се налази у положају који је израчунат након завршетка кретања. Са те слике се може уочити да је сигурност мобилног робота поново смањена, што је резултат тога да локализација до краја кретања није више ниједном извршена.



Слика 5.1.8: Графички приказ сигурности мобилног робота да се налази у израчунатом положају након завршетка кретања

Иначе, са графичког приказа сигурности мобилног робота може се у сваком тренутку прочитати у ком положају, односно позицији се он налази. Тако да се са слике 5.1.8 читавају вредности $x \approx 40$ [cm] и $y \approx 54$ [cm] што је, када се упореди са задатим координатама крајње тачке, резултат на основу ког се може констатовати да је постављени задатак у конкретном примеру успешно извршен.

6. Анализа и дискусија експерименталних резултата

Из претходне тачке може се констатовати да су експериментални резултати позитивни, али су се приликом реализације пројектног задатка јављали и одређени проблеми.

У вршењу експерименталних проба се јављао проблем оптимизације дозвољених грешака. Конкретно, ради се о успостављању оптималне вредности дозвољене грешке приликом позиционирања робота у току кретања кроз тачке добијене A^* алгоритмом. Такође, и вредност дозвољене угаоне грешке, коју робот прави уколико не изврши ротацију пре кретања унапред мора се оптимизовати како би робот што веродостојније пратио задату путању. Треба нагласити да се под углом ротације робота подразумева тренутна оријентација робота у односу на правац ка циљној тачки.

Одређени проблем представљало је и то што се у општем случају у току кретања међусобно разликују координате тачака које су задате, које су израчунате на основу информација са енкодера и координате тачака у којима се робот физички налази. Ова разлика се све више повећава док се не изврши локализација помоћу Калмановог филтера, на основи контролних тачака.

Распоред контролних тачака које дефинишу мапу на основу које се врши локализација мобилног робота у експерименталном технолошком окружењу, њихове димензије, као и њихов број у окружењу одређују могућност робота да повећа своју сигурност да се налази у позицији која је израчуната. Уколико је контролна тачка малих димензија, постоји велика вероватноћа да је сензорски систем робота неће детектовати. Насупрот томе, уколико је контролна тачка већих димензија њена функција на основу које робот врши корекцију положаја израчунатог у претходној итерацији главног управљачког алгоритма је значајно умањена.

Поред претходно дефинисаних параметара јако је важно успоставити тачну вредност растојања између погонских тачака, као и њихов пречник. У супротном разлика између израчунатог и физичког положаја мобилног робота коју остварује у току кретања биће значајно повећана. Ови параметри зависе од конкретне конфигурације мобилног робота, и за реализовану конфигурацију дати су у претходној тачки.

На основу кретања које је робот успешно реализовао у оквиру примера из пете тачке може се закључити и да вештачка неуронска мрежа која одређује команду за ротацију вратила мотора на основу потребне ротације мобилног робота у глобалном координатном систему своју функцију успешно. Такође приликом експерименталних тестирања робота није се јавио ни проблем при класификацији нивоа одбијене светлости од подлогу, тј. при одређивњу боје подлоге.

7. Закључак

На основу усвојеног и развијеног модела кретања, обучених и усвојених неуронских мрежа, Калмановог филтера као ефикасног система за естимацију, предвиђање, положаја робота и детаљно развијеног A* алгоритма за генерисање оптималне путање омогућено је интелигентно понашање мобилног агента – робота у задатом експерименталном технолошком окружењу.

За одређивање команде ротације вратила мотора у односу на потребну ротацију робота у глобалном координатном систему извршена је имплементација вештачке неуронске мреже. Такође, правци будућих размишљања могли би бити у правцу развоју вештачке неуронске мреже која би генерисала команду ротације вратила мотора за кретање робота унапред.

Управљачка стратегија робота се заснива на сигналимa прикупљеним употребом само једног сензора. Употреба већег броја сензора који се налазе у *Lego* комплету и који су описани у претходним тачкама пројекта, омогућила би прикупљање већег броја података. Истовремено би се могле добити информације о постојању препреке, удаљености препреке, контакту са ограничењем радног окружења, као и реакције робота на говорне команде човека. На тај начин јавила би се и потреба за анализом додатних алгоритама за обраду тих сигнала. Камера, као сензор, као и обрада слике генерисане камером пружила би мобилном роботу додатне информације за одређивање позиције и оријентације, али и ова унапређења захтевала би додатни хардвер, као и софтверска решења.

Сви параметри који се односе на производну линију технолошког окружења су тестирани и верификовани у *AnyLogic* софтверском пакету. Времена транспорта материјала и обраде припремка, капацитети бафера и конвејера, као и брзине кретања конвејера су неки од параметара коришћени у овом софтверу. Симулирањем и анализом добијених резултата симулације одређене параметре било је потребно оптимизовати како би се повећала продуктивност и производност а самим тим и перформансе интелигентног технолошког система.

8. Литература

- [1] Бабић, Б., Предавања на предмету Интелигентни технолошки системи (ПРО220-0131), Универзитет у Београду -Машински факултет, Београд, 2009.
- [2] Бабић, Б., Предавања на предмету Компјутерска симулација и вештачка интелигенција Универзитет у Београду -Машински факултет, Београд, 2007.
- [3] Бабић, Б., *Flexu – интелигентни експерт систем за пројектовање ФТС*, Серија монографских дела Интелигентни технолошки системи (Уредник серије: Проф. др Владимир Милачић), Књига 5, Универзитет у Београду -Машински факултет, Београд, 2003.
- [4] Вуковић, Н., Материјали са аудиторних вежби на предмету интелигентни технолошки системи (ПРО220-0131), Универзитет у Београду -Машински факултет, Београд, 2009.
- [5] Кикојевић - Бабић, Ј., Развој система за аутоматизовано пројектовање технолошких процеса за класу ротационих делова, магистарски рад, Универзитет у Београду –Машински факултет, Београд, 2002.
- [6] Лазић, Д., Предавања на предмету Аутоматско управљање, Универзитет у Београду - Машински факултет, Београд, 2008.
- [7] Милутиновић, Д., Предавања на предмету Индустијски роботи Универзитет у Београду -Машински факултет, Београд, 2008.
- [8] Миљковић, З., Предавања на предмету Интелигентни технолошки системи (ПРО220-0131), Универзитет у Београду -Машински факултет, Београд, 2009.
- [9] 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.
- [10] Национални институт Америке за стандард и технологије www.itl.nist.gov/div897/sqg/dads/HTML/manhatanDistance.html), последњи приступ фебруар 2010.
- [11] Петровић, П., Предавања на предмету Мехатронски системи, Универзитет у Београду -Машински факултет, Београд, 2009.