

Универзитет у Београду
Машински факултет

Бранко М. Милорадовић

Прилог развоју интелигентног понашања
мобилног робота у дефинисаном
технолошком окружењу коришћењем
метода одлучивања

ДИПЛОМСКИ (M.Sc.) РАД

Београд, 2010

Бранко Милорадовић¹

Резиме

У дипломском раду приказан је један од могућих начина замене конвенционалног начина за обављање унутрашњег транспорта у технолошким окружењима. Ново, побољшано, решење укључује коришћење интелигентног мобилног робота за транспорт материјала. Ово је остварено имплементирањем модела кретања и алгоритмом за планирање и одабир оптималне путање. Епитет „интелигентни” заслужен је коришћењем вештачких неуронских мрежа у домену препознавања постојања препрека и доношењу одлука о начину њиховог успешног избегавања. Експериментални резултати, изведени на Khepera II мобилином роботу, оправдали су очекивања и показали да овакво решење има практичну употребу у данашњим технолошким окружењима.

Кључне речи: интелигентни мобилни робот, вештачке неуронске мреже, методе одлучивања, граф видљивости, избегавање препрека.

¹ Бранко Милорадовић 1059/08, Универзитет у Београду – Машински факултет, студент друге године дипломских академских студија.
Е-пошта: dunker23@gmail.com

Садржај

| | |
|---|-----------|
| Садржај | 3 |
| Номенклатура..... | 5 |
| Списак слика..... | 7 |
| Списак табела | 9 |
| 1. Увод | 10 |
| 2. Поставка проблема | 12 |
| 2.1 Модел технолошког окружења | 12 |
| 3. Развој и имплементација модула за одређивање положаја мобилног робота у лабораторијском моделу технолошког окружења | 14 |
| 3.1 Модел кретања на основу пређеног пута | 15 |
| 4. Алгоритми за планирање путање, навигација, теорија графова, алгоритми за претрагу графова и одређивање оптималне путање | 19 |
| 4.1 Теорија графова..... | 24 |
| 4.2 Дајкстрин алгоритам | 25 |
| 5. Имплементација метода одлучивања, базираних на машинском учењу, у домену препознавања постојања препрека током кретања мобилног робота..... | 29 |
| 5.1 Инфрацрвени сензори | 29 |
| 5.2 Вештачке неуронске мреже..... | 34 |
| 5.2.1 Левенберг – Маркеов алгоритам учења | 35 |
| 5.2.2 Активационе функције и модели неуронских мрежа..... | 35 |
| 5.2.3 Перцептрон..... | 37 |
| 5.2.4 Backpropagation (BP) неуронска мрежа | 39 |
| 5.2.6 МРФ Мреже..... | 40 |
| 5.2.7 Топологија МРФ мрежа | 40 |
| 5.2.8 Обучавање МРФ мреже..... | 42 |
| 5.2.9 Поређење МРФ мреже са вишеслојним перцептроном | 43 |
| 5.3 Примена вештачких неуронских мрежа у домену препознавања постојања препрека..... | 44 |
| 5.4 Ток информација..... | 49 |
| 6. Функционални захтеви и параметри пројектовања у домену пројектовања интелигентног понашања мобилног робота | 52 |
| 7. Дрво и матрица одлучивања | 56 |
| 7.1 Одлучивање са ризиком – матрица одлучивања..... | 57 |
| 7.2 Дрво одлучивања..... | 57 |
| 8. Тестирање и верификација тачности и перформанси развијених решења интелигентног понашања мобилног робота у радном простору дефинисаном у оквиру лабораторијског модела технолошког окружења ... | 59 |
| 8.1 Резултати обучавања МРФ | 59 |
| 8.2 Резултати обучавања мрежа за препознавање препрека..... | 62 |
| 8.3 Резултати експеримента праћења путање са избегавањем препрека | 68 |
| 9. Закључак | 70 |

| | | |
|------------|--|-----------|
| 10. | Литература..... | 71 |
| 11. | Прилог | 73 |
| 11.1 | Главни фајл..... | 73 |
| 11.2 | Код са функцијама за кретање робота | 74 |
| 11.3 | Код за скретање робота..... | 75 |
| 11.4 | Код за праволинијско кретање | 76 |
| 11.5 | Код за заобилажење препрека и праћење зида | 77 |

Номенклатура

x_t - вектор стања система
 x - позиција мобилног робота у координатном систему по оси X
 y - позиција мобилног робота у координатном систему по оси Y
 z - позиција мобилног робота у координатном систему по оси Z
 θ - угао под којим се робот налази у односу на Z осу
 Ψ - угао под којим се робот налази у односу на X осу
 φ - угао под којим се робот налази у односу на Y осу
 Δx - прираштај по x оси
 Δy - прираштај по y оси
 $\Delta\theta$ - прираштај угла
 Δs_d - пређени пут десног точка
 Δs_l - пређени пут левог точка
 b - растојање између точка
 G - уређени скуп (V, E)
 V - скуп чворова грфа G
 E - скуп грана графа G
 $\Gamma_{(x)}$ - скуп свих суседа чвора x
 A - подскуп скупа V
 ω - тежина гране
 s - почетни чвор у G скупу
 S - еуклидска дистанца између 2 чвора
 M_{x_1} - координата x првог чвора
 M_{x_2} - координата x другог чвора
 M_{y_1} - координата y првог чвора
 M_{y_2} - координата y другог чвора
 $IC_{\text{средње}}$ - средња вредност мерења
 IC_n - n -то мерење
 n - укупан број обављених мерења
 Dev - средња девијација током мерног периода
 A_i - алгоритам учења
 w_{ij} - вектор тежинских односа
 x_i - улазни вектор
 H - Хесијан матрица
 I - јединична матрица
 λ - константан параметар
 o^k - циљни вектор
 X - улазни вектор
 L - број неурона у скривеном слоју
 μ_j - средња вредност j -те Гаусове функције
 d - димензија улазног вектора
 λ_{jk} - излазни тежински односи
 M - број неурона у излазном слоју
ПП - параметри пројектовања
ФЗ - функционални захтеви
МА - матрица пројектовања
 p_i - вероватноћа понашања
Std_mse_test - стандардна девијација за тест скуп
Std_rmse_train - стандардна девијација за обучавајући скуп

m_rmse_train - средња вредност корена средње квадратне грешке за тест скуп
m_rmse_test - средња вредност корена средње квадратне грешке за обучавајући скуп
min_rmse_train - минимална вредност корена средње квадратне грешке за тест скуп
min_rmse_test - минимална вредност корена средње квадратне грешке за обучавајући скуп
m_iter_no - просечан број итерација за обучавање мреже
m_time - просечно време трајања обучавања мреже, изражено у секундама

Списак слика

| | |
|---|----|
| Слика 1.1: Kherera II мобилни робот | 10 |
| Слика 2.1: Модел технолошког окружења | 12 |
| Слика 2.2: Layout производно монтажног погона предузећа „МОНТПРОЈЕКТ” | 13 |
| Слика 3.1: Приказ интелигентног мобилног робота у два узастопна тренутка t и t' | 14 |
| Слика 3.2: Изглед инкременталног енкодера са свим његовим елементима | 15 |
| Слика 3.3: Однос ON и OFF периода у току рада мотора | 16 |
| Слика 4.1: Блок дијаграм процеса планирања путање | 20 |
| Слика 4.2: Приказ модела технолошког окружења са уцртаном стартом и циљном позицијом робота | 21 |
| Слика 4.3: Блок дијаграм формирања графа видљивости..... | 22 |
| Слика 4.4: Приказ модела технолошког окружења након извршавања графа видљивости | 23 |
| Слика 4.5: Пример једног усмереног графа..... | 24 |
| Слика 4.6: Пример једног неусмереног графа..... | 24 |
| Слика 4.7: Пример графа који се назива дрво | 25 |
| Слика 4.8: Пример претраге једног неусмереног графа од чвора О до чвора Т | 27 |
| Слика 4.9: Приказ модела технолошког окружења након завршавања процеса планирања кретања и добијања оптималне путање..... | 28 |
| Слика 5.1: а) Приказ упутством декларисаног распореда ИЦ сензора на Kherera мобилном роботу. б) Стварни распоред ИЦ сензора | 29 |
| Слика 5.2: Изглед ИЦ сензора TCRT1000 | 30 |
| Слика 5.3: Начин рада ИЦ сензора..... | 30 |
| Слика 5.4: Вредности са сензора добијене за време читавања амбијенталног осветљења (приказани су само предњи сензори) | 31 |
| Слика 5.5: Добијена читавања са сензора 1 и 2 за растојања од 0,5-5cm | 32 |
| Слика 5.6: Добијена читавања са сензора 3 и 4 за растојања од 0,5-5cm..... | 32 |
| Слика 5.7: Добијена читавања са сензора 5 и 6 за растојања од 0,5-5cm | 33 |
| Слика 5.8: Линеарна активациона функција..... | 36 |
| Слика 5.9: matlab „logsig“ активациона функција..... | 36 |
| Слика 5.10: matlab „tansig“ активациона функција | 37 |
| Слика 5.11: Елементарна структура перцептрона | 38 |
| Слика 5.12: Backpropagation (BP) неуронска мрежа | 39 |
| Слика 5.13: Приказ МРФ мреже са три слоја (улазни, скривени и излазни слој) | 41 |
| Слика 5.14: Приказ утицаја растојања неурона на његову активацију | 41 |
| Слика 5.15: Приказ начина класификације података код ВСП и МРФ мрежа. ВСП то ради помоћу хиперравни којима раздваја регионе, док се МРФ служи кластеровањем..... | 44 |
| Слика 5.16: Приказ вредности добијених са сензора и вредности добијених од ВНМ за сензоре 1 и 2 | 45 |
| Слика 5.17: Приказ вредности добијених са сензора и вредности добијених од ВНМ за сензоре 3 и 4..... | 45 |
| Слика 5.18: Приказ вредности добијених са сензора и вредности добијених од ВНМ за сензоре 5 и 6..... | 46 |
| Слика 5.19: Скице карактеристичних препрека као и одговарајућа читавања сензора | 47 |
| Слика 5.20: На слици са леве стране робот наилази на препреку, а десна слика приказује успешно препознавање и избегавање | 48 |
| Слика 5.21: Приказ локалног минимума који може да се деси у току кретања робота..... | 48 |
| Слика 5.22: Ток информација унутар система интелигентног мобилног робота..... | 49 |
| Слика 7.1: Дрво одлучивања за информације добијене са сензора | 58 |

| | |
|--|----|
| Слика 8.1: Графички приказ резултата обучавања за обучавајући скуп као и за тест скуп. Сlike 1),2),3),4) одговарају сензорима 1,2,3,4, респективно. X оса представља очитане вредности са сензора, а Y растојање у mm | 60 |
| Слика 8.2: Графички приказ резултата обучавања за обучавајући скуп као и за тест скуп. Сlike 5),6),7),8) одговарају сензорима 5,6,7,8, респективно. X оса представља очитане вредности са сензора, а Y растојање у mm | 61 |
| Слика 8.3: Приказ излаза неуронске мреже након завршеног процеса обучавања..... | 64 |
| Слика 8.4: Приказ једног неуспешног обучавања ВНМ..... | 65 |
| Слика 8.5: Приказ излаза неуронске мреже након завршеног процеса обучавања..... | 67 |
| Слика 8.6: Са леве стране приказ генерисане путање, са десне стране остварена путања у току кретања (експеримент 1) | 68 |
| Слика 8.7: Са леве стране приказ генерисане путање, са десне стране остварена путања у току кретања (експеримент 1) | 69 |

Списак табела

| | |
|---|----|
| Табела 2.1: Списак препрека у моделу технолошког окружења | 12 |
| Табела 4.1: Приказ псеудо кода Дајкстриног алгоритма..... | 26 |
| Табела 4.2: Репрезентација графа помоћу матрице суседства | 27 |
| Табела 5.1: Средња вредност и стандардна девијација корена средње квадратне грешке | 42 |
| Табела 6.1: Приказ функционалних захтева (ФЗ) при пројектовању интелигентног понашања мобилног робота за транспорт материјала..... | 53 |
| Табела 6.2: Приказ параметара пројектовања (ПП) при пројектовању интелигентног понашања мобилног робота за транспорт материјала..... | 54 |
| Табела 6.3: Основна једначина пројектовања за цео систем интелигентног мобилног робота (Матрица пројектовања) | 55 |
| Табела 7.1: Матрица одлучивања | 57 |
| Табела 8.1: Приказ параметара добијених при обучавању МРФ мреже за препознавање растојања помоћу ИЦ сензора (за прва 4 сензора) | 59 |
| Табела 8.2: Приказ параметара добијених при обучавању МРФ мреже за препознавање растојања помоћу ИЦ сензора (за друга 4 сензора) | 60 |
| Табела 8.3: Приказ средње вредности средње квадратне грешке за излаз из ВНМ где се препрека налази на 40 [mm] | 61 |
| Табела 8.4: Приказ средње вредности средње квадратне грешке за излаз из ВНМ где се препрека налази на 20 [mm] | 61 |
| Табела 8.5: Приказ средње вредности средње квадратне грешке за излаз из ВНМ где се препрека налази на 12 [mm]..... | 61 |
| Табела 8.6: m_gmse процентуално приказан за свих осам сензора и сва три растојања која су коришћена у овом експерименту | 62 |
| Табела 8.7: Приказ резултата тренирања вештачке неуронске мреже са једним скирвеним слојем (растојање до препрека 20 [mm])..... | 62 |
| Табела 8.8: Приказ резултата тренирања вештачке неуронске мреже са два скирвена слоја (растојање до препрека 20 [mm]) | 63 |
| Табела 8.9: Приказ резултата тренирања вештачке неуронске мреже са три скирвена слоја (растојање до препрека 20 [mm]) | 63 |
| Табела 8.10: Варирање параметра учења за мрежу са најбољим резултатима (растојање до препрека 20 [mm])..... | 64 |
| Табела 8.11: Приказ резултата тренирања вештачке неуронске мреже са једним скирвеним слојем (растојање до препрека 40 [mm])..... | 65 |
| Табела 8.12: Приказ резултата тренирања вештачке неуронске мреже са два скирвена слоја (растојање до препрека 40 [mm]) | 66 |
| Табела 8.13: Приказ резултата тренирања вештачке неуронске мреже са три скирвена слоја (растојање до препрека 40 [mm]) | 66 |
| Табела 8.14: Варирање параметра учења за мрежу са најбољим резултатима (растојање до препрека 40 [mm])..... | 67 |

1. Увод

У овом дипломском раду приказано је решење инжењерских проблема у домену пројектовања интелигентног понашања агента - мобилног робота, ради моделирања унутрашњег транспорта сировина, материјала и готових делова у производном окружењу. Пројектно решење је имплементирано на *Khepera II* мобилном роботу (слика 1.1). За потребе кретања мобилног робота потребно је развити и извршити имплементацију софтверског решења, базираног на методама одлучивања, које ће са унапред задатом тачношћу извршити одговарајуће пресликавање, одређено избором који робот треба да направи.

С обзиром на комплексност постављеног проблема који подразумева избегавања препрека коришћењем само информација са инфрацрвених (ИЦ) сензора, развијено решење је базирано на машинском учењу, односно применом система вештачких неуронских мрежа.



Слика 1.1: Khepera II мобилни робот

Решење овог проблема описано је кроз десет поглавља у овом раду:

Поглавље 2 - Поставка проблема где је описан проблем који треба решити и чија су решења представљена у овом дипломском раду. Такође, овде је приказан модел и *layout* технолошког окружења.

Поглавље 3 - С обзиром да је у питању мобилни робот, неопходан је развој и имплементација модула за одређивање положаја мобилног робота у лабораторијском моделу технолошког окружења. Одометријска локализација.

Поглавље 4 – Овде су представљени алгоритми за планирање путање и навигацију, осврт на основне појмове о графу, претрага графа и алгоритми за проналажење оптималне путање. Граф видљивости. Дајкстра алгоритам.

Поглавље 5 – Информације о коришћеним сензорима као и вештачким неуронским мрежама. Њихова топологија, активационе функције и алгоритми обучавања. Примена вештачких неуронских мрежа у домену препознавања постојања препрека. Класификација препрека. Ток информација у систему интелигентног мобилног робота.

Поглавље 6 – Одређивање скупа параметара који задовољавају функционалне захтеве, сходно усвојеном критеријуму одлучивања у домену пројектовања интелигентног понашања мобилног робота.

Поглавље 7 – Формирање дрвета и матрице одлучивања за задати проблем експлоатације мобилног робота у дефинисаном експерименталном окружењу.

Поглавље 8 – У циљу утврђивања оправданости примењеног концепта машинског учења, извршено је тестирање и верификација тачности развијених решења интелигентног понашања мобилног робота у радном простору дефинисаном у оквиру лабораторијског модела технолошког окружења.

Поглавље 9 – Добијени закључак и анализа пројектног решења.

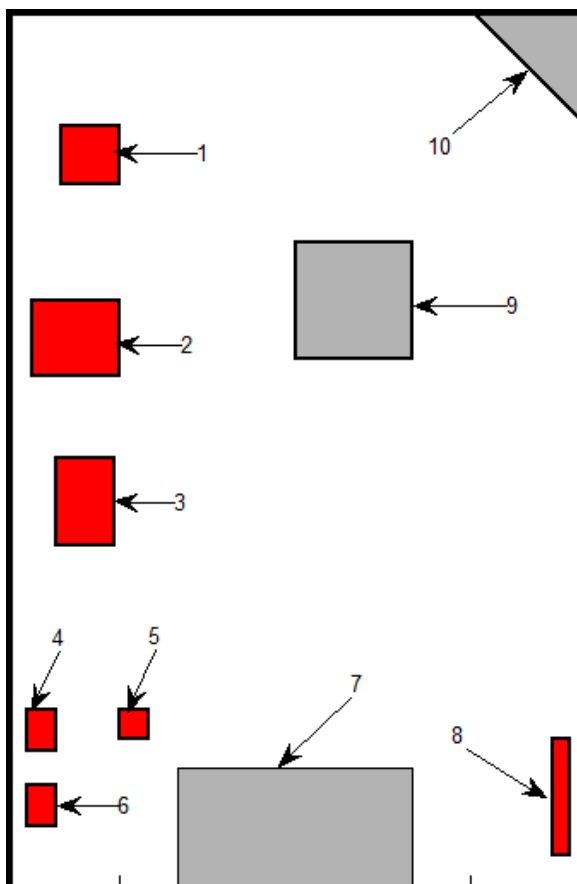
Поглавље 10 – Прилог. Неки од *matlab* кодова развијених и имплементираних на интелигентном мобилном роботу.

2. Поставка проблема

2.1 Модел технолошког окружења

Предузеће „МОНТПРОЈЕКТ” се бави производњом електро-ормана, где се транспорт унутар радног постројења врши помоћу виљушкара. Циљ овог пројекта је побољшање унутрашњег транспорта, заменом садашњег вида транспорта мобилним роботом. Робот би извршавао функцију преношења додатног материјала, обрадака, алата, и сл. између машина, складишта, транспортних трака и магацина за одлагање делова унутар постројења.

Производни капацитети предузећа „МОНТПРОЈЕКТ” готово су у потпуној сагласности са препорученим обрадним системима – машинама алаткама које налаже *Siemens AG*. Предузеће „МОНТПРОЈЕКТ” располаже следећим машинама:



Табела 2.1: Списак препрека у моделу технолошког окружења

| | |
|----|--|
| 1 | Маказе за сечење |
| 2 | НУ машина за пробијање и просецање |
| 3 | НУ хидраулична апкант преса |
| 4 | Стубна бушилица |
| 5 | Линија за обраду делова од бакра |
| 6 | Кружна тестера |
| 7 | Гардероба, Магацин, Архива, Котларница |
| 8 | Међускладиште |
| 9 | Монтажни сто |
| 10 | Складиште готових производа |

Слика 2.1: Модел технолошког окружења

Поред ових машина алатки „МОНТПРОЈЕКТ” располаже и оштрилицом алата, што је пожељна машина у оваквом производном систему, јер је због серијске производње, која је у сталном квантитативном порасту, изражено хабање расположивих алата за обраду лима. За опслуживање машина 1, 2 и 4 потребна су два извршиоца, за машину 3 један извршилац, за пројектовање технологије и програмирање НУ машина 2 и 3 такође један извршилац, као и за машине 4, 6 и линију за бакар 5.

3. Развој и имплементација модула за одређивање положаја мобилног робота у лабораторијском моделу технолошког окружења

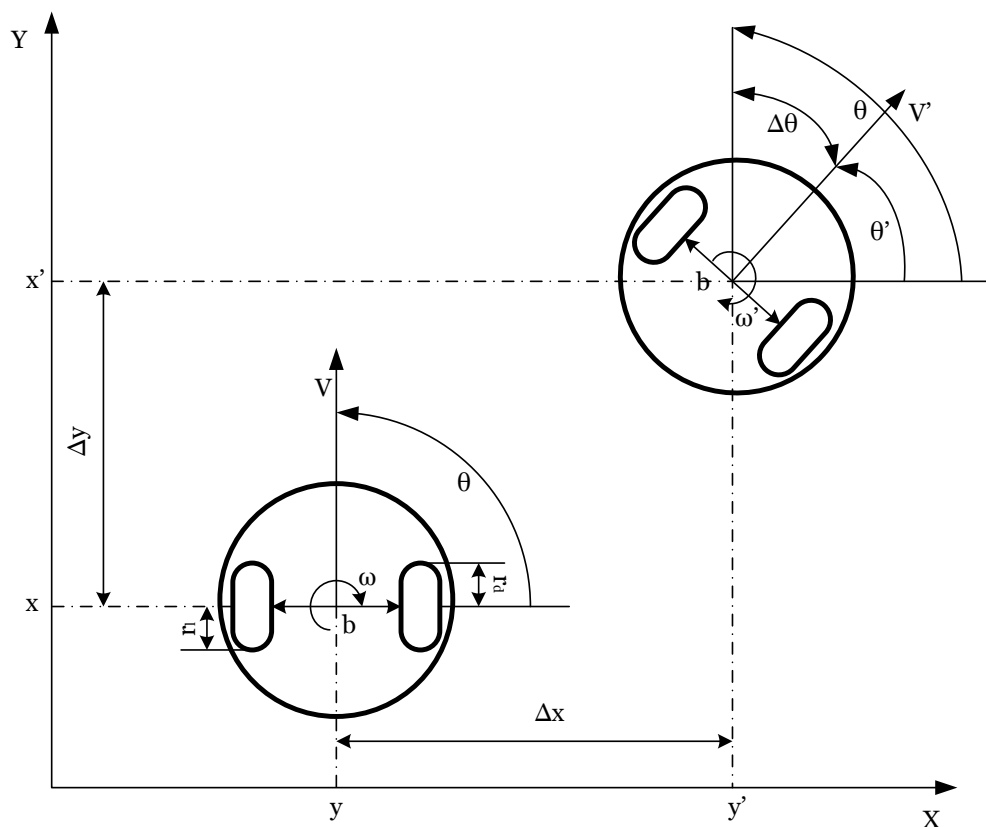
Фундаментални проблем у мобилној роботизици јесте имати право знање о положају робота у сваком временском тренутку. За извршавање одређене функције, која захтева померање робота, неопходно је решити проблем његовог кретања у простору. Одређивање положаја мобилног робота у радном окружењу састоји се из проналажења параметара који дефинишу његову позицију и оријентацију. Координате тачака у којима се мобилни робот налази у односу на раније дефинисани апсолутни координатни систем, представљају позицију, док угао, представља оријентацију робота. Број позиција и оријентација које робот може да оствари, директно зависи од броја степени слободе. У општем случају мобилни робот може имати 6 степени слободе где је вектор стања описан на следећи начин :

$$x_t = (x, y, z, \theta, \Psi, \varphi) \quad (3.1)$$

где су:

x, y, z – позиција мобилног робота у координатном систему по оси X, Y, Z респективно;

θ, Ψ, φ – угао под којим се робот налази у односу на Z, X, Y осу респективно;



Слика 3.1: Приказ интелигентног мобилног робота у два узастопна тренутка t и t'

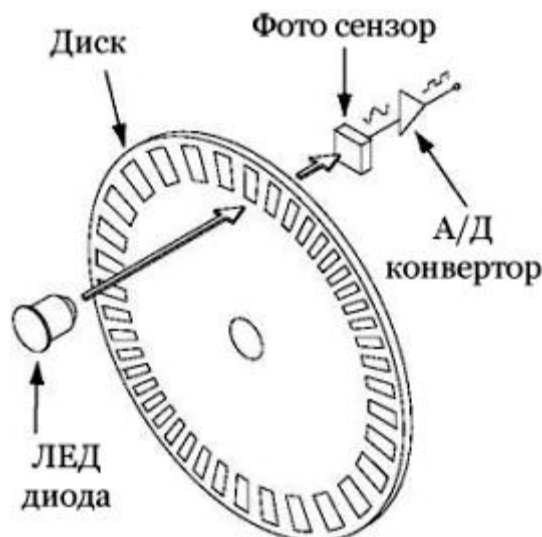
У овом случају у питању је мобилни робот који врши раванско кретање, што значи да се кретање обавља у X и Y равни, док се угао θ односи на заокретање робота око z осе. Узимајући ово у обзир, наша једначина се своди на следећи облик :

$$x_t = (x, y, \theta) \quad (3.2)$$

Модел кретања, који је коришћен у овом експерименту, рачуна положај робота на основу пређеног пута. Овај модел кретања познатији је под називом одометријска локализација (одометрија).

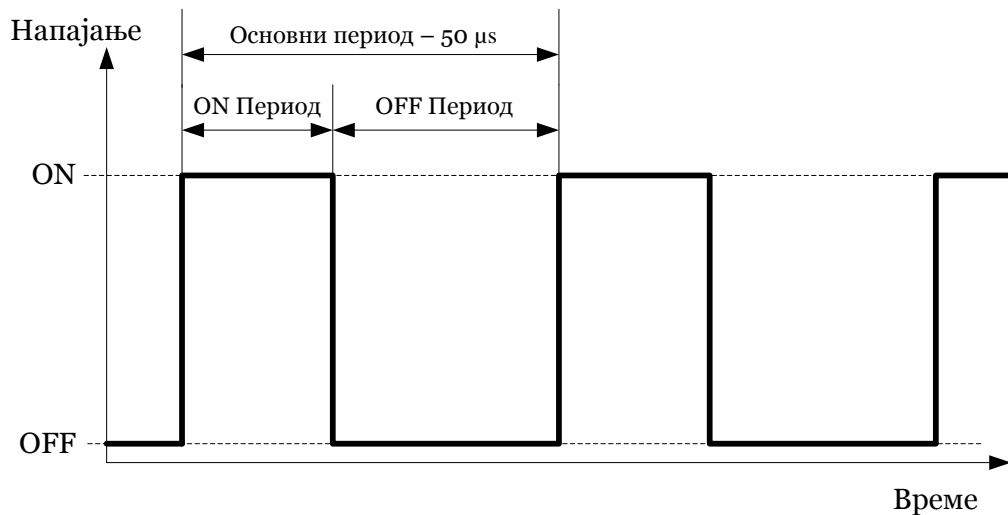
3.1 Модел кретања на основу пређеног пута

Модел кретања помоћу пређеног пута представља управљање кретањем мобилног робота на основу информације о пређеном путу левог и десног точка. Тај податак се добија тако што се на основу броја обртаја и пречника точка, за одређени временски интервал, израчунава пређени пут. У току кретања долази до појаве различитих губитака, па је неопходно добијати повратну информацију шта је мотор заиста остварио од оног што је било предвиђено. Те информације се добијају са енкодера који су уграђени на вратилу мотора. Конкретно, они служе за читавање угла ротације точкава. Мотори добијају информацију од рачунара у облику импулса, на основу којих се ротира вратило погонског мотора за одређени угао. Сваки точкав покреће DC мотор преко 25:1 редуктора. Инкрементални енкодер, који се налази на вратилу мотора, даје 24 импулса за једну пуну ротацију. То нам омогућује резолуцију од 600 импулса за један точкав, што одговара односу од 12 импулса за 1 [mm] пређеног пута.



Слика 3.2: Изглед инкременталног енкодера са свим његовим елементима

Главни процесор на *Khepera* мобилном роботу, има директну контролу над напајањем мотора и може да чита импулсе са инкременталног енкодера. Интерапт (енгл. interrupt) рутина детектује сваки импулс инкременталног енкодера и ажурира вредности позиције точка.



Слика 3.3: Однос ON и OFF периода у току рада мотора

Одометрија се темељи на чињеници да ротација тачкова може бити преведена у линијске помераје релативне у односу на равну површину по којој се мобилни робот креће. Међутим, у реалним условима, ово није могуће остварити због проклизавања тачкова, ограничености енкодера и многих других ствари. Грешке које се јављају у одометрији могу бити системске и несистемске.

Системске :

- неједнаки полупречници тачкова
- неподешеност тачкова
- коначна резолуција енкодера

Несистемске:

- прелазак преко неравног терена
- прелазак преко неочекиваних објеката на подлози
- клизање тачкова због:
 - клизаве подлоге
 - превеликог убрзања
 - брзог скретања (заношења)
 - интеракције са другим објектима

Системске грешке су посебно изражене, јер се константно акумулирају. На равним површинама оне су доминантне и више доприносе порасту укупне грешке, док су несистемске доминантније на неравним површинама. Проблем са несистемским грешкама се огледа у томе што се појављују изненадно и тешко их је предвидети.

Математички модел кретања на основу одометрије се представља на следећи начин. Прво је потребно за ограничени простор усвојити одређени координатни систем. Потом је неопходно дефинисати позицију и оријентацију мобилног робота у њему, што је представљено матрицом $x' = [x \ y \ \theta]^T$, где су променљиве претходно објашњене. Сада се одређују подаци о прираштајима величина које дефинишу кретање:

$$\Delta x = \Delta s \cdot \cos\left(\theta + \frac{\Delta\theta}{2}\right) \quad (3.3)$$

$$\Delta y = \Delta s \cdot \sin\left(\theta + \frac{\Delta\theta}{2}\right) \quad (3.4)$$

$$\Delta\theta = \frac{\Delta s_d - \Delta s_l}{b} \quad (3.5)$$

$$\Delta s = \frac{\Delta s_d + \Delta s_l}{2} \quad (3.6)$$

Пошто су познати параметри тренутне позиције мобилног робота, као и прираштај, збиром ових параметра се добија наредна позиција у којој ће се наћи мобилни робот.

$$x' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cdot \cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \Delta s \cdot \sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \Delta\theta \end{bmatrix} \quad (3.7)$$

У развијеном облику ова матрица изгледа овако:

$$x' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_d + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{b}\right) \\ \frac{\Delta s_d + \Delta s_l}{2} \cdot \sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{b}\right) \\ \frac{\Delta s_d - \Delta s_l}{b} \end{bmatrix} \quad (3.8)$$

Где су:

Δx – прираштај по x оси;

Δy – прираштај по y оси;

$\Delta\theta$ – прираштај угла;

Δs_d – пређени пут десног точка;

Δs_l – пређени пут левог точка;

b – растојање између точкава;

Познавајући резолуцију енкодера и полупречник точка можемо у сваком тренутку одредити пређени пут $(\Delta s_d, \Delta s_l)$. Уврштавањем ових вредности у једначину (3.8), добијамо вредности координата x, y и угла θ .

4. Алгоритми за планирање путање, навигација, теорија графова, алгоритми за претрагу графова и одређивање оптималне путање

Когнитивност генерално представља сврсисходно одлучивање и извршавање циља које систем покушава да достигне на највишем нивоу. У случају мобилног робота, посебан аспект когнитивности, директно везан за мобилност робота је навигациона компетентност. Поседујући делимична знања о својој околини, о позицији циља или серији позиција, навигација обухвата способност мобилног робота (агента) да реагује у складу са претходним знањима и сензорским информацијама како би достигао жељену позицију што је ефикасније и поузданије могуће.

До сада је, у заједници мобилне роботике, предложено много приступа решавању овог проблема. Сви приступи, иако не делује тако на први поглед, су у суштини слични. Главна разлика се огледа у начину декомпозиције проблема у мање подгрупе. Постоје још два кључна услова за успешно решавање проблема навигације мобилног робота. Познајући мапу и позицију циља, *планирање путање* представља идентификацију трајекторије која ће омогућити мобилном роботу да достигне жељену позицију, односно, способност робота да генерише план сопственог кретања с циљем извршавања одређеног задатка.

Код обављања одређеног задатка захтева се кретање робота од старта до циља без колизије са објектима и препрекама које се могу наћи на његовој путањи. Узимајући у обзир читавања са сензора, у реалном времену, *избегавање препрека* подразумева измену трајекторије у циљу ескивирања судара са објектима у околини. Да би се ово успешно остварило робот мора реаговати на основу прочитаних сензорских информација. Међутим, планирање и реаговање делују као супротни приступи. У ствари, када су у питању физички системи, као што је интелигентни мобилни робот, планирање и реаговање имају јаку комплементарност [1].

Постоје две основне парадигме за планирање путања са избегавањем препрека:

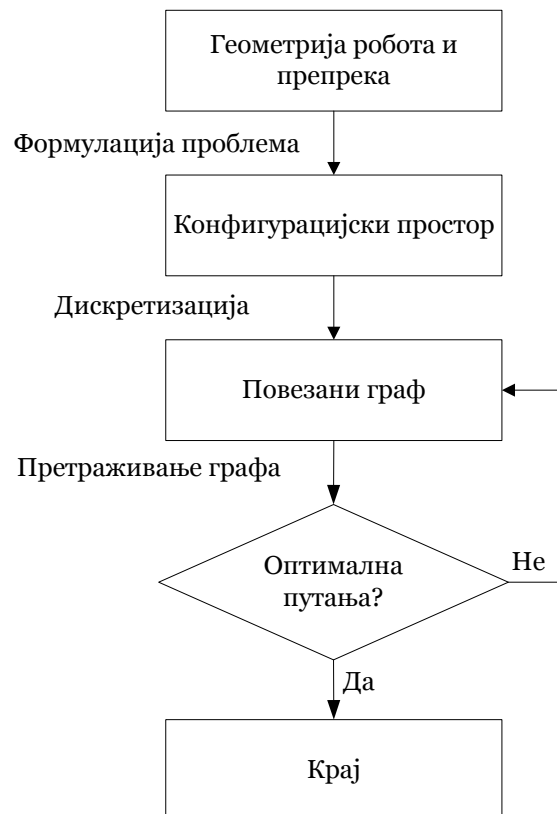
- 1) Глобални приступ који генерише путању у *off-line* режиму коришћењем познатих информација о средини.
 - Захтева прецизно дефинисан модел окружења што са собом повлачи и веће време рачунања.
 - Овим приступом није могуће реаговати на непредвиђена дешавања у радном окружењу.
- 2) Локални приступ се обавља у *on-line* режиму коришћењем малог броја познатих информација о простору, али је зато неопходно стално добијање нових података путем сензора.
 - Могуће је реаговати на непредвиђене промене и не захтева пуно времена за израчунавање.
 - На овај начин није могуће добити оптималну путању.

Навигација код робота представља извршавање одређеног тока акција, планирања, како би достигао циљ. За време тог извршавања, робот мора да реагује на

непредвиђене догађаје, на такав начин, да ипак достигне свој циљ. Долазимо до закључка да без реаговања, робот не може никад физички да досегне до свог циља, док, без планирања, само реаговање не може да контролише свеукупно понашање робота и да га доведе до жељеног циља. Да би циљ био испуњен, мобилни робот мора узети у обзир нове информације, добијене са сензора, за време извршавања плана кретања. Како време пролази, околина се мења и добијају се нова очитавања са сензора. У том тренутку реаговање постаје кључна ствар. У најбољем случају, робот ће одреаговати тако да измени своје понашање само на локалном нивоу, у циљу корекције унапред дефинисане трајекторије, тако да робот ипак испуни свој задатак и дође до своје циљне позиције.

Још пре појаве интелигентних мобилних робота, планирање путање је истраживано ради примене у области индустријских робота. Планирање путање за индустријског робота са пет степени слободе је далеко компликованије него планирање путање мобилног робота са диференцијалним погоном који се креће у раванском окружењу. Дакле и сами алгоритми за планирање путање су далеко једноставнији. Шта више, индустријски роботи морају да раде максималном брзином због економске исплативости. Зато се ту мора обратити пажња како на кинематику тако и на динамику, што додатно компликује планирање и извршавање, што није случај код мобилних робота. Они углавном раде на малим брзинама и динамика се ретко узима у обзир, чиме додатно упрошћавамо проблем.

Први корак у упрошћавању проблема јесте пресликавање робота, који има комплексан геометријски облик, у тачку, у новом апстрактном простору који се назива конфигурацијски простор [2]. Овим добијамо трансформацију оригиналног проблема у проблем планирања путање покретне тачке. Након тога се ради дискретизација континуираног конфигурацијског простора и конструише повезани граф. На крају се претражује граф како би се нашла оптимална путња којом би робот досегао циљ.

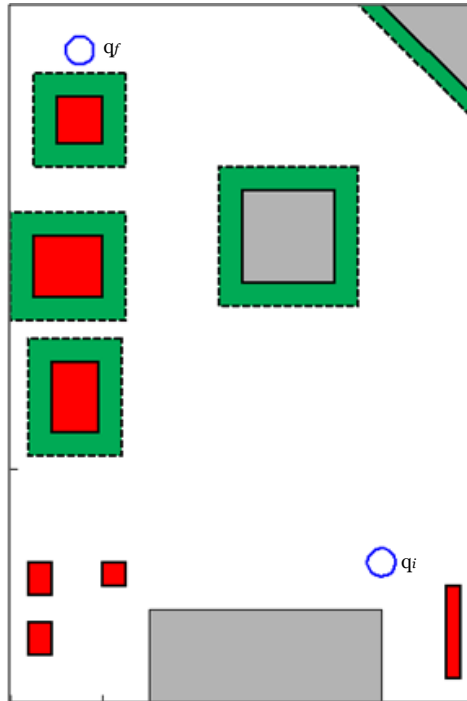


Слика 4.1: Блок дијаграм процеса планирања путање

На основу наведеног може се рећи да су улази у процес планирања:

- Геометријски опис робота и окружења
- Почетни и крајњи положај робота (позиција и оријентација)

Излаз из процеса планирања путање је трајекторија од почетног положаја q_i до циљног положаја q_f .



Слика 4.2: Приказ модела технолошког окружења са уцртаном стартном и циљном позицијом робота

Неки од захтева при процесу планирања путање могу бити :

- Минимални пређени пут
- Средина са покретним препрекама
- Средина са покретним циљем
- Минимални износ времена за прелажење путање
- Минимални утрошак енергије

или избор трајекторија које задовољавају нека друга ограничења.

Планирање путање захтева разматрање својства робота и структуре околине. Околина, односно простор у коме се креће мобилни робот, састоји се од препрека и слободног простора. Препреке представљају заузети део простора, односно део простора кроз који се мобилни робот не може кретати. Слободни простор је незаузети део простора кроз који се мобилни робот може слободно кретати. За одређивање куда се мобилни робот може кретати служи нам претходно дефинисани конфигурацијски простор. Скуп параметара који једнозначно одређују положај робота назива се конфигурација робота. Као што је претходно речено, конфигурација робота у овом случају је одређена његовом позицијом (x, y) и оријентацијом θ . Ако се претпостави да се конфигурација робота састоји од n параметара, тада се она може посматрати као тачка у n -димензионалном простору S , названом конфигурацијски простор. Конфигурацијски простор је скуп свих могућих положаја робота [3].

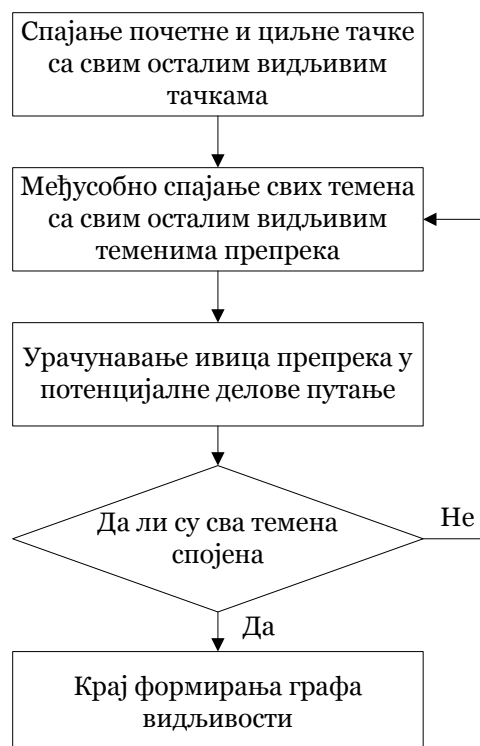
Први корак у планирању путање јесте трансформација реалног окружења у дискретизован модел, прикладан за одређени алгоритам планирања путање. Постоје три општа приступа за декомпозицију:

- Мапа путева (*road map*) – идентификује скуп трајекторија у слободном простору.
- Декомпозиција на ћелије – разликују се заузете и слободне ћелије.
- Потенцијално поље – Увођење привлачне и одбојне силе
 - Привлачна сила коју има циљ и која наводи мобилног робота.
 - Одбојна сила коју имају препреке чиме се спречава колизија мобилног робота са објектима у окружењу.

Road map алгоритми представљају групу алгорита који служе за планирање путање, а сви су базирани на формирању мреже могућих трајекторија од стартне до циљне позиције и обухвата следеће алгоритме :

- Граф видљивости
- Ворнојев дијаграм

У овом раду се користи *road map* алгоритам, тачније један од могућих приступа, који се назива граф видљивости. Он се најчешће примењује у дводимензионалном конфигурацијском простору са полигоналним препрекама. Добија се спајањем свих видљивих темена препрека са почетном и циљном конфигурацијом робота. Две тачке, односно темена, су међусобно видљиве ако између њих нема препрека, па се линија која спаја те две тачке може узети као потенцијални део трајекторије. Такође, робот се може кретати око ивице препреке ако су два суседна темена видљива. Тако конструисан граф путања унутар мапе простора омогућава да свака линија буде потенцијални део путање мобилног робота од стартне до циљне позиције.

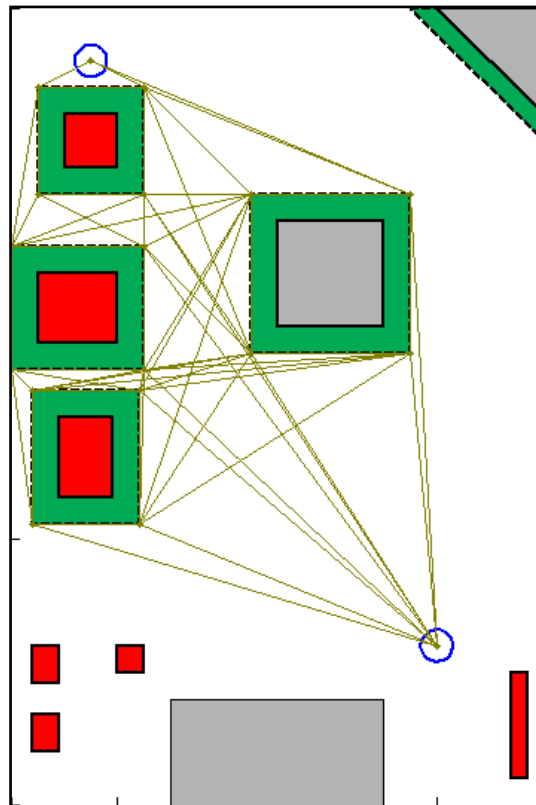


Слика 4.3: Блок дијаграм формирања графа видљивости

Основни алгоритам за израду графа видљивости изгледа овако:

1. Линијама се споје почетна и циљна тачка са свим осталим видљивим теменима препрека.
2. Међусобно се споје линијама сва темена са свим осталим видљивим теменима препрека.
3. Линије уздуж ивица препрека се такође рачунају као спојене линије.
4. Понављају се кораци 2 и 3 док се сва темена не споје.

Када применимо горе описани алгоритам, тј. граф видљивости на модел нашег технолошког окружења добијамо мрежу потенцијалних трајекторија кретања мобилног робота, што је приказано на слици 4.4 за произвољне почетне и циљне координате.



Слика 4.4: Приказ модела технолошког окружења након извршавања графа видљивости

Након добијања мреже физички могућих трајекторија по којима робот може да се креће, неопходно је одредити, не само ону путању која води од старта до циља, већ та путања мора бити и оптимална путања. Пре него што може да се крене са претрагом, мора се решити један од главних проблема графа видљивости. Наиме, оптимална путања добијена применом графа видљивости тежи да одведе робота што је могуће ближе препреци на путу до циља. Како је најкраће растојање између две тачке права линија, тако је и путања добијена графом видљивости оптимална путња у погледу растојања од старта до циља. Међутим, оваквим избором путање жртвујемо сигурност, јер робот прилази препреци што је ближе могуће. Постоје два начина за решавање овог проблема:

- Да се увећају препреке за величину полупречника робота или више, што се своди на примену суме Минковског (Minkowski sum)[4].
- Други начин за решење овог проблема јесте да се након добијања путање, изврши њена корекција тако да робот остане на сигурном растојању у односу на препреке у току кретања.

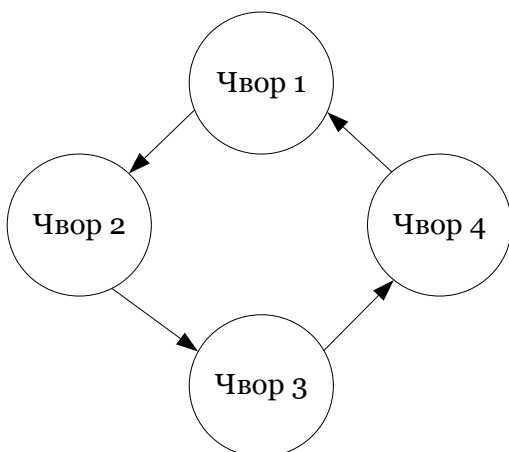
У овом раду решење овог проблема изведено је помоћу првог наведеног начина, с тим што је додатно упрошћен проблем, јер су само препреке које су релевантне за кретање мобилног робота подлегле промени величине. На слици 4.4 зелена боја представља проширење препрека, док су црвеном и сивом представљене стварне димензије препрека.

Постоји још један проблем код примене графа видљивости. С повећањем броја препрека повећава се и број страница и темена, а самим тим и број могућих трајекторија. Дакле, граф видљивости је изузетно брз и ефикасан у окружењу где нема много препрека, док је у окружењима са мноштвом препрека комплексних облика јако спор у поређењу са другим методама. Модел технолошког окружења у овом раду је упрошћен, број препрека је мали и углавном су правоугаоног облика што омогућује веома лаку и ефикасну примену графа видљивости.

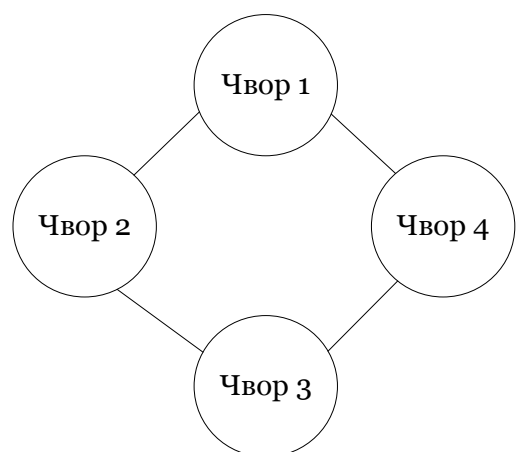
4.1 Теорија графова

Граф G представља уређени скуп (V, E) , где је V скуп чворова, а E скуп ивица, у коме свака ивица представља пар (x, y) чворова из V .

Кажемо да су два чвора *суседна* ако и само ако постоји ивица која их спаја, као и да је чвор v суседан ивици e ако и само ако је v један од чворова спојених ивицом e . Број ивица које су суседне са неким чвором је *степен* тог чвора. Сви чворови који су суседни неком чвору x се називају *суседима* x . Скуп свих суседа неког чвора x обележава се са $\Gamma_{(x)}$. За неки подскуп $A \subseteq V$, дефинишемо $\Gamma_{(A)} = \cup_{a \in A} \Gamma_{(a)}$. Ивица која спаја неки чвор x са самим собом зове се *петља*. *Прост* граф је граф који нема ни петљи ни *вишеструких ивица*, што значи да ниједна два чвора нису спојена са више од једном ивицом.



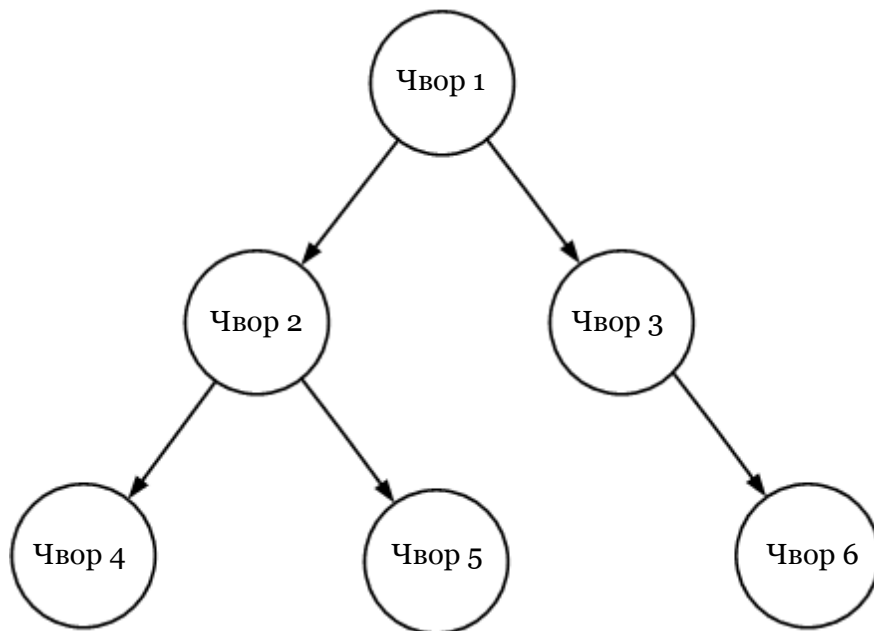
Слика 4.5: Пример једног усмереног графа



Слика 4.6: Пример једног неусмереног графа

Граф може бити неусмерен или усмерен, код кога свака ивица има и свој смер (што значи да ако је v повезано са w , не мора бити и да је w повезано са v). Такође, граф може бити и тежински ако свака ивица себи има додељену неку вредност. Нетежински граф може се посматрати као тежински, с тим што би тежина сваке ивице била једнака (нпр. 1).

Пут од чвора x до неког другог чвора y је низ ивица које спајају ова два чвора, у коме се свака ивица из графа појављује највише једанпут. Ако неки низ ивица спаја чвор x са самим собом, онда се такав низ зове *круг*. Дрво је график који нема кругова. За график кажемо да је *повезан* ако и само ако за свака два његова чвора постоји пут који их спаја. Ако граф није повезан, онда се он може разбити на *повезане компоненте*, дисјунктне по чворовима. Детаљније описана теорија графа може се наћи у [5].



Слика 4.7: Пример графа који се назива дрво

У овом раду темена препрека представљају чворове. Повезивањем свих видљивих чворова са свим осталим чворовима добијамо комплетан граф неопходан за планирања путање. Након формирања графа, неопходно га је претражити у циљу проналажења оптималне путање. Један од могућих алгоритама претраге најкраћег пута у графу је Дајкстрин алгоритам. Име носи по холандском математичару Едшеру Дајкстри (*Edsger Wybe Dijkstra*), који га је објавио 1959. године.

4.2 Дајкстрин алгоритам

Нека је дат тежински усмерени граф G и почетни чвор s из G . Ако скуп свих чворова графа обележимо са V , скуп ивица са E , тада је свака ивица из E , представљена паром чворова (u, v) које она повезује из V . Такође, нека свака ивица добије одређену вредност (тежину) w . Тежина сваке ивице се може представити као растојање између два чвора које она повезује. Дужина пута између два чвора је сума тежина ивица на том

пути. За дати пар чворова s и t из V Дајкстрин алгоритам налази вредност најкраћег пута. Основна операција Дајкстриног алгоритма је ослобађање ивица:

Уколико постоји ивица из u ка v , тада тренутно најкраћи пут из s у u ($d[u]$) може добити вредност суме $d[u] + \omega(u, v)$ и тежине ивице (u, v) . Дакле, његова дужина ће износити $d[u] + \omega(u, v)$, уколико је ова вредност мања од $d[v]$. Процес ослобађања ивица наставља се док вредност $d[v]$ не одређује најкраћи пут из s у v , за сваки чвор v .

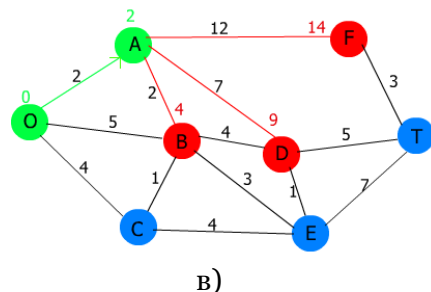
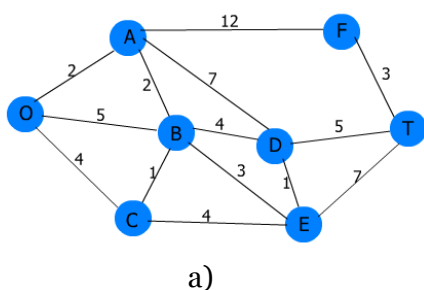
Током извршавања алгоритма издвајају се два скупа чворова S и Q . У скупу S су они чворови за које је позната вредност $d[v]$, а у скупу Q сви остали. На почетку је скуп S празан, а у свакој следећој итерацији један чвор се премешта из Q у S . То је онај чвор који има најмању вредност $d[u]$. На крају се ослобађају све ивице (u, v) горе описаним поступком [6].

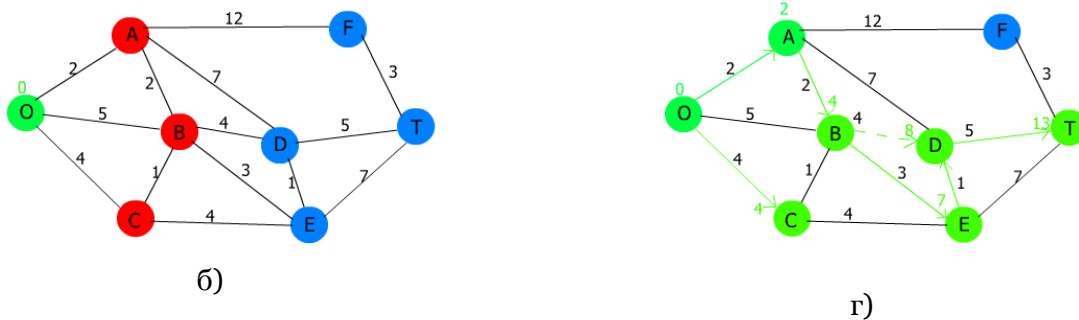
Табела 4.1: Приказ псеудо кода Дајкстриног алгоритма

```

1 function Dijkstra( $G, w, s$ )
2   за сваку ивицу  $v$  у  $V[G]$            // Иницијализација
3      $d[v] := \text{infinity}$ 
4      $\text{previous}[v] := \text{undefined}$ 
5    $d[s] := 0$ 
6    $S := \text{prazan skup}$ 
7    $Q := V[G]$ 
8   while  $Q$  nije prazan skup           // Дајкстрин алгоритам
9      $u := \text{Extract\_Min}(Q)$            // налази чвор  $u$  из скупа  $Q$  који има најмању вредност
 $d[u]$ 
10     $S := S \text{ unija } \{u\}$ 
11    за сваку ивицу  $(u, v)$  која иде из  $u$ 
12      if  $d[u] + w(u, v) < d[v]$        // Ослобађање ивице  $(u, v)$ 
13         $d[v] := d[u] + w(u, v)$ 
14         $Q := Q \text{ unija } \{v\}$ 
15         $\text{previous}[v] := u$ 
16      end
17 end

```





Слика 4.8: Пример претраге једног неусмереног графа од чвора *O* до чвора *T*

Једна од могућих репрезентација графа са слике 4.8 је помоћу матрице суседства приказане у табели 4.2.

Табела 4.2: Репрезентација графа помоћу матрице суседства

| | O | A | B | C | D | E | F | T |
|---|---|----|---|---|---|---|----|---|
| O | 0 | 2 | 5 | 4 | ∞ | ∞ | ∞ | ∞ |
| A | 2 | 0 | 2 | ∞ | 7 | ∞ | 12 | ∞ |
| B | 5 | 2 | 0 | 1 | 4 | 3 | ∞ | ∞ |
| C | 4 | ∞ | 1 | 0 | ∞ | 4 | ∞ | ∞ |
| D | ∞ | 7 | 4 | ∞ | 0 | 1 | ∞ | 5 |
| E | ∞ | ∞ | 3 | 4 | 1 | 0 | ∞ | 7 |
| F | ∞ | 12 | ∞ | ∞ | ∞ | ∞ | 0 | 3 |
| T | ∞ | ∞ | ∞ | ∞ | 5 | 7 | 3 | 0 |

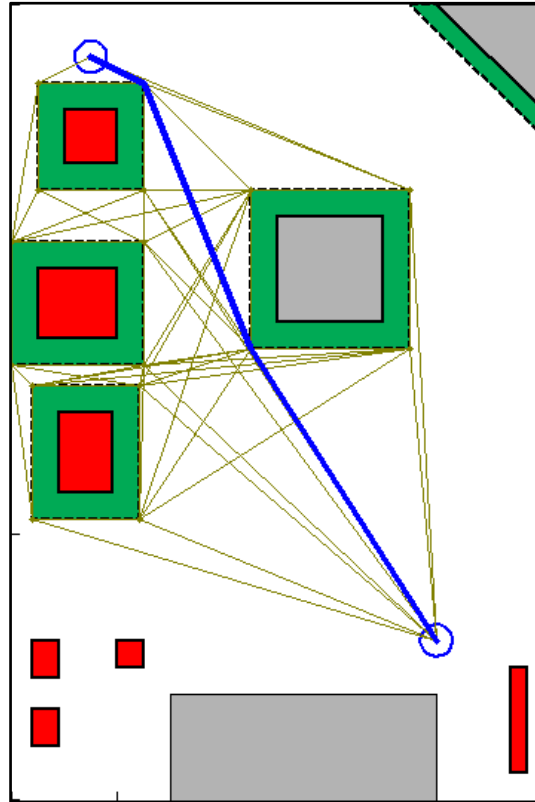
На слици 4.8 а) приказан је један граф од 8 чворова који треба претражити и наћи најкраћу путању од чвора *O* до чвора *T*. Сви неразрешени чворови приказани су плавом бојом, зеленом бојом су означени разрешени чворови, док су црвеном представљени они чије је решавање у току. Дакле, креће се из чвора *O* и рачуна се растојање до свих суседних чворова што је приказано на слици 4.8 б). Затим се узимају следећи чворови, у овом случају *A, B, C* и за њих се понавља поступак као за почетни чвор *O* што је приказано на слици 4.8 в). Исти поступак се понавља за све остале чворове док се не дође до жељеног чвора *T*, слика 4.8 г). Тако добијамо једну или више могућих путања до циља. Сада је потребно израчунати растојање између чворова и за то користимо Еуклидску норму. Математички приказ овог поступка изгледа овако:

$$S = \sqrt{(M_{x_1} - M_{x_2})^2 + (M_{y_1} - M_{y_2})^2} \tag{4.1}$$

- S* – Еуклидска дистанца између 2 чвора
- M_{x_1}, M_{x_2} – Координата *x* првог и другог чвора
- M_{y_1}, M_{y_2} – Координата *y* првог и другог чвора

Након тога, сабирамо растојање између свих чворова који су посећени у једној путањи. То понављамо док не добијемо вредности за све трајекторије које повезују старт и циљ. Путања која има минималну вредност, усваја се као оптимална. Овај

алгоритам који је горе описан и који је примењен у овом раду одступа у малој мери од Дајкстриног алгоритма. Разлика је у томе што се у Дајкстрином алгоритму рачуна пређени пут до сваког чвора који је посећен за време рада алгоритма, док алгоритам примењен у овом раду, прво израчуна све могуће трајекторије и њихове вредности, а затим међу њима одреди оптималну.



Слика 4.9: Приказ модела технолошког окружења након завршавања процеса планирања кретања и добијања оптималне путање

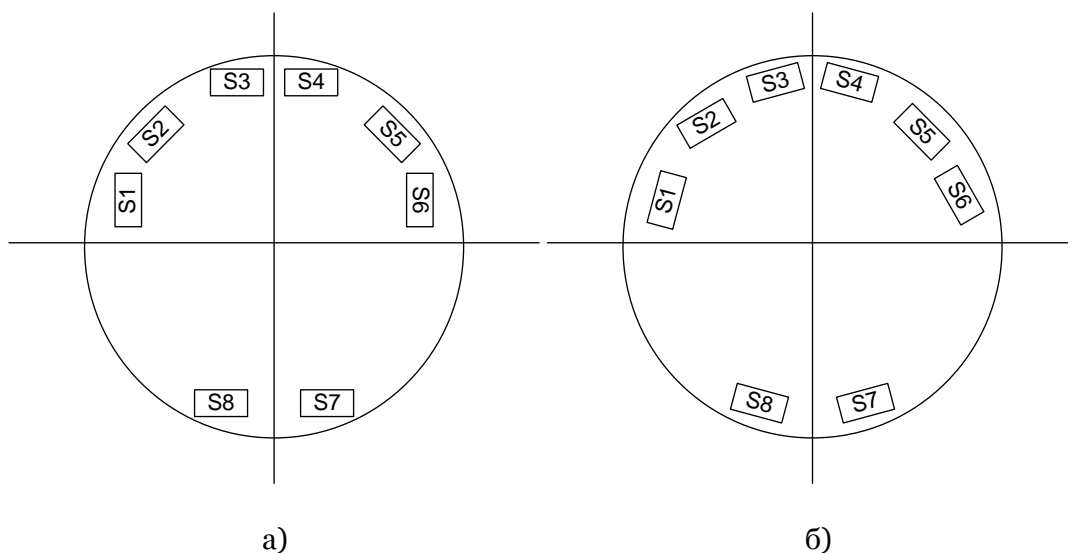
Постављањем модела кретања обезбеђене су информације о положају мобилног робота у сваком тренутку. Одабиром алгоритма за планирање путање и решавањем проблема навигације обезбеђено је успешно планирање и извршавање кретања мобилног робота од почетне позиције до циљне. Коришћењем графа видљивости и Дајкстриног алгоритма за претрагу графова осигурано је да изабрана трајекторија буде уједно и оптимална. Сада је неопходно обезбедити робусност система тј. да интелигентни мобилни робот, који има унапред дефинисану путању и редослед опслуживања машина, буде у могућности да избегне евентуалне препреке у окружењу и настави са обављањем свог задатка.

5. Имплементација метода одлучивања, базираних на машинском учењу, у домену препознавања постојања препрека током кретања мобилног робота

За препознавање објеката у окружењу користе се инфрацрвени сензори. Међутим за доношење одлука неопходно је увести вештачке неуронске мреже које у овом случају имају задатак да препознају поремећаје који су настали у систему и да одреде начин на који интелигентни мобилни робот треба да их превазиђе. Примена вештачких неуронских мрежа има широку заступљеност код интелигентних мобилних робота, највише због њихове способности да нађу корелацију између улазних и жељених излазних података. Метода која је коришћена базира се на машинском учењу, применом система вештачких неуронских мрежа, због њихових могућности да се прилагоде променама, које у систему изазивају поремећајни фактори и да на основу тога уче и минимализују грешку после сваке следеће итерације. Међутим да би знали како правилно да искористимо сензорске информације неопходно је детаљно упознати сам принцип рада сензора, предности и мане, као и њихове лимите. Све ово је детаљно објашњено у наредном поглављу.

5.1 Инфрацрвени сензор

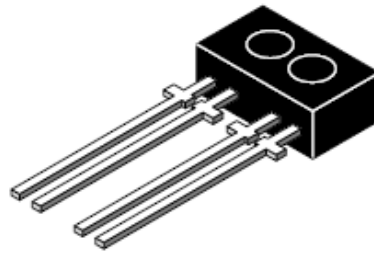
Khepera има 8 инфрацрвених сензора и распоређени су тако да се 6 налазе са предње стране, док су остала два постављена на задњој страни робота. На слици 5.1 је приказан распоред ИЦ сензора на *Khepera* мобилном роботу.



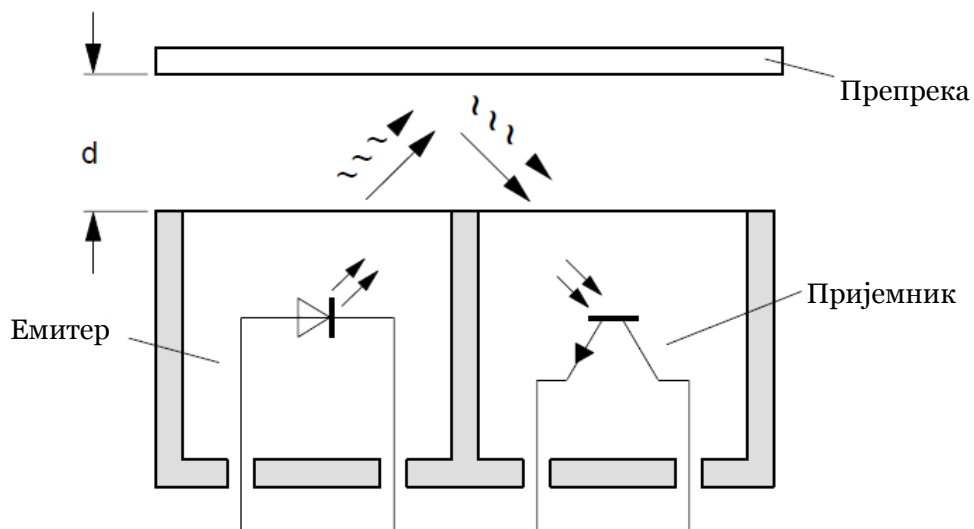
Слика 5.1: а) Приказ упутством декларисаног распореда ИЦ сензора на *Khepera* мобилном роботу б) Стварни распоред ИЦ сензора

Као што се види са слике 5.1 стварни распоред ИЦ сензора одступа од декларисаног распореда, што је додатно отежало рад са њима.

Ови сензори су опремљени емитером ИЦ зрака и пријемником. Сензори носе ознаку TCRT1000, а производи их фирма *Vishay Telefunken*.



Слика 5.2: Изглед ИЦ сензора TCRT1000



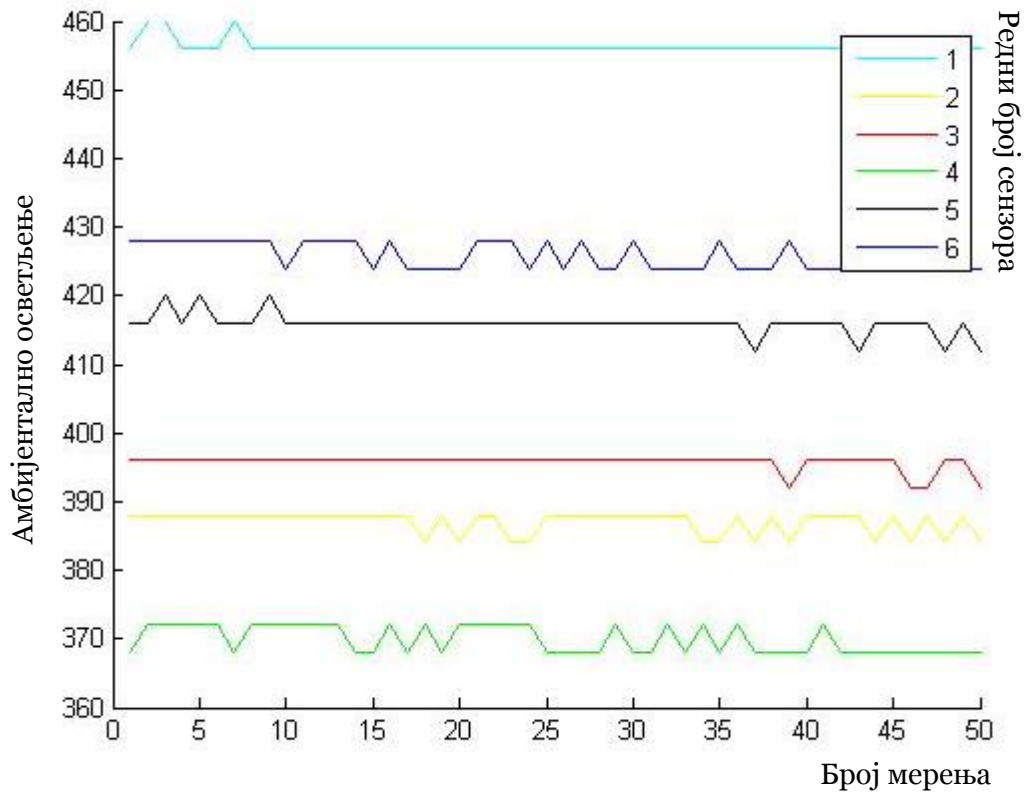
Слика 5.3: Начин рада ИЦ сензора

Сензори омогућују две врсте мерења:

- *Мерење нормалног амбијенталног осветљења.* Мерење се изводи коришћењем само пријемника, без емитовања зрака помоћу емитера. Ново мерење се остварује сваких 20 ms, сензори се читају секвенцијално сваких 2.5 ms.
- *Мерење светлости коју рефлектује неки предмет.* Мерење се обавља емитовањем светлости помоћу емитера. Добијена вредност представља разлику између мерења које је добијено емитовањем светлости и амбијенталног осветљења које је измерено без емитовања светлости. Као и у пређашњем случају мерења се врше сваких 20 ms, а сензори се читају секвенцијално сваких 2.5 ms.

У наредном делу биће приказани резултати за амбијентално осветљење као и за мерења рефлектоване светлости.

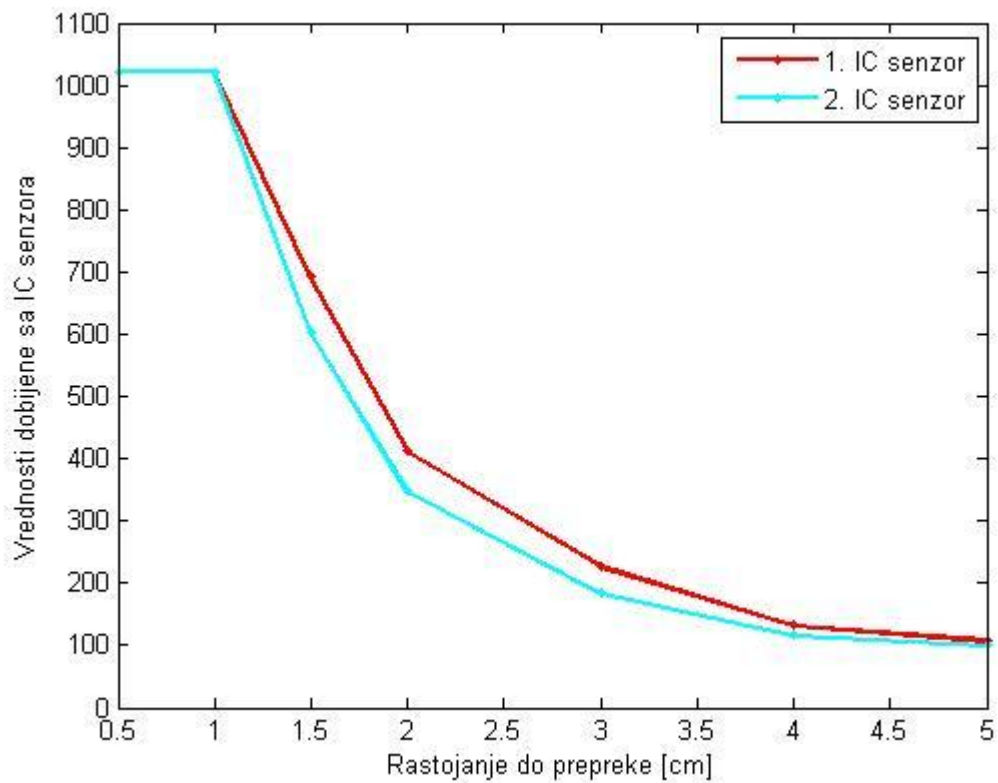
Амбијентално осветљење је од велике важности за перформансе сензора. На резултате мерења највише утиче околина робота. Способност детекције објеката драстично опада са смањењем амбијенталног осветљења и рад у лоше осветљеном окружењу није могућ. У зависности од извора светлости, боје и удаљености вредности амбијенталног осветљења могу да варирају. Емитовање велике количине светлости у домету ИЦ сензора није препоручљиво, јер може довести до ометања рада ИЦ сензора. Слика 5.4 приказује профил осветљења, у нормалним радним условима при којима се обављао највећи део експеримената.



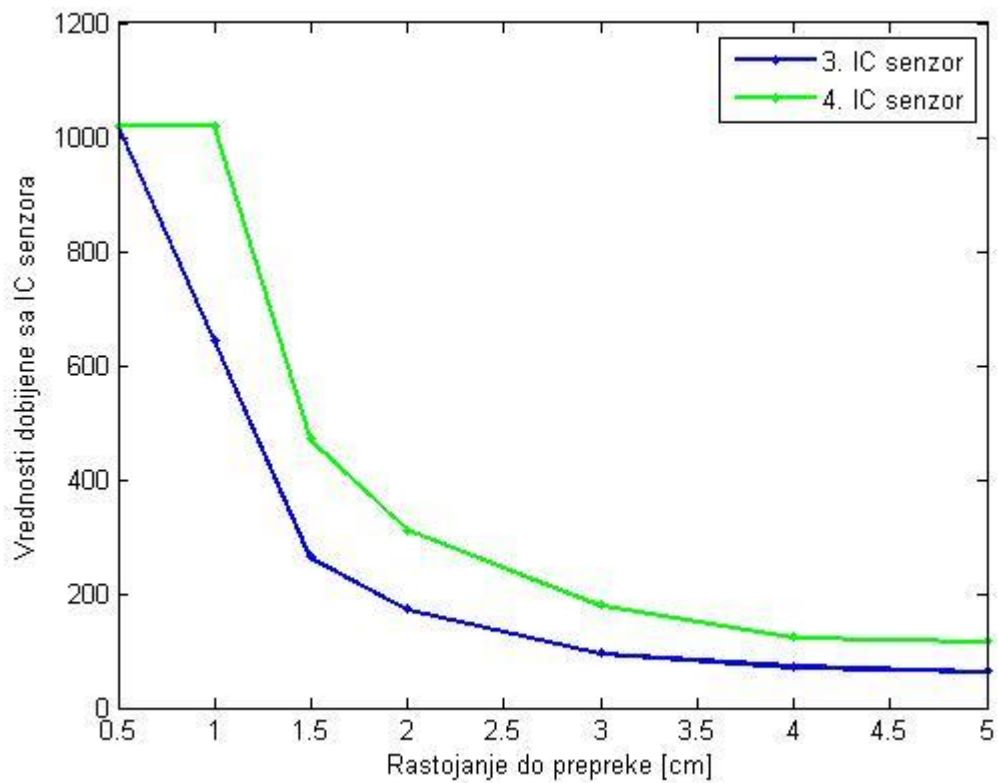
Слика 5.4: Вредности са сензора добијене за време читавања амбијенталног осветљења (приказани су само предњи сензори)

Вредности које враћају ИЦ сензори се повећавају са смањењем интензитета светлости. Максимална вредност је 500 и по упутству обично се достиже у скоро потпуно мрачним условима. Међутим са слике 5.4 видимо да у нормално осветљеној просторији вредности варирају између 400 и 500, што је на горњој граници за успешан рад сензора. Прејакно осветљење у радном окружењу може довести до лоших читавања и неправилног рада ИЦ сензора, зато је најбоље да вредности амбијенталног осветљења буду између 200 и 400.

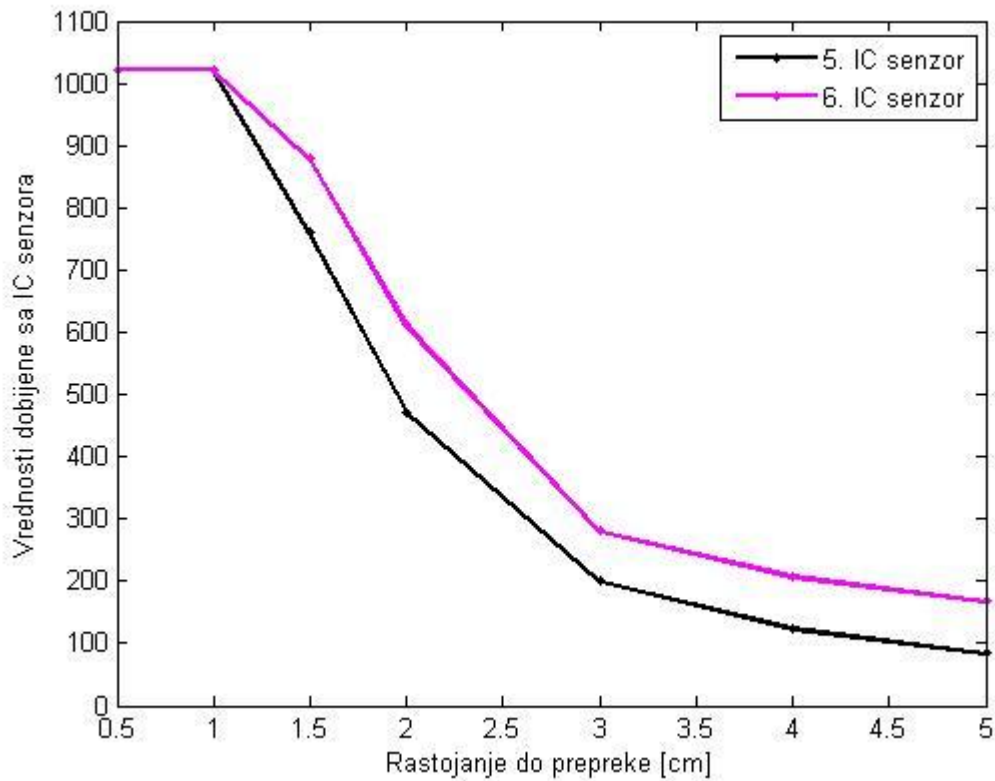
Мерење рефлектоване светлости углавном се користи за детектовање објеката око *Kamera*. Овакво мерење у највећој мери зависи од рефлективности објекта и од амбијенталног осветљења. Боја објекта, материјал и површина такође могу да утичу на резултате мерења. Из ових разлога направљен је график како би се стекао бољи увид у могућности и мане ових ИЦ сензора. За тестирање коришћени су модели машина из технолошког окружења, који су офарбани црвеном бојом. На сликама 5.5, 5.6 и 5.7 су дати резултати мерења.



Слика 5.5: Добијена очитавања са сензора 1 и 2 за растојања од 0.5-5[cm]



Слика 5.6: Добијена очитавања са сензора 3 и 4 за растојања од 0.5-5[cm]



Слика 5.7: Добијена очитавања са сензора 5 и 6 за растојања од 0.5-5[cm]

Идеални услови за коришћење сензора на *Khepera* мобилном роботу јесу при униформној и константној светлости, али сензори су довољно робустни да могу да функционишу исправно у већини случајева.

Како вредности ИЦ сензора брзо варирају у току времена, нека тренутна добијена вредност може бити ирелевантна. Из овог разлога средња вредност мерења сензора се рачуна у току мерног периода. Средња вредност је дефинисана као:

$$IC_{\text{средње}} = \frac{\sum_{n=0}^N IC_n}{n} \quad (5.1)$$

$IC_{\text{средње}}$ – средња вредност мерења
 IC_n – n-то мерење
 n – укупан број обављених мерења

Варијација сензорских вредности у току мерења апроксимирана је коришћењем средње девијације током мерног периода. Што је већа девијација, већа је и варијација вредности од узорка до узорка.

$$Dev = \frac{\sum_{n=0}^N |IC_{\text{средње}} - IC_n|}{n} \quad (5.2)$$

С обзиром да је једини начин да *Khepera* детектује своју околину (рефлектујуће објекте у свим правцима) помоћу 8 ИЦ сензора, правилна употреба сензорског система је од круцијалне важности за успешно решавање задатка, у овом случају избегавање

препрека. Као и сви сензори, тако су и ИЦ сензори подложни шуму и ометањима из околине. Ове појаве могу да доведу до погрешно детектованих препрека или да се деси да препрека уопште и не буде детектована. Генерално постоје два начина за решавање овог проблема. Први, који даје боље резултате, је обезбеђивање робусности система софтверским путем, у овом раду је то урађено коришћењем вештачких неуронских мрежа (ВНМ). Други начин је хардверско модулирање и адаптирање сензора одређеном окружењу. У овом раду акценат је стављен на примену ВНМ, као једне од парадигми вештачке интелигенције, за решавање оваквих проблема као и доношења правовремених одлука од стране робота.

5.2 Вештачке неуронске мреже

Неуронска мрежа је парадигма вештачке интелигенције која се дефинише као конективни модел за резонување заснован на аналогiji са мозгом, уз наглашену когнитивну способност да учи и врши генерализацију стеченог знања. [7]

При моделирању и управљању системима, углавном се претпоставља да општи, аналитички модел система може да се дефинише. Међутим, комплексни системи у машинском инжењерству (машине алатке, роботи, аутомобили, авиони, итд.), као и многи технолошки процеси, толико су компликовани да се општи модел ретко може успоставити. У тим случајевима вештачке неуронске мреже се успешно примењују за решавање неких од проблема (нпр. доношење правовремених одлука), тако да се рецимо на основу анализе технолошког процеса, без експлицитног модела, оне користе као *универзални апроксиматори*. Оне су примењиве и као „софистицирани сензори“, са задатком да изврше естимацију вредности сигнала или променљивих у процесу (нпр. одлучивања), уз пажљив избор скупа стабилних, квалитетних нумеричких података које верно описују сам процес [7].

У општем смислу, неуронске мреже представљају скуп једноставних процесирајућих елемената-неурона, међусобно повезаних везама са одговарајућим тежинским односима. Неуронске мреже имају способност адаптивног понашања 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.

Вештачка неуронска мрежа или конективни модел има три основне компоненте:

1. Неурон - *процесирајући елемент*, прима улазне сигнале/информације од окружења или од других неурона преко синапси или веза које могу бити *ексцитационе* (побуђивачке) или *инхибиторне*. Праг активације неурона је његов „окидач“;
2. Топологија мреже - изглед мреже;
3. Алгоритам учења;

Додатне компоненте су: *величина мреже* (број нивоа-слојева, број неурона у слоју), *функционалност неурона*, *обучавање/валидност* (величина обучавајућег узорка, формат података, итд.), *имплементација/реализација* (аналогна, дискретна, софтверска, итд.)

Једна од најбитних ствари при обучавању ВНМ је одабир алгоритма учења који је приказан у формули (5.3) система спрегнутих нелинеарних диференцијалних једначина :

$$\dot{w}_{ij} = A_i(w_{ij}, x_i, x_j, \dots) \quad (5.3)$$

где је A_i алгоритам учења, w_{ij} вектор тежинских односа, x_i, x_j улазни вектори. Најзначајнији алгоритми учења су: градијентални поступак, Хебово правило, Видров-Хофово правило, Њутн – Рафсонов метод, генерализано делта правило и Левенберг-Маркеов алгоритам учења који је управо и коришћен у овом раду.

5.2.1 Левенберг – Маркеов алгоритам учења

Овај алгоритам је први пут формулисан 1944. године. Левенберг је предложио да се модификација вредности параметара врши на следећи начин:

$$W^{(k+1)} = W^k - \mu(H - \lambda I)^{-1} \nabla J(W) \quad (5.4)$$

где је са H означена Хесијан матрица, I представља јединичну матрицу, а λ је константан параметар. Овакво дефинисано правило модификације даје на већем значају локалној кривини функције циља и на неки начин представља средину између *алгоритма најмањег градијента* и *Њутновог поступка*.

Прво унапређење основне формулације овог алгоритма предложио је Марке 1963. године увођењем дијагоналних елемената Хесијан матрице уместо јединичне матрице коју је предложио Левенберг. На основу овог предлога правило модификације је:

$$W^{(k+1)} = W^k - \mu(H - \lambda \text{diag}(H))^{-1} \nabla J(W) \quad (5.5)$$

што представља Левенберг – Маркеов алгоритам. Дакле, када λ има велику вредност алгоритам спроводи градијентални поступак, али ипак можемо искористити ову информацију о локалној закривљености приликом ажурирања параметара, што иницијална формулација алгоритма не дозвољава [9].

5.2.2 Активационе функције и модели неуронских мрежа

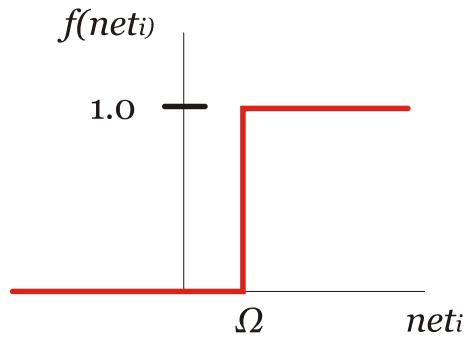
Функционалност неурона обезбеђује мрежи способност репрезентовања знања и чине је три саставна елемента:

1. Улазни оператор;
2. Функција преноса;
3. Активациона функција;

Активационе функције играју веома важну улогу у различитим моделима вештачких неуронских мрежа. Оне се могу сврстати у неколико заједничких група као што су: *линеарне*, *бинарне*, *сигмоидне*, *компетитивне* и *Гаусове*.

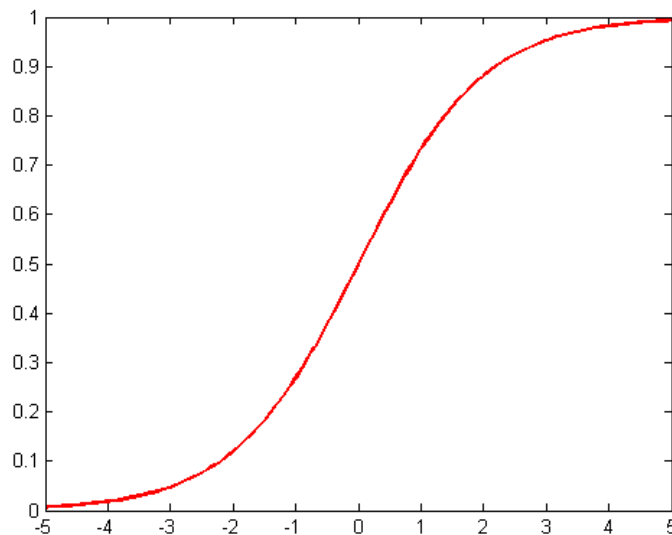
Линеарне активационе функције практично представљају суму свих улаза у неурон, пропуштајући их на излазу из неурона интегрисане у кохерентном облику. Ово су најједноставније активационе функције.

Бинарне функције дају неурону два стабилна стања, активно и неактивно.



Слика 5.8: Линеарна активациона функција

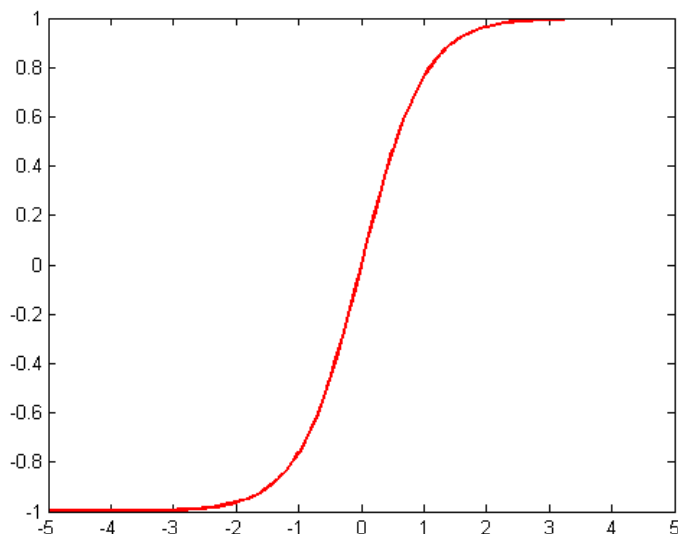
Сигмоидна или S-активациона функција функционише слично бинарној, тако што генерише излазни сигнал који има такође два стабилна стања. Разлика између ове две активационе функције је у томе што је сигмоидна математички континуална и диференцијабилна, а може се рећи и да је нелинеарна и неоппадајућа. Сигмоидна функција асимптотски тежи 0 и 1, па су све вредности преко 0.8 активни излази, а вредности испод 0.2 су неактивни излази.



Слика 5.9: matlab „logsig“ активациона функција

$$\text{logsig}(n) = \frac{1}{(1 + \exp(-n))} \quad (5.6)$$

При обучавању вештачке неуронске мреже, која је употребљена у овом раду, коришћена је “tansig” сигмоидна функција, као активациона функција.



Слика 5.10: *matlab „tansig“ активациона функција*

$$tansig(n) = \frac{2}{(1 + \exp(-2 \cdot n))} - 1 \quad (5.6)$$

Модели неуронских мрежа (неурона) су најчешће динамички, јер се развијају у функцији од времена, па је потребно и тежинске односе између неурона посматрати као динамички систем. То значи да имамо систем диференцијалних једначина за тежинске односе. Процес учења, базиран на алгоритмима учења, одређује тежинске односе кроз итеративни поступак њихове модификације. Систем вештачке неуронске мреже на тај начин учи. Примена система вештачких неуронских мрежа је веома широка. Користе се у различитим областима производног инжењерства, при чему су се издвојиле три основне категорије употребе: *класификација*, *предикција* и *функционална апроксимација*.

Најраспрострањенији модели вештачких неуронских мрежа су:

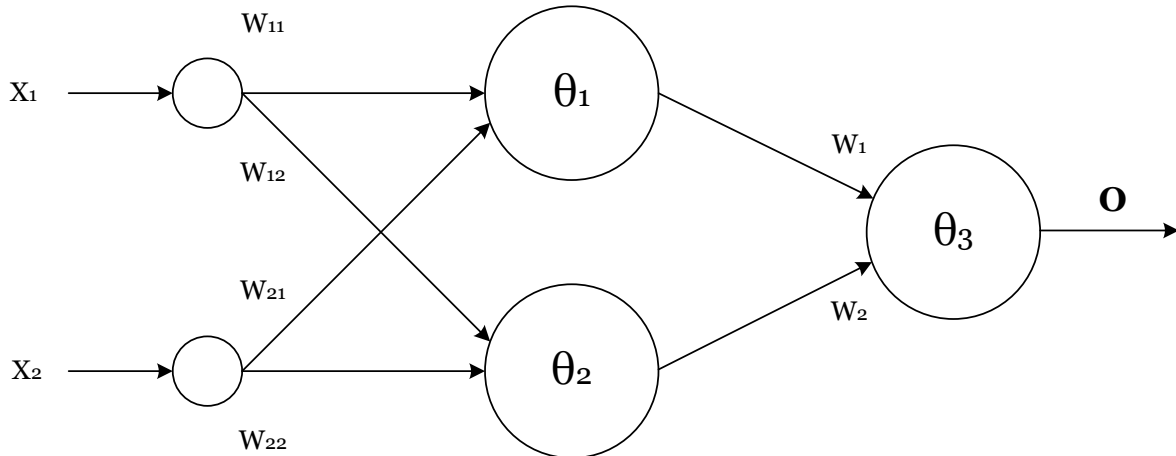
- Перцептрон;
- „Backpropagation“ (BP) неуронска мрежа;
- Асоцијативне неуронске мреже;
- Hopfield-ове неуронске мреже;
- ART неуронске мреже (ART-1, ART-2, ART-3);
- *Fuzzy* асоцијативне неуронске мреже;
- Самоорганизујуће неуронске мреже.

5.2.3 Перцептрон

Перцептрон представља најранији модел вештачке неуронске мреже, а први пионирски радови датирају од Warren McCulloch-а и Walter Pitts-а из 1943. године, па преко Rosenblatt-а (1958 и 1962) до Minsky-ог и Papert-а (1969). Архитектура

перцептрона је једноставна, *feed-forward*² мрежа која има један или више слојева неурона између улазних и излазних неурона.

Најрепрезентативније објашњење архитектуре перцептрона, као система вештачке неуронске мреже, дао је Rosenblatt. Структуру перцептрона чини скуп улазних неурона, скуп скривених неурона и један излазни неурон. Rosenblatt-ова основна идеја је била везана за то, да би јачину веза између улазног слоја неурона и скривеног слоја неурона требало случајно изабрати по неком закону вероватноће и фиксирати њихове вредности током целог процеса учења. Алгоритам учења би затим могао да реализује подешавање тежинских односа између скривеног слоја неурона и јединог излазног неурона [8].



Слика 5.11: Елементарна структура перцептрона

Код ове неуронске мреже која користи линеарни тип неурона са прагом θ , излазна вредност је дата следећим изразом:

$$o = f(net) = \begin{cases} 1 & net \geq \theta \\ 0 & net < \theta \end{cases} \quad (5.7)$$

$$net = \sum W_{ij} X_j \quad (5.8)$$

Алгоритам за учење перцептрона се може представити кроз шест корака. Констатовано је да процес учења увек конвергира у коначном броју итерација приликом коректне класификације свих чланова обучавајућег скупа и да перцептронски алгоритам учења тражи минималну вредност активационе функције.

Корак 1: Иницијализација. Нека је $t=0$ и нека вектор иницијалних тежинских односа $w(t) \in \mathbb{R}^a$ дефинише иницијално пресликавање од скривених неурона до излазног неурона.

Корак 2: Изабрати стимулацију (подстреке) перцептрона примењујући узорке из обучавајућег скупа x^k на случајан начин, што се може исказати помоћу $x^k(t) = (s^k(t), o^k(t)) \in \{0,1\}$, где је o^k циљни вектор, који представља захтевани одговор на стимулацију.

² Мреже са простирањем сигнала унапред.

Корак 3: Израчунати активацију неурона у скривеном слоју h^k , на основу активације улазних неурона, $h^k(t) = q(s^k(t))$.

Корак 4: Израчунати активацију излазног неурона мреже r^k , на основу активације неурона из скривеног слоја, $r^k(t) = f(net) = f(w(t)^T, h^k(t))$, при чему је $f(net) = 1$ ако је $net \geq 0$ и $f(net) = 0$ ако је $net < 0$.

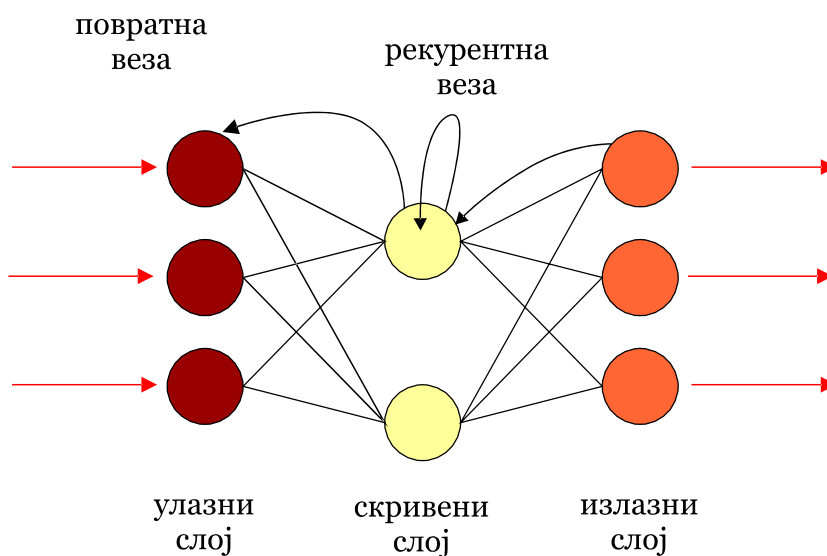
Корак 5: Модификовати тежинске односе између скривених неурона и излазног неурона, $w(t+1) = w(t) + [o^k(t) - r^k(t)]h^k(t)$.

Корак 6: Зауставити учење ако је $t > t_{max}$ или $o^k(t) = r^k(t)$, за $k = 1, \dots, n$, иначе нека је $t = t + 1$ и тада је неопходно вратити се на корак 1.

Алгоритам за учење перцептрона преузет је из [8].

5.2.4 Backpropagation (BP) неуронска мрежа

Backpropagation (BP) неуронска мрежа је развијена са циљем да се проблем нелинеарног пресликавања из улазног простора у излазни простор успешно реши, при чему се остварује модификација тежинских односа и између улазног и скривеног слоја неурона. BP је неуронска мрежа са простирањем сигнала унапред, која такође реализује супервизорски вид учења, са различитим активационим функцијама и алгоритмима учења. BP мрежа користи *градијентни поступак* при обучавању. То значи да учи пресликавања из улазног простора узорка у излазни простор, кроз процес минимизације грешке између актуелног излаза који је остварила мрежа и захтеваног излаза, на основу скупа обучавајућих парова, односно примера. Процес учења почиње са презентацијом улазног облика узорка BP мрежи, који простирањем кроз мрежу остварује излазни облик. BP мрежа затим примењује *генерализано делта правило* да би се утврдила грешка на излазу, коју простирањем уназад преко скривеног слоја, користи за „лагано“ модификовање сваког тежинског односа између неурона, што се понавља за сваки нови узорак. Генерализано делта правило обезбеђује конвергенцију процеса учења до задатог нивоа тачности кроз итеративни процес адаптације тежинских односа [7].



Слика 5.12: Backpropagation (BP) неуронска мрежа

5.2.6 МРФ Мреже

Мреже са радијалним функцијама³ (у даљем тексту МРФ) су се појавиле као варијација ВММ крајем 1980-их. Међутим, њихови корени потичу из много старијих модела и техника препознавања као што је кластеровање, апроксимација функција, *spline*⁴ интерполација и многе друге. Улаз у МРФ мрежу је нелинеаран док је излаз линеаран. Њихове одличне апроксимационе способности су проучаване у [10],[11]. Захваљујући њиховим способностима нелинеарне апроксимације, МРФ мреже могу да врше комплексно мапирање, док перцептрон неуронске мреже могу да праве само модел у смислу вишеструких посредничких слојева [12].

Да би могли да користимо МРФ мреже морамо прво одредити активациону функцију скривеног слоја, број процесирајућих јединица (неурона) и алгоритам обучавања за проналажење параметара мреже. Проналажење МРФ тежинских односа назива се обучавање мреже. Ако имамо скуп улазних парова, назван обучавајући скуп, онда оптимизујемо параметре мреже како би поклопили излазе из мреже са вредностима жељеног излаза. Поклопање се рачуна по одређеном критеријуму, обично је то средња квадратна грешка. Након тренирања, МРФ мреже могу да се користе за податке који су слични подацима из обучавајућег скупа. МРФ мреже имају широки спектар примене укључујући интерполацију, идентификацију система, моделирање параметара електронских уређаја, препознавање говора, рестаурација слика, моделовање 3Д објеката, естимацију кретања и у многе друге сврхе [13].

Када се МРФ мреже користе за решавање неког проблема неопходно је одредити :

- Број неурона у скривеном слоју;
- Активациону функцију;
- Параметре обучавања;
- Тип учења (надгледано, ненадгледано или комбиновани приступ);
- Алгоритам учења;

5.2.7 Топологија МРФ мрежа

МРФ су *feed-forward* неуронске мреже. Такве мреже карактерише улазни скуп и излазни скуп. Између улаза и излаза постоји слој који се састоји од процесирајућих јединица неурона који се назива скривени слој. Разне функције се могу користити као активационе. Најчешће коришћена је Гаусова функција. Гаусова активациона функција у МРФ мрежама дата је изразом:

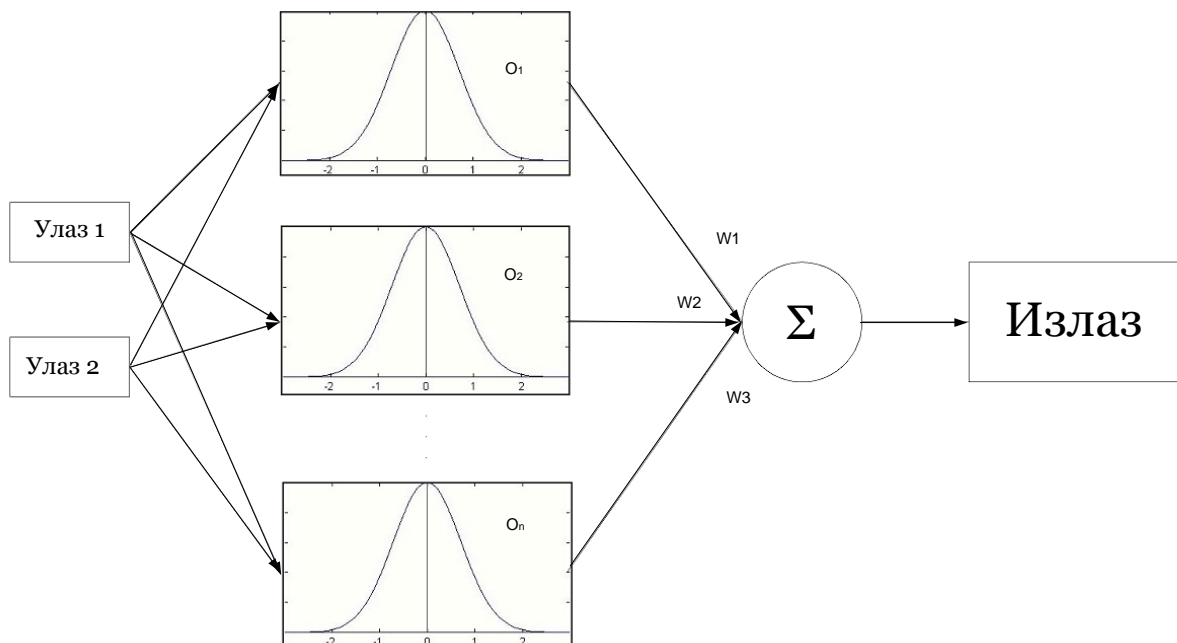
$$\phi_j(X) = \exp \left[-\frac{1}{2} (X - \mu_j)^T \Sigma_j^{-1} (X - \mu_j) \right] \quad (5.9)$$

за $j = 1, \dots, L$, где је X улазни вектор, L је број неурона у скривеном слоју, μ_j и Σ_j су средња вредност и матрица коваријансе j -те Гаусове функције. Вектор μ_j представља локацију, док Σ_j представља облик активационе функције. Пошто је матрица

³ Мреже са радијалним функцијама – енгл. *radial bias function*

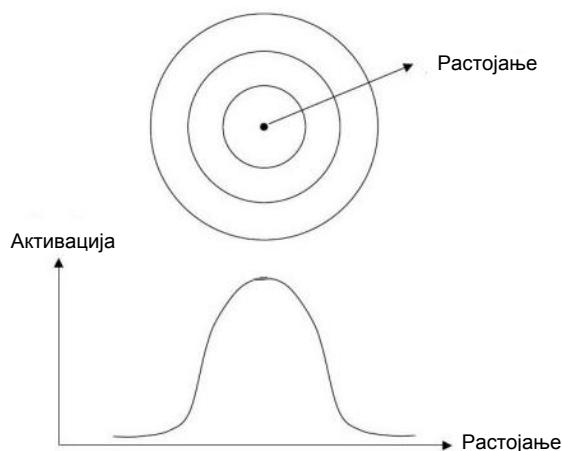
⁴ Израз *spline* се односи на широку класу функција које се користе за интерполацију и изглађивање података

коваријансе Σ_j симетрична, онда свака функција има $\frac{d(d+3)}{2}$ независних подесивих параметара (где d представља димензију улазног вектора). Детаљно описано у [14].



Слика 5.13: Приказ МРФ мреже са три слоја (улазни, скривени и излазни слој)

У скривеном слоју сваки неурон се састоји од МРФ центриране у тачки. Центар и ширина МРФ функције се одређују у процесу тренирања. Када добију вредности улазних вектора, неурони у скривеном слоју рачунају Еуклидску дитанцу (једначина 4.1) и на основу тога се утврђују тежински односи (утицаји) на сваки од неурона. *Radial basis function* је добила име због тога што је полупречник раздаљина која се користи као аргумент у активационој функцији. Што је неурон даље од тачке која се рачуна, мање утицаја има.



Слика 5.14: Приказ утицаја растојања неурона на његову активацију

Вредности које излазе из неурона у скривеном слоју помножене су са тежинским односом неурона (у овом случају W_1, W_2, \dots, W_n) затим се сумирају где се додају тежинске вредности и ова сума представља излаз из мреже. Излазни слој имплементира тежинске односе добијене у скривеном слоју:

$$y_k(X) = \sum_{j=1}^L w_{jk} \varphi_j(X) \quad (5.10)$$

за $k = 1, \dots, M$ где су λ_{jk} излазни тежински односи, сваки одговара вези између неурона у скривеном слоју и неурона у излазном слоју, а M представља број неурона у излазном слоју.

5.2.8 Обучавање МРФ мреже

У процесу обучавања се одређују следећи параметри:

- Број неурона у скривеном слоју
- Координате центра активационе функције сваког неурона у скривеном слоју
- Радиус сваке функције у свим димензијама
- Тежински односи који се додају свакој МРФ функцији док се шаљу на сумирање

Број неурона се може одредити на 2 начина. Један је да сама мрежа одреди њихов број, док је други начин да се број неурона унапред зада. Неки од алгоритама за проналажење оптималног центра и радиуса неурона могу се пронаћи у [15],[16].

Као што је раније речено, повећање броја неурона у скривеном слоју не значи да ће добијени излаз из мреже бити бољи. У табели 5.1 приказани су резултати обучавања МРФ мреже где је једини параметар који је мењан управо број неурона у скривеном слоју.

| <i>Табела 5.1: Средња вредност и стандардна девијација корена средње квадратне грешке</i> | | |
|---|--------|----------|
| Број неурона | m_rmse | std_rmse |
| 5 | 0.6635 | 0.2213 |
| 10 | 0.5553 | 0.2429 |
| 15 | 0.5300 | 0.1674 |
| 20 | 0.4792 | 0.1437 |
| 50 | 0.4847 | 0.1379 |
| 100 | 0.4589 | 0.1405 |
| 500 | 0.4657 | 0.1527 |

Из табеле 5.1 се види да је најмања грешка, за овај пример, када у скривеном слоју постоји 100 неурона. Међутим такође се види да се грешка знатно смањује до 20 неурона, а затим до 500 се јако мало мења. Постоји друга велика разлика, а то је време обучавања. За 20 неурона време обучавања је око 20 секунди, док се с повећањем броја неурона повећава и време обучавања. За 500 неурона време обучавања износи око 60

минута. Иако време обучавања у off line режиму није од пресудне важности, за обучавање мреже у овом раду је коришћено 20 неурона (детаљније описано у поглављу 8).

5.2.9 Поређење МРФ мреже са вишеслојним перцептроном

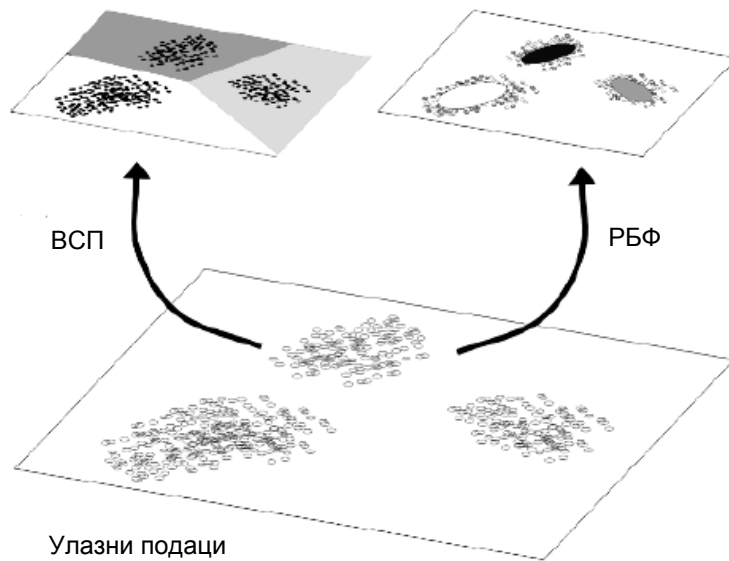
МРФ мреже и вишеслојни перцептрони (ВСП) су примери нелинеарних *feedforward* мрежа. Обе мреже су универзални апроксиматори. Углавном постоје МРФ мреже које су способне да прецизно опонашају ВСП, и обрнуто. Међутим, ове две мреже се ипак разликују у неколико важних аспеката:

1. МРФ мрежа у свом основном облику има један скривени слој, док ВСП може имати један или више скривених слојева.
2. Неурони који се налазе у скривеном или излазном слоју код ВСП мрежа имају исти неуронски модел. С друге стране неурони у скривеном слоју МРФ мреже се знатно разликују и имају различиту сврху него они у излазном слоју.
3. Скривени слој код МРФ мрежа је нелинеаран, али је излаз линеаран. Скривени и излазни слој код ВСП мрежа је обично нелинеаран. Када се ВСП користи за решавање нелинеарних регресионих проблема, преферира се да излаз буде линеаран [14].
4. Активационе функције у сваком неурону у скривеном слоју код МРФ мрежа рачунају Еуклидску дистанцу између улазног вектора и центра тог неурона. Док активациона функција неурона у скривеном слоју код ВСП мрежа рачуна скаларни производ улазног вектора и тежинских односа.
5. ВСП мрежа прави глобалну апроксимацију за нелинеарно пресликавање из улазног у излазни слој. С друге стране МРФ мреже коришћењем нпр. Гаусове функције праве локалне апроксимације за нелинеарно пресликавање из улазног у излазни слој.

Ово у суштини значи да за апроксимацију нелинеарног пресликавања из улазног у излазни слој, ВСП мреже захтевају мањи број параметара него МРФ мреже за исти степен тачности.

МРФ мрежа се разликује од ВСП по томе што има способност имплементирања произвољне нелинеарне трансформације из улазног простора. Ово се јако добро види на XOR⁵ проблему, који не може бити решен ни са једним линеарним перцептроном већ само са МРФ мрежама[17].

⁵ Проблем логичког **ИЛИ** (XOR – eXclusive OR)



Слика 5.15: Приказ начина класификације података код ВСП и МРФ мрежа. ВСП то ради помоћу хиперравни којима раздваја регионе, док се МРФ служи кластеровањем

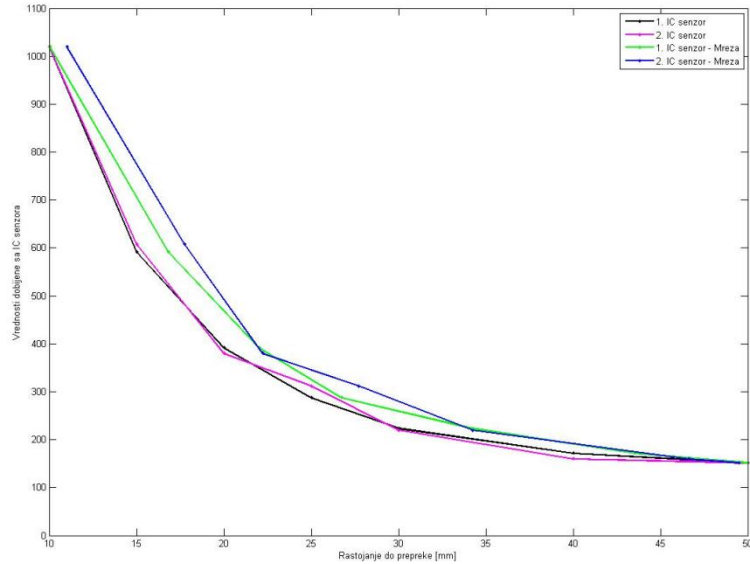
5.3 Примена вештачких неуронских мрежа у домену препознавања постојања препрека

Овај део проблема се односи на успешну имплементацију метода одлучивања, базираних на машинском учењу, у домену препознавања постојања препрека током кретања мобилног робота.

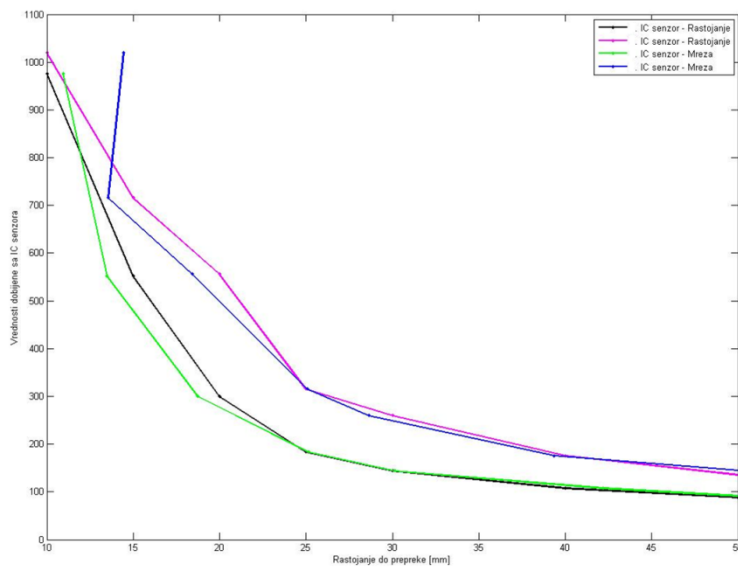
Постоји више метода одлучивања која могу да се искористе за решавање овог проблема. Почевши од, сада већ превазиђених, експертних система, преко правила базираних на индукцији, до техника *soft-computing-a*: *fuzzy* система и генетичких алгоритама. У овом раду је акценат стављен на примену вештачке интелигенције, односно машинског учења за решавање проблема доношења одлука. За ово су коришћене две различите врсте ВМ. МРФ мреже и вишеслојни перцептрон. МРФ мреже као одлични апроксиматори искоришћене су за одређивање растојања до препреке на основу добијених вредности са ИЦ сензора. Дакле, сензорске информације представљају улаз у ВМ, док је излаз дат у [mm].

Да би робот уопште могао правовремено да примети препреку, неопходно је стално добијати нове сензорске информације, што са собом аутоматски повлачи и стални рад МРФ мреже за одређивање растојања. Зато је јако битно да та мрежа даје тачне вредности, јер тиме омогућујемо осталим мрежама да правовремено донесу очекиване одлуке и самим тим умањујемо шансу да дође до колизије робота са објектима у његовом окружењу. Пошто имамо осам ИЦ сензора, сходно томе имамо и осам МРФ мрежа, по једна за сваки сензор. Свака мрежа је посебно обучавана за вредности од одређеног сензора. Током аквизиције података потребних за обучавање мреже извршено је укупно 1200 мерења по сензору, односно по 100 мерења за сваки од 12 случајева, тј. растојања до препреке. Мерења су вршена на растојањима од 1 до 2 [cm] са кораком од 2 [mm], док је на растојању од 2 до 5 [cm] корак био 5 [mm]. Укупан број мерења износи $8 \cdot 1200 = 9\ 600$. Мреже су обучаване само са 50% прикупљених података, док је остатак искоришћен за њихово тестирање и провере извршеног обучавања. Сензори су такви да на растојањима до 2 [cm] могу да примете и помераје од 1 [mm],

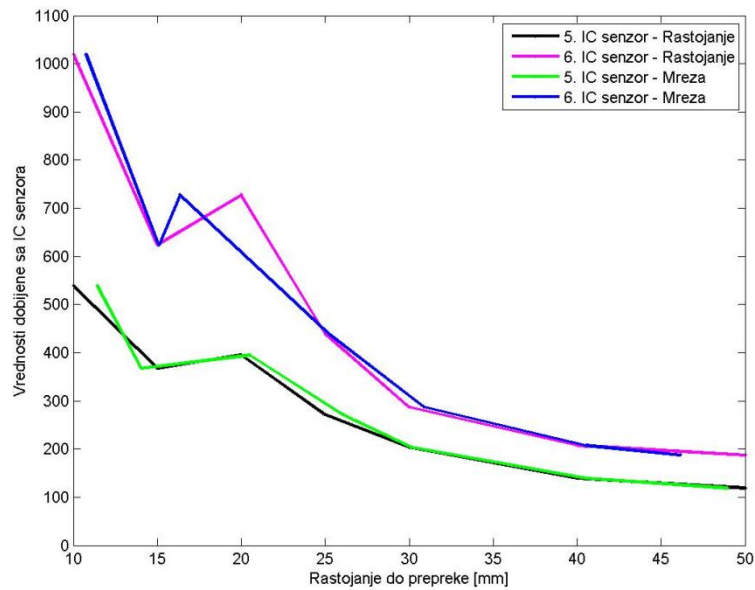
али на растојањима од 3 и више [cm] постаје теже утврдити помераје мање од 5 [mm]. У поглављу 8 дати су експериментални резултати добијени при обучавању ових мрежа, док су на сликама 5.16, 5.17 и 5.18 приказане разлике у стварним растојањима и вредностима које су добијене као излаз из неуронских мрежа након извршеног обучавања.



Слика 5.16: Приказ вредности добијених са сензора и вредности добијених од ВММ за сензоре 1 и 2



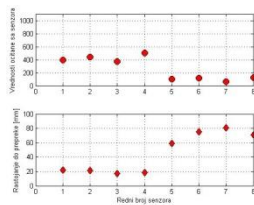
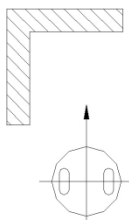
Слика 5.17: Приказ вредности добијених са сензора и вредности добијених од ВММ за сензоре 3 и 4



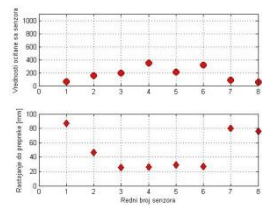
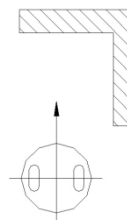
Слика 5.18: Приказ вредности добијених са сензора и вредности добијених од ВММ за сензоре 5 и 6

Вишеслојни перцептрон, због својих способности класификације, искоришћен је управо за класификацију препрека. Након извршене анализе технолошког окружења, утврђени су кључни случајеви који ће бити коришћени за обучавање вештачких неуронских мрежа. Сваки случај представља јединствену позицију препреке на коју робот може да наиђе у радном окружењу. Препреке које могу да поремете систем су класификоване на десет различитих случајева.

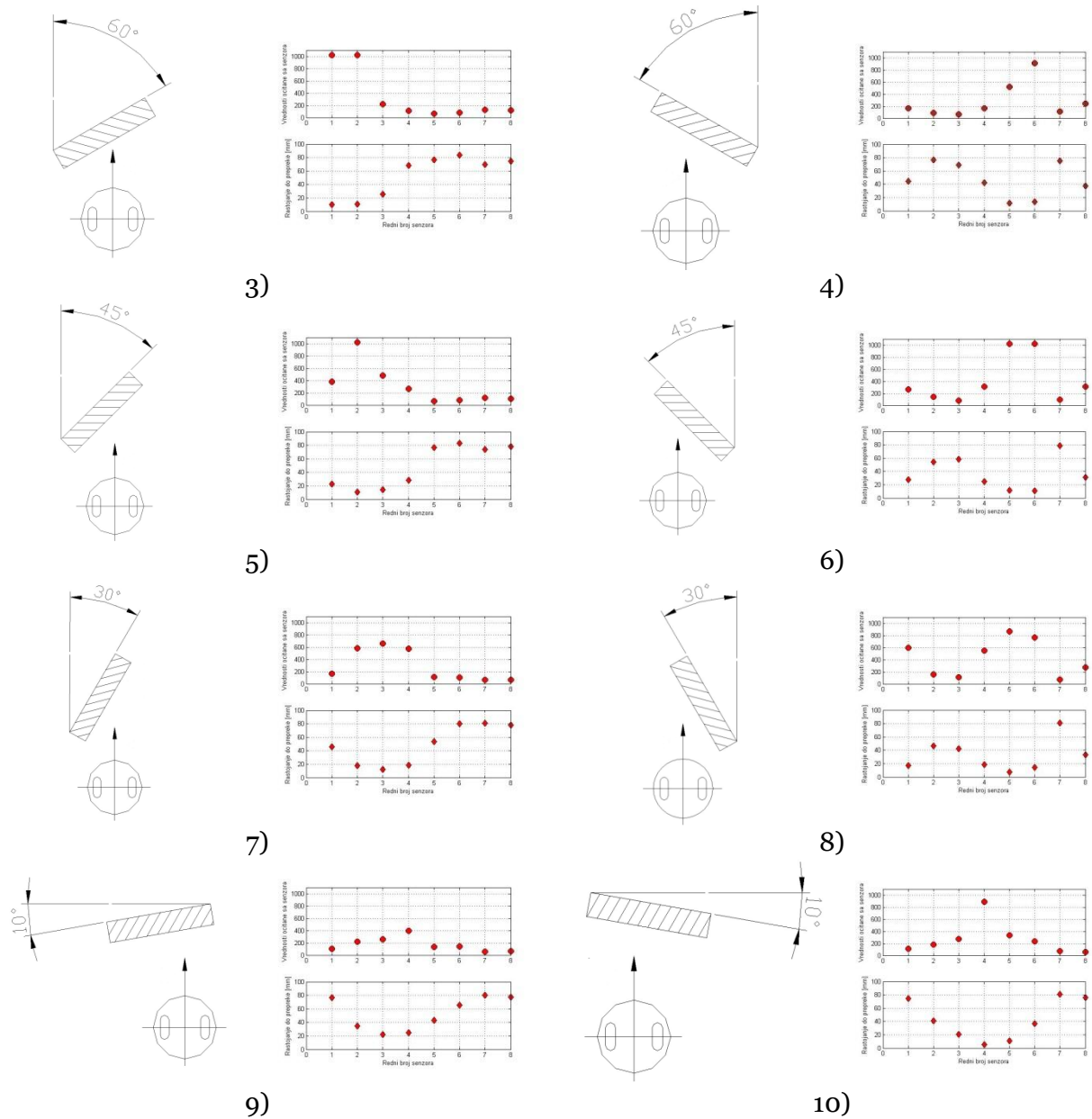
- 1) Ђошак са леве стране, под углом 90° ;
- 2) Ђошак са десне стране, под углом од 90° ;
- 3) Препрека са леве стране под углом од 60° ;
- 4) Препрека са десне стране под углом од 60° ;
- 5) Препрека са леве стране под углом од 45° ;
- 6) Препрека са десне стране под углом од 45° ;
- 7) Препрека са леве стране под углом од 30° ;
- 8) Препрека са десне стране под углом од 30° ;
- 9) Препрека са леве стране од 10° у односу на X осу;
- 10) Препрека са десне стране од 10° у односу на X осу;



1)

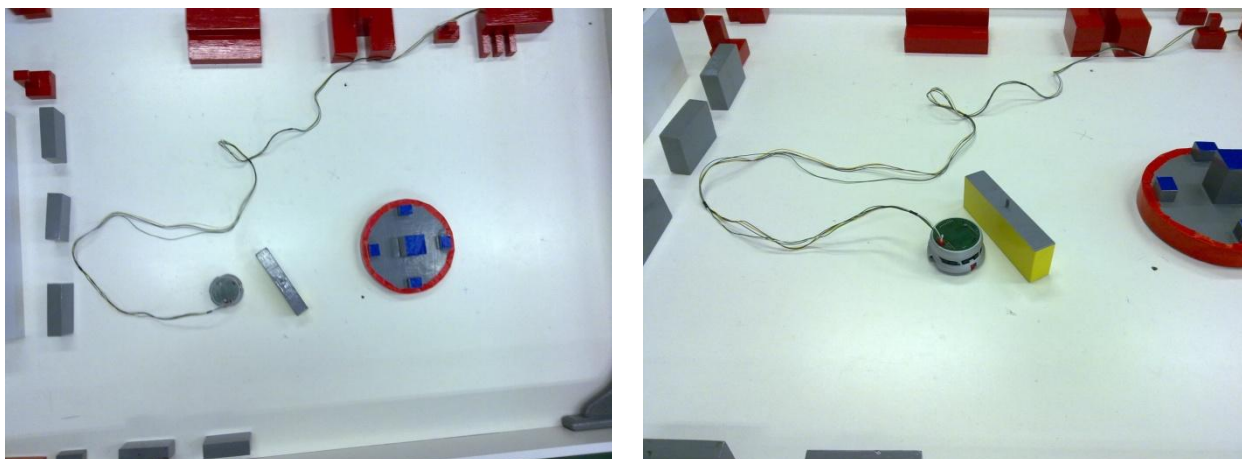


2)



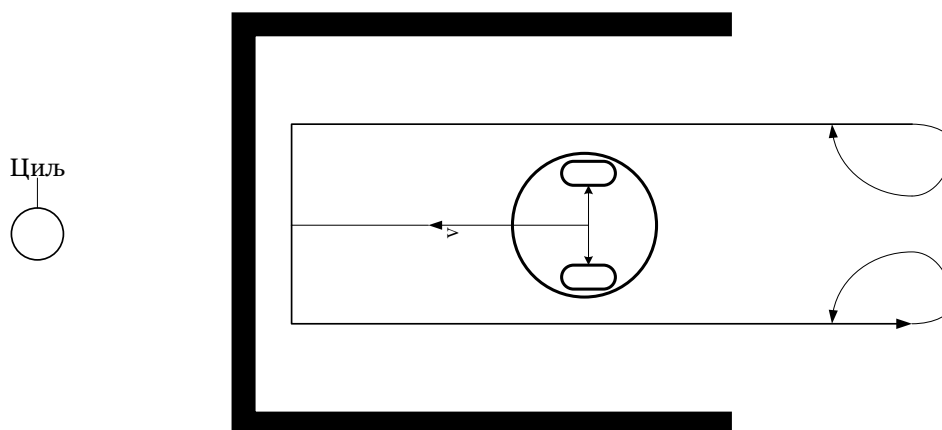
Слика 5.19: Скице карактеристичних препрека као и одговарајућа очитавања сензора

С обзиром да су различита очитавања за различита растојања робота од препрека, што је и нормално, уведене су две ВМ мреже за доношење одлука које се тичу заобилажења препрека. Наиме, једна ВМ мрежа је обучавана за свих осам случајева, али када су препреке на растојању од 2 [cm] до робота, док су за другу коришћени подаци за обучавање који су добијени када је робот на удаљености 4 [cm] до препреке. Процент успешности препознавања препрека са растојања од 4 [cm] није задовољавајући и систем се не може ослонити само на одлуке које та мрежа донесе. Првенствена сврха ове мреже је рано детектовање препрека, која даје роботу информацију о евентуалном објекту који се налази испред њега. Након тога робот смањује брзину како би се што боље припремио да изврши очитавања на 2 [cm] до препреке. Након ових очитавања робот са задовољавајућом успешношћу, која је детаљно приказана матрицом одлучивања у поглављу 7 (табела 7.1), доноси одлуке о томе како и за који угао треба да се заокрене да би благовремено избегао препреку која се налази испред њега.



Слика 5.20: На слици са леве стране робот наилази на препреку, а десна слика приказује успешно препознавање и избегавање

У току кретања робота може јавити проблем, познат под именом проблем локалног минимума. Робот се тада врти у бесконачној петљи из које не може да изађе. Пример је дат на слици 5.21.



Слика 5.21: Приказ локалног минимума који може да се деси у току кретања робота

На слици 5.21 дат је приказ једне U препреке, коју робот не може успешно да заобиђе. У овом случају робот ће доћи до зида скренути лево или десно пратити зид до краја и изаћи из препреке, али у супротном смеру од жељеног. Онда ће поново урадити планирање путање са том стартном позицијом и оријентацијом и нова путања ће га вратити назад у препреку у положај који је приказан на слици 5.21. Једно од решења овог проблема могуће је извести помоћу мапирања простора, тј. да робот када наиђе на овакву препреку, у току кружења унутар ње прикупи сензорске информације о облику препреке, који би се унели у модел окружења као неки вид „привремене препреке”. Тако да када робот изађе из препреке и покрене алгоритам за планирање путање, алгоритам ће имати информације о постојању и положају те препреке и генерисаће путању којом ће је робот заобићи.

Цео процес започиње уношењем почетних и циљних координата робота. Затим се те координате проверавају да ли су физички оствариве, ако јесу наставља се даље извршавање програма, ако нису, враћа се на почетак где се поново уносе координате старта или циља у зависности које су биле погрешне. У случају исправних координата, оне се прослеђују у функцију⁶ за формирање графа видљивости, који се затим претражује и проналази оптимална путања по критеријуму најмањег пређеног пута до циља. Затим се формира матрица са координатама свих чворова које робот мора да посети у току извршавања кретања. Кретање се изводи по сегментима, односно први чвор се узима као почетна позиција, а други као крајња, по достизању те друге позиције она се узима као почетна, а следећа из матрице као крајња и овај процес се понавља док крајња позиција не буде уједно и циљна позиција задата на почетку кретања. Дакле након формирања те матрице проверава се да ли је робот достигао циљ, ако јесте програм се ту завршава, у супротном прелази у функцију за рачунање угла заокретања и растојања које треба да пређе да би достигао следећу тачку. Ови подаци се прослеђују у функцију за одређивање брзине и смера ротације погонског вратила. Уједно се утврђује и колика ће бити вредност енкодера за то растојање које треба робот да пређе. Ако је израчунато да је угао 0 , значи да је оријентација робота добра и да само треба да се обави праволинијско кретање до следеће тачке, ако је угао различит од 0 , прелази се у функцију за скретање. Она служи за утврђивање, на основу израчунатог угла, у ком смеру тачкови треба да се ротирају и за који угао, како би након извршеног скретања, робот могао само праволинијским кретањем да стигне до следеће тачке. У току скретања енкодери читавају вредности за колико се окренуло вратило мотора. Док је та вредност мања од претходно израчунате, вратило мотора се окреће, када та вредност буде једнака или већа претходно израчунатој, процес скретања се завршава. Након успешно обављеног процеса скретања прво се узимају информације са ИЦ сензора. Ако се утврди да је робот паралелан са зидом са његове леве или десне стране, онда има упутство да га прати. Ако није детектован паралелан зид роботу, прелази се у функцију за обраду сензорских информација. Овде се сензорске информације користе као улазни скуп у ВНМ на основу којих се доноси одлука о даљем деловању робота. Након донешене одлуке од стране ВНМ о даљем деловању, битно је утврдити, да ли је претходно одрађено кретање или скретање последица наиласка робота на непредвиђену препреку или је оно последица праћења оптималне путање. У случају да је робот вршио кретање у циљу избегавања препреке, онда се тај положај робота у коме се он тренутно налази узима за стартни положај, док циљ остаје исти и процес се даље наставља као на почетку извршавања програма. Ако је то кретање било остварено због праћења путање, онда прелазимо у део где се проверава излаз који је дала ВНМ мрежа.

Ако је путања чиста, тј. нема непредвиђених препрека, проверава се да ли је достигнута жељена позиција, ако није покреће се функција за кретање која покреће функцију за израчунавање растојања које треба робот да пређе да би достигао жељену позицију. Овде се не обраћа пажња на оријентацију, јер је скретање извршено раније и угао заокретања је сада 0 . Након добијања неопходних података, поново се покреће робот и даље се наставља процес аналоган описаном процесу скретања.

Ако путања није чиста, тј. има непредвиђених препрека, прво се проверава да ли је робот на сигурном растојању до препреке да може да изврши обилажење без колизије са препреком. У случају да овај услов није задовољен, покреће се функција која враћа робота на сигурно растојање до препреке. Сада се поново узимају сензорске информације и користе као улазни скуп у ВНМ која треба, на основу претходног обучавања са јединственим карактеристичним случајевима са којима робот може да се сретне у окружењу, донесе одлуку о ком случају је реч као и одлуку на који начин та препрека да се заобиђе. Излаз из ове мреже је угао заокретања робота који се

⁶ Реч функција коришћена је овде у *matlab* контексту, односно представља одређени део кода који је направљен искључиво за извршавање одређеног задатка - функције (нпр. функција за кретање, функција за граф видљивости итд.)

прослеђује у фунцкију за израчунавање брзине и смера ротације погонског вратила, а затим се наставља процес по претходно описаном начину извршавања.

Овим су покривени сви случајеви који могу да се десе у току кретања интелигентног мобилног робота по моделу технолошког окружења. Горе описани процес се извршава у зависности од стања окружења и понавља док робот не прође кроз све дефинисане чворове и стигне до циља. Тада се програм завршава и на ред долази провера извршеног кретања као и израчунавање грешке (детално описано у поглављу 8).

6. Функционални захтеви и параметри пројектовања у домену пројектовања интелигентног понашања мобилног робота

Аксиоматско пројектовање је теорија пројектовања коју је креирао и популаризовао професор Нам Су (Nam Suh). У ствари то је више општи оквир за пројектовање него теорија пројектовања. То значи да се може применити на све активности пројектовања. Састоји се од два аксиома. Први је аксиом независности, а други је информациони аксиом. Дobar пројекат би требало да задовољи оба аксиома. Пројектовање је преплитање између онога „шта желимо да остваримо“ и онога „како то постижемо“. Пројектант покушава да оствари оно шта жели кроз одговарајуће интерактивно преплитање између обе стране.

Аксиом 1: Аксиом независности

Код прихватљивог пројекта, ПП и ФЗ су у таквим релацијама да одређени ПП може да се подеси да задовољи одговарајући ФЗ без утицаја на остале функционалне захтеве.

Аксиом 2: Информациони аксиом

Најбољи пројекат је функционално неспрегнут пројекат који има минимални информациони садржај.

Када пројектујемо комплексни систем овај аксиом нам говори да ПП треба да буду дефинисани да независно задовољавају одговарајуће ФЗ. Другим речима, ФЗ из функционалног домена би требало да буду независно задовољени параметрима пројектовања из физичког домена. У противном, пројекат није добар. Када се више пројектних решења нађе помоћу аксиома 1 најбоље решење се бира на основу аксиома 2. Најбоље решење има минимални информациони садржај који се обично квантификује вероватноћом успеха [18].

Када је више функционалних захтева дефинисано, сваки ПП треба да задовољи одговарајући ФЗ. Овај однос се може изразити матрицом пројектовања. Коришћењем векторског приказа ФЗ и ПП однос се изражава следећом једначином пројектовања:

$$\text{ФЗ} = \text{МА} \cdot \text{ПП} \quad (6.1)$$

Матрица МА се назива матрицом пројектовања. Карактеристике матрице МА одређују да ли је аксиом независности задовољен. Ако је матрица пројектовања дијагонална онда је у питању неспрегнути пројекат. Када је матрица пројектовања троугаона пројекат је распрегнут. Распрегнути пројекат задовољава аксиом независности уколико је редослед пројектовања коректан. Када матрица пројектовања није ни дијагонална ни троугаона, пројекат је спрегнут. Код спрегнутог пројекта не постоји редослед параметара пројектовања којим се функционални захтеви могу задовољити независно. Према томе, неспрегнут и распрегнут пројекат задовољавају аксиом независности, а спрегнут пројекат га не задовољава. Уколико је пројекат спрегнут мора се пронаћи неспрегнут или распрегнут пројекат кроз нови избор параметра пројектовања. Над матрицама пројектовања дозвољене су операције множења и сабирања, док друге операције као трансформација координата нису дозвољене [18].

Процес пројектовања система почиње успостављањем ФЗ у функционалном домену, с циљем стварања физичког објекта или процеса који их задовољава. Након усвајања функционалних захтева следи дефинисање параметара пројекта помоћу којих могу да се остваре задати функционални захтеви.

Табела 6.1: Приказ функционалних захтева (ФЗ) при пројектовању интелигентног понашања мобилног робота за транспорт материјала

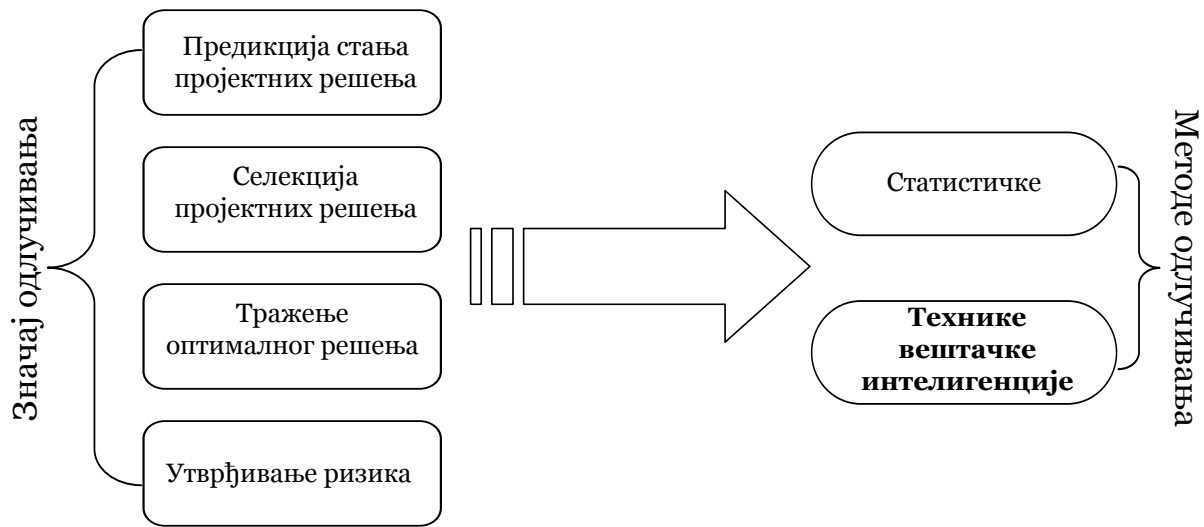
| Вектор функционалних захтева (ФЗ) | |
|-----------------------------------|---|
| ФЗ | Извршавање задатих операција – транспорт материјала |
| ФЗ ₁ | Положај мобилног робота у току кретања |
| ФЗ ₁₁ | Позиција робота |
| ФЗ ₁₁₁ | Помераји робота |
| ФЗ ₁₁₂ | Прецизност при кретању |
| ФЗ ₁₂ | Оријентација робота |
| ФЗ ₁₂₁ | Помераји робота |
| ФЗ ₁₂₂ | Прецизност при заокретању |
| ФЗ ₂ | Навигација |
| ФЗ ₂₁ | План кретања |
| ФЗ ₂₁₁ | Модел технолошког окружења |
| ФЗ ₂₁₁₁ | Репрезентовање модела |
| ФЗ ₂₁₂ | Формирање могућих трајекторија |
| ФЗ ₂₁₃ | Критеријум за оптималну путању |
| ФЗ ₂₁₃₁ | Оптимална путања |
| ФЗ ₂₂ | Реаговање на непредвиђене догађаје |
| ФЗ ₂₂₁ | Уочавање препрека |
| ФЗ ₂₂₁₁ | ВНМ за детекцију препрека |
| ФЗ ₂₂₁₂ | Одређивање растојања до препрека |
| ФЗ ₂₂₂ | Заобилажење препрека |
| ФЗ ₂₂₂₁ | Класификација карактеристичних препрека |
| ФЗ ₂₂₂₂ | Начин заобилажења препрека |
| ФЗ ₂₂₂₂₁ | Препрека са леве стране |
| ФЗ ₂₂₂₂₁₁ | Угао заокретања |
| ФЗ ₂₂₂₂₂ | Препрека са десне стране |
| ФЗ ₂₂₂₂₂₁ | Угао заокретања |
| ФЗ ₂₂₃ | Успешност избегавања препрека |

Табела 6.2: Приказ параметара пројектовања (ПП) при пројектовању интелигентног понашања мобилног робота за транспорт материјала

| Вектор параметара пројектовања (ПП) | |
|-------------------------------------|--|
| ПП | Кретање интелигентног мобилног робота |
| ПП ₁ | Модул за кретање мобилног робота на основу пређеног пута |
| ПП ₁₁ | Очитавања са енкодера |
| ПП ₁₁₁ | Број послатих импулса |
| ПП ₁₁₂ | Резолуција енкодера |
| ПП ₁₂ | Очитавања са енкодера |
| ПП ₁₂₁ | Број послатих импулса |
| ПП ₁₁₂ | Резолуција енкодера |
| ПП ₂ | Модул за навигацију и планирање путање |
| ПП ₂₁ | Алгоритам за планирање путање |
| ПП ₂₁₁ | Дискретизација континуираног конфигурацијског простора |
| ПП ₂₁₁₁ | Повезани граф |
| ПП ₂₁₂ | Граф видљивости |
| ПП ₂₁₃ | Најкраће растојање од старта до циља |
| ПП ₂₁₃₁ | Алгоритам за претрагу графа – Дајкстрин алгор. |
| ПП ₂₂ | Модул за избегавање препрека |
| ПП ₂₂₁ | Инфрацрвени сензори |
| ПП ₂₂₁₁ | Параметри за обучавање – инфо. са ИЦ сензора |
| ПП ₂₂₁₂ | ВНМ за одређ. растојања и детекцију препрека |
| ПП ₂₂₂ | Модул за заобилажење препрека |
| ПП ₂₂₂₁ | ВНМ за одлуч. о начину заобилажења препрека |
| ПП ₂₂₂₂ | ВНМ за препознавање препрека |
| ПП ₂₂₂₂₁ | Скретање у десно – излаз из ВНМ |
| ПП ₂₂₂₂₁₁ | ВНМ генерише угао |
| ПП ₂₂₂₂₂ | Скретање у лево – излаз из ВНМ |
| ПП ₂₂₂₂₂₁ | ВНМ генерише угао |
| ПП ₂₂₃ | Матрица одлучивања |

7. Дрво и матрица одлучивања

Када постоји много опција-решења неког проблема која могу да се реализују, процес одлучивања представља тражење вероватноће за свако од њих, са крајњим циљем да се одреди једно решење са вероватноћом $p=1$ (100%) тако да су све остале опције са вероватноћом 0. Одлучивање је процес промишљања и разумевања проблема, уз пресуђивање (према теорији одлучивања - раздвајање добрих акција од лоших) које је коначни исход процеса[7].



Процес одлучивања одвија се у оквиру следећих корака:

- Утврђивање више од једне опције могућег правца будуће акције или стратегије. Ови правци се идентификују коришћењем система за одлучивање (у овом случају примена техника вештачке интелигенције).
- Предикција излаза сваког од утврђених праваца акције или стратегије.
- Одређивање вероватноће реализације утврђених праваца акције или стратегије.
- Одлучивање на основу критеријумске селекције решења са највећом вредношћу вероватноће остваривања.

Класификација процеса одлучивања обухвата утврђивање услова под којима до одлучивања долази и разликује се:

1. Одлучивање под тачно одређеним условима – без ризика при остваривању специфичног излаза. Оствариво је ако се примењује подсистем за предикцију са стриктним ланцем догађаја.

2. Одлучивање са ризиком при остваривању специфичних излаза за које је вероватноћа реализације позната. Оствариво је ако се примењује подсистем за предикцију са пробабилистичким ланцем догађаја.

3. Одлучивање са ризиком при остваривању специфичних излаза за које је вероватноћа реализације непозната. Оствариво је ако се примењује подсистем за предикцију са емпиријским ланцем догађаја. Коришћено у овом раду, применом техника вештачке интелигенције.

7.1 Одлучивање са ризиком – матрица одлучивања

Одлучивање са ризиком се формализује увођењем матрице одлучивања која поред активности - A_j током процеса одлучивања (акције или стратегије у пројектовању решења задатог проблема) обухвата и веома важан утицај догађаја у окружењу - Y_i у коме селековано пројектно решење треба да егзистира. Она одређује све комбинације понашања, тј. акције у датом окружењу, са непознатом вероватноћом p_i остваривања догађаја. У овом случају се мора увести машинско учење.

Табела 7.1: Матрица одлучивања

| | P_1 | P_2 | P_3 | P_4 | P_5 | P_6 | P_7 | P_8 | P_9 | P_{10} |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 |
| Y1 | 83.34 | 3.33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Y2 | 13.33 | 93.34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Y3 | 0 | 3.33 | 96.67 | 0 | 3.33 | 0 | 0 | 0 | 0 | 3.33 |
| Y4 | 0 | 0 | 0 | 80.01 | 0 | 0 | 3.33 | 0 | 0 | 0 |
| Y5 | 0 | 0 | 0 | 0 | 96.67 | 19.99 | 3.33 | 3.33 | 0 | 0 |
| Y6 | 3.33 | 0 | 0 | 0 | 0 | 80.01 | 0 | 0 | 0 | 0 |
| Y7 | 0 | 0 | 0 | 19.99 | 0 | 0 | 90.01 | 16.66 | 3.33 | 0 |
| Y8 | 0 | 0 | 3.33 | 0 | 0 | 0 | 3.33 | 80.01 | 0 | 0 |
| Y9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 76.68 | 23.33 |
| Y10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19.99 | 73.34 |

Колоне представљају могуће реакције, док врсте представљају број могућих подстрек. Када су ове вредности познате може се описати понашање посматраног агента матрицом вероватноћа понашања (Матрица одлучивања). Може се рећи да њени елементи карактеришу вероватноћу датог агента да у одређеном стању одговори реакцијом A_j на подстрек Y_i . Ако је скуп Y_1, Y_2, \dots, Y_i потпуни скуп свих могућих реакција и ако на улазни подстрек агент увек одговара неком излазном реакцијом, тада је сума вероватноћа свих реакција на сваки подстрек једнака јединици.

$$\sum p_i = 1$$

На основу матрице одлучивања лако је видети за које случајеве ВНМ даје најбоље резултате. Овде су то случајеви 3 и 5 са успешношћу од 96,67%, док су најлошији резултати за случајеве 9 и 10, 76,68% и 73,34%. Међутим, ово је последица мале разлике у вредностима сензорских читавања за случајеве 9 и 10. Јасно се види да највећи проценат неуспелих покушаја за случај 9 одлази на случај 10 и обрнуто. Мора се нагласити да ово не мора да значи да је ВНМ лоше обучена, већ при овако малим разликама између случаја, велику улогу у доношењу правилне одлуке игра и шум сензора, о коме је било више речи у поглављу 5.

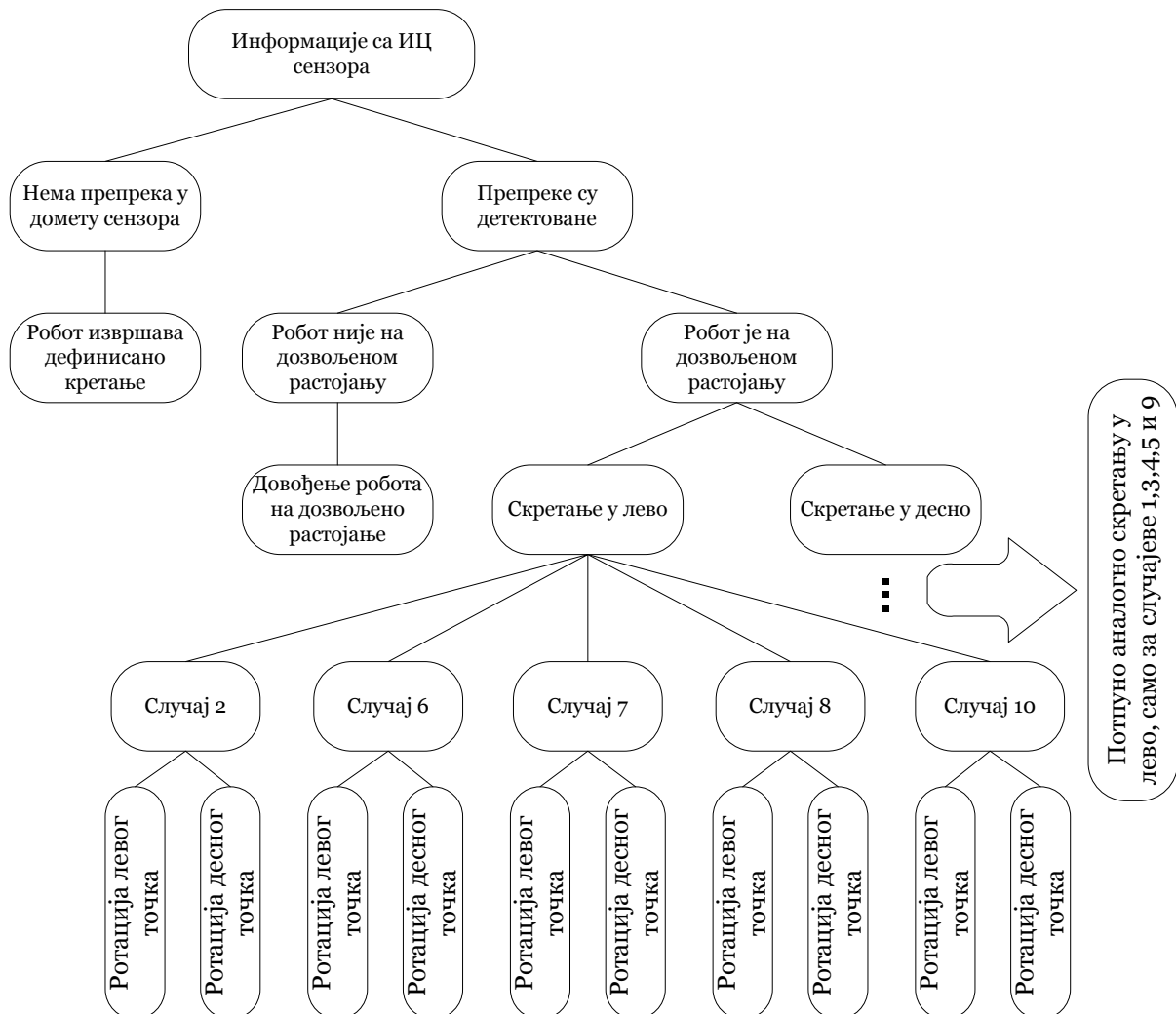
7.2 Дрво одлучивања

„Дрво одлучивања” представља хијерархијску структуру података са имплементираним стратегијом за доношење одлука по принципу „раздвоји-и-изабери”. То је ефикасан непараметарски метод који се користи за класификацију и

регресију. Овај хијерархијски модел за супервизорско машинско учење чине елементи који учествују у доношењу одлука на основу имплементирања тест-функције $f_m(x)$ са дискретним излазним вредностима. [7]

Хијерархијска структура дрвета одлучивања је организована тако да поседује:

- Интерне чворове одлучивања који могу да буду :
 - Униформни: користе један атрибут, x_i
 - Нумерички x_i : Бинарно раздвајање : $f_m(x): x_i > W_{m0}$
 - Дискретни x_i : n-могућности раздвајања за n могућих вредности
 - Мултиваријантни: Користе све атрибуте, x
- Терминале-излазне чворове који врше :
 - Класификацију: лабелишу класе, или пропорције података
 - Регресију: нумеричку; r усредњавање (средња квадратна разлика), локално „фитовање” грешке



Слика 7.1: Дрво одлучивања за информације добијене са сензора

Случајеви 1-10 који се спомињу овде представљају карактеристичне случајеве на које робот може да наиђе у току кретања и детаљно су описани у поглављу 5.3.

8. Тестирање и верификација тачности и перформанси развијених решења интелигентног понашања мобилног робота у радном простору дефинисаном у оквиру лабораторијског модела технолошког окружења

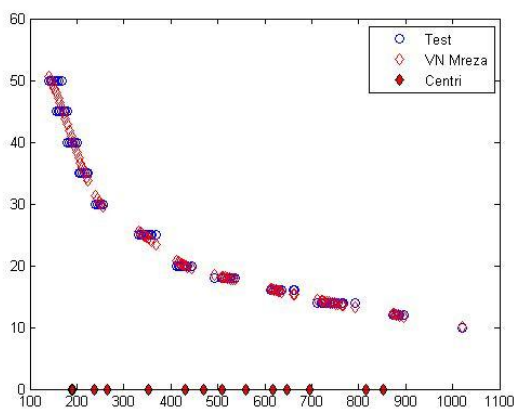
За управљање интелигентним мобилним роботом, као и за тренирање мрежа коришћен је искључиво софтверски пакет фирме *MathWorks, Matlab*[®]. Сви кодови су специјално развијани за *Khepera II*, док су основне функције обезбеђене *kmatrix toolbox-om*. За обучавање мрежа коришћене су интегрисане matlab-ове функције. Неки од основних кодова могу се наћи у поглављу 11.

8.1 Резултати обучавања МРФ

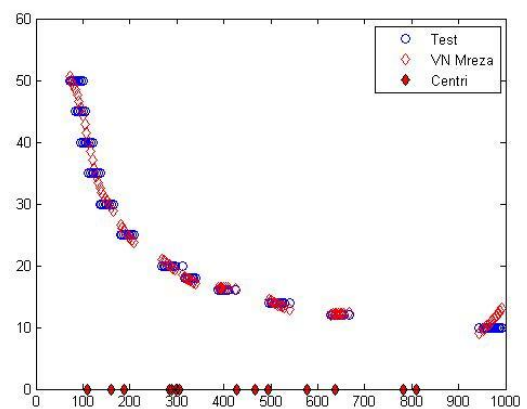
При обучавању захтевано је да грешка буде мања од 10^{-3} што је у сваком случају испуњено. За сваки сензор је урађено по 10 обучавања на основу којих су добијене вредности које се налазе у табелама 8.1 и 8.2. Свака мрежа поседовала је 20 неурона у скривеном слоју и имала 240 улазних вредности и 12 излазних. У табелама 8.1 и 8.2 као и на сликама 8.1 и 8.2 приказани су добијени резултати.

Табела 8.1: Приказ параметара добијених при обучавању МРФ мреже за препознавање растојања помоћу ИЦ сензора (за прва 4 сензора)

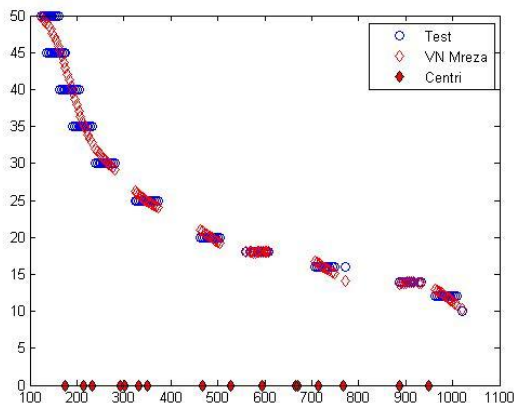
| | 1 | 2 | 3 | 4 |
|----------------|----------|----------|----------|---------|
| Std_rmse_test | 0.2335 | 0.6403 | 0.0936 | 0.3941 |
| Std_rmse_train | 0.1520 | 0.6730 | 0.0656 | 0.3847 |
| m_rmse_test | 0.9518 | 1.8608 | 1.6757 | 1.6199 |
| m_rmse_train | 1.0046 | 1.7974 | 1.6899 | 1.5401 |
| min_rmse_test | 0.8256 | 1.4649 | 1.5406 | 1.3074 |
| min_rmse_train | 0.8014 | 1.3819 | 1.6120 | 1.2548 |
| m_iter_no | 122.3000 | 165.8000 | 158.1000 | 250.900 |
| m_time [s] | 19.1252 | 21.5541 | 20.2208 | 22.9571 |



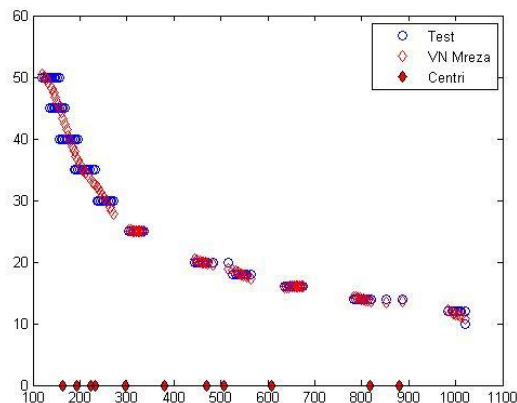
1)



2)



3)

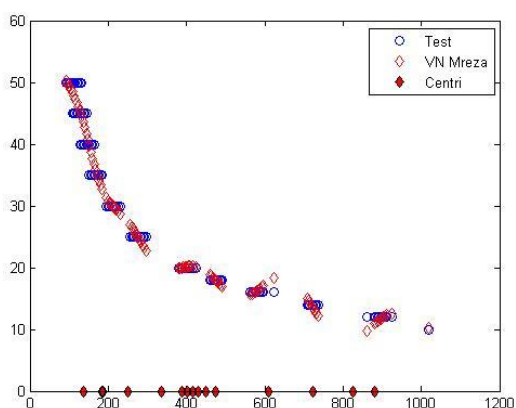


4)

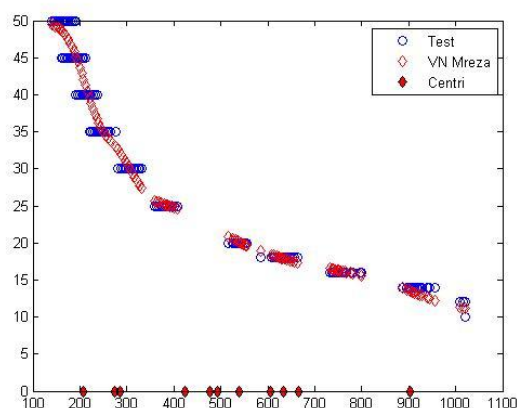
Слика 8.1: Графички приказ резултата обучавања за обучавајући скуп као и за тест скуп. Сlike 1),2),3),4) одговарају сензорима 1,2,3,4, респективно. X оса представља очитане вредности са сензора, а Y растојање у [mm]

Табела 8.2: Приказ параметара добијених при обучавању МРФ мреже за препознавање растојања помоћу ИЦ сензора (за друга 4 сензора)

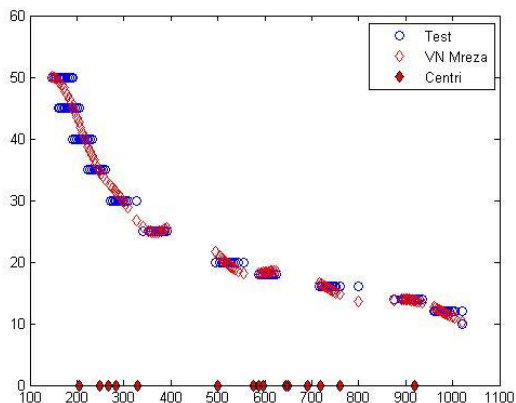
| | 5 | 6 | 7 | 8 |
|----------------|----------|----------|----------|----------|
| Std_rmse_test | 0.2942 | 0.7431 | 0.1889 | 0.5206 |
| Std_rmse_train | 0.2616 | 0.7586 | 0.1711 | 0.5333 |
| m_rmse_test | 1.8956 | 1.9791 | 1.7046 | 1.8490 |
| m_rmse_train | 1.8343 | 1.9181 | 1.6921 | 1.8373 |
| min_rmse_test | 1.5176 | 1.4600 | 1.5732 | 1.3855 |
| min_rmse_train | 1.4892 | 1.3798 | 1.5803 | 1.3690 |
| m_iter_no | 163.0000 | 304.8000 | 168.0000 | 221.5000 |
| m_time [s] | 17.3348 | 32.8616 | 21.3550 | 25.5686 |



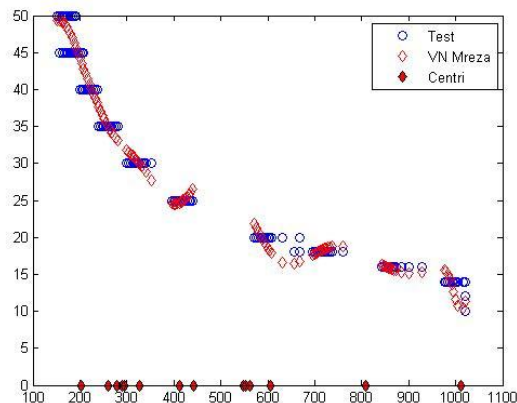
5)



6)



7)



8)

Слика 8.2: Графички приказ резултата обучавања за обучавајући скуп као и за тест скуп. Сlike 5),6),7),8) одговарају сензорима 5,6,7,8, респективно. X оса представља очитане вредности са сензора, а Y растојање у [mm]

Где су :

Std_rmse_test - Стандардна девијација за тест скуп

Std_rmse_train - Стандардна девијација за обучавајући скуп

m_rmse_train – Средња вредност корена средње квадратне грешке за тест скуп

m_rmse_test – Средња вредност корена средње квадратне грешке за обучавајући скуп

min_rmse_train – Минимална вредност корена средње квадратне грешке за тест скуп

min_rmse_test – Минимална вредност корена средње квадратне грешке за обучавајући скуп

m_iter_no – Просечан број итерација за обучавање мреже

m_time [s] – Просечно време трајања обучавања мреже, изражено у секундама

Након извршеног обучавања мреже су тестиране у реалним условима за растојања од 12, 20 и 40 [mm]. Свако мерење поновљено је 100 пута. Добијени резултати дати су у табелама 8.3, 8.4 и 8.5.

Табела 8.3: Приказ средње вредности средње квадратне грешке за излаз из ВНМ где се препрека налази на 40 [mm]

| Сензор | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| m_rmse | 6.4290 | 5.1402 | 4.7414 | 5.8069 | 4.0873 | 5.0295 | 5.0327 | 5.0097 |

Табела 8.4: Приказ средње вредности средње квадратне грешке за излаз из ВНМ где се препрека налази на 20 [mm]

| Сензор | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| m_rmse | 1.2021 | 0.6702 | 1.0142 | 0.5016 | 0.9543 | 0.3938 | 1.8845 | 1.4544 |

Табела 8.5: Приказ средње вредности средње квадратне грешке за излаз из ВНМ где се препрека налази на 12 [mm]

| Сензор | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| m_rmse | 0.1707 | 0.1712 | 0.1606 | 0.1254 | 0.0955 | 0.1540 | 0.1878 | 0.1787 |

m_rmse – Средња вредност средње квадратне грешке излаза из ВНМ

Табела 8.6: m_rmse процентуално приказан за свих осам сензора и сва три растојања која су коришћена у овом експерименту

| Ред. бр. сензора | 12[mm] | 20[mm] | 40[mm] |
|------------------|--------|--------|--------|
| 1 | 1.423 | 6.011 | 16.073 |
| 2 | 1.427 | 3.351 | 12.851 |
| 3 | 1.338 | 5.071 | 11.854 |
| 4 | 1.045 | 2.508 | 14.517 |
| 5 | 0.796 | 4.772 | 10.218 |
| 6 | 1.283 | 1.969 | 12.574 |
| 7 | 1.565 | 9.423 | 12.582 |
| 8 | 1.489 | 7.282 | 12.524 |

Из ових резултата мерења види се да грешка при мерењу растојања од 12 [mm] до препреке варира између 0.796% и 1.56%. Што је јако мала грешка. При мерењу растојања од 20 [mm] до препреке, грешка варира између 1.969% и 9.423%. С обзиром да је највећа грешка забележена код сензора 7, који се налази на задњој страни робота, можемо је занемарити, јер су мерења првих 6 сензора круцијална за доношење правилних одлука, док сензори 7 и 8 имају мању улогу. Грешка на првих 6 сензора не прелази 6,011%, што се може сматрати јако добрим резултатом. Резултати за растојање од 40 [mm] варирају између 10.218% и 16.073%. Грешка прелази 10% и ограниченост сензора долази до изражаја. Из овог разлога су и обучаване две мреже за избегавање препрека, за једну су коришћена мерења са растојања од 20 [mm], а за другу мерења са растојања од 40 [mm]. Резултати обучавања ових мрежа приказани су у табелама 8.7, 8.8, 8.9, 8.10, 8.11, 8.12, 8.13 и 8.14.

8.2 Резултати обучавања мрежа за препознавање препрека

У процесу обучавања коришћене су *feed – forward backpropagation* неуронске мреже. Преносна функција скривеног слоја је била *tansig*, док је излазни слој имао линеарну преносну функцију *purelin*. Примењен је Левенберг–Маркеов алгоритам учења (описан у поглављу 5.2.1). Постоје два услова за заустављање процеса обучавања. Први је граница од 200 итерација, а други када се градијент смањи испод 10^{-10} .

У табелама су дате вредности броја неурона у скривеном слоју, корена средње квадратне грешке за обучавајући скуп као и за тест скуп, затим вредност параметра учења, броја итерација, протеклог времена и броја итерација по секунди. У табелама где су вредности мрежа са 2 или 3 скривена слоја, прве 2 или 3 колоне означавају бројеве неурона у 1., 2. и 3. скривеном слоју у зависности од мреже.

Табела 8.7: Приказ резултата тренирања веитачке неуронске мреже са једним скривеним слојем (растојање до препрека 20 [mm])

| Бр. н. | rmse_train | rmse_test | Пар. учења | Бр. итерација | Време [s] | Итер./ [s] |
|--------|------------|-----------|------------|---------------|-----------|------------|
| 5 | 1.2097 | 1.4279 | 0.4 | 200 | 5 | 40.00 |
| 5 | 1.0290 | 1.2811 | 0.1 | 139 | 4 | 34.75 |
| 5 | 0.5828 | 1.0177 | 0.02 | 197 | 5 | 39.40 |
| 15 | 0.1029 | 0.6732 | 0.4 | 167 | 11 | 15.18 |
| 15 | 0.1998 | 0.7442 | 0.1 | 200 | 13 | 15.38 |

| | | | | | | |
|----|-------------------------|--------|------|-----|-----|------|
| 15 | 0.3547 | 1.0027 | 0.02 | 148 | 10 | 14.8 |
| 30 | $5.6239 \cdot 10^{-9}$ | 0.5693 | 0.4 | 150 | 30 | 5 |
| 30 | $6.3590 \cdot 10^{-10}$ | 0.3948 | 0.1 | 24 | 5 | 4.8 |
| 30 | $2.6585 \cdot 10^{-10}$ | 0.1168 | 0.02 | 42 | 12 | 3.5 |
| 50 | $4.0739 \cdot 10^{-10}$ | 0.1866 | 0.4 | 200 | 112 | 1.79 |
| 50 | $2.3189 \cdot 10^{-9}$ | 0.1751 | 0.1 | 26 | 14 | 1.86 |
| 50 | $5.5292 \cdot 10^{-9}$ | 0.2758 | 0.02 | 200 | 109 | 1.83 |

Табела 8.8: Приказ резултата тренирања вештачке неуронске мреже са два скривена слоја (растојање до препрека 20 [mm])

| Бр. н. с. с. 1 | Бр. н. с. с. 2 | rmse_train | rmse_test | Пар. учења | Бр. Итер. | Време [s] | Итер./ [s] |
|-------------------|-------------------|------------------------|-----------|------------|--------------|--------------|---------------|
| 10 | 5 | 0.4249 | 0.8022 | 0.4 | 200 | 8 | 25.00 |
| 10 | 5 | 0.9559 | 1.5042 | 0.1 | 200 | 8 | 25.00 |
| 10 | 5 | 1.0072 | 1.3737 | 0.02 | 200 | 8 | 25.00 |
| 10 | 10 | 0.5269 | 1.0640 | 0.4 | 200 | 14 | 14.26 |
| 10 | 10 | 0.3614 | 0.6166 | 0.1 | 26 | 2 | 13.00 |
| 10 | 10 | 1.1797 | 1.3854 | 0.02 | 200 | 15 | 13.33 |
| 10 | 20 | 0.0620 | 0.4046 | 0.4 | 200 | 33 | 6.66 |
| 10 | 20 | 0.2657 | 0.4126 | 0.1 | 85 | 14 | 6.07 |
| 10 | 20 | 0.1960 | 0.2414 | 0.02 | 112 | 18 | 6.22 |
| 20 | 10 | $3.3682 \cdot 10^{-7}$ | 0.6971 | 0.4 | 200 | 38 | 5.26 |
| 20 | 10 | 0.0864 | 0.6568 | 0.1 | 200 | 29 | 6.89 |
| 20 | 10 | $7.2357 \cdot 10^{-7}$ | 0.3277 | 0.02 | 200 | 40 | 5.00 |
| 20 | 20 | $5.6817 \cdot 10^{-7}$ | 0.4045 | 0.4 | 200 | 74 | 2.70 |
| 20 | 20 | $1.6154 \cdot 10^{-6}$ | 0.6864 | 0.1 | 200 | 73 | 2.74 |
| 20 | 20 | $1.1286 \cdot 10^{-5}$ | 0.3876 | 0.02 | 200 | 73 | 2.74 |
| 30 | 30 | 0.0877 | 0.1916 | 0.4 | 84 | 115 | 0.73 |
| 30 | 30 | $1.5902 \cdot 10^{-6}$ | 0.1931 | 0.1 | 102 | 141 | 0.72 |
| 30 | 30 | 0.0876 | 0.1036 | 0.02 | 155 | 213 | 0.73 |

Табела 8.9: Приказ резултата тренирања вештачке неуронске мреже са три скривена слоја (растојање до препрека 20 [mm])

| Бр. н. с. с. 1. | Бр. н. с. с. 2. | Бр. н. с. с. 3. | rmse_train | rmse_test | Пар. учења | Бр. Итер. | Време [s] | Итер./ [s] |
|--------------------|--------------------|--------------------|------------------------|-----------|---------------|--------------|--------------|---------------|
| 4 | 8 | 4 | 1.3022 | 1.5486 | 0.2 | 42 | 2 | 21.00 |
| 4 | 8 | 4 | 0.4506 | 0.5448 | 0.02 | 26 | 1 | 26.00 |
| 10 | 5 | 5 | 1.1371 | 1.4379 | 0.2 | 200 | 10 | 20.00 |
| 10 | 5 | 5 | 0.3800 | 0.5289 | 0.02 | 200 | 9 | 22.22 |
| 6 | 6 | 12 | 1.0321 | 1.2171 | 0.2 | 200 | 13 | 15.38 |
| 6 | 6 | 12 | 1.0160 | 1.4918 | 0.02 | 200 | 14 | 14.28 |
| 10 | 10 | 10 | 1.0376 | 1.2257 | 0.2 | 25 | 3 | 8.33 |
| 10 | 10 | 10 | 0.3551 | 0.4468 | 0.02 | 200 | 21 | 9.52 |
| 12 | 16 | 12 | 0.0877 | 0.6088 | 0.2 | 124 | 29 | 4.28 |
| 12 | 16 | 12 | 0.1080 | 0.6006 | 0.02 | 200 | 46 | 4.35 |
| 19 | 13 | 13 | $1.8729 \cdot 10^{-8}$ | 0.5218 | 0.2 | 200 | 63 | 3.17 |
| 19 | 13 | 13 | $1.2749 \cdot 10^{-5}$ | 0.5100 | 0.02 | 200 | 66 | 3.03 |
| 17 | 11 | 23 | $1.1430 \cdot 10^{-7}$ | 0.6888 | 0.2 | 200 | 86 | 2.33 |
| 17 | 11 | 23 | $2.0120 \cdot 10^{-7}$ | 0.2615 | 0.02 | 200 | 86 | 2.33 |

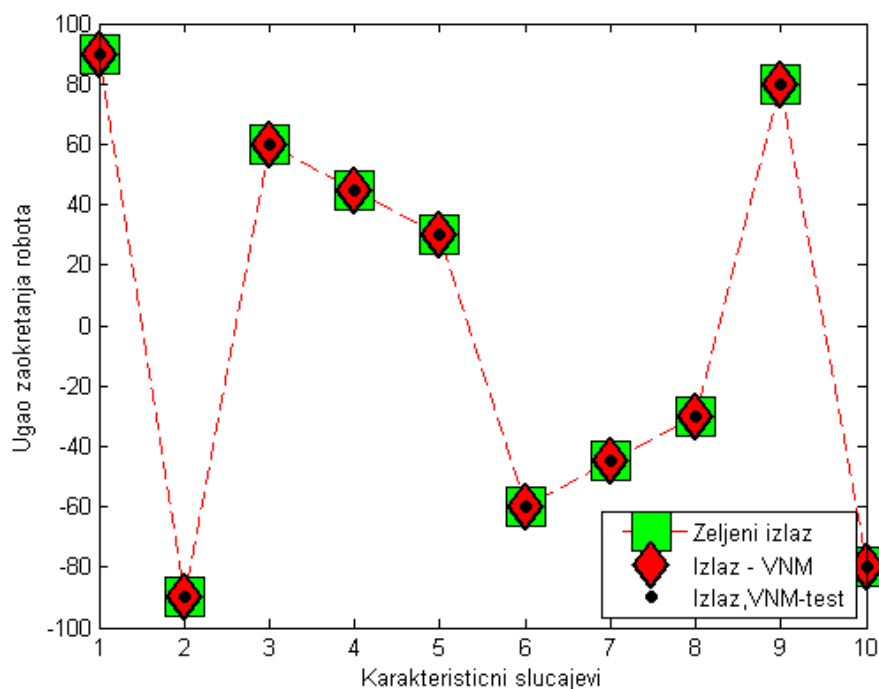
| | | | | | | | | |
|----|----|----|------------------------|--------|------|-----|-----|------|
| 20 | 20 | 20 | $4.0176 \cdot 10^{-9}$ | 0.1641 | 0.2 | 36 | 29 | 1.24 |
| 20 | 20 | 20 | $1.1179 \cdot 10^{-6}$ | 0.4651 | 0.02 | 200 | 179 | 1.12 |
| 21 | 27 | 21 | $3.0755 \cdot 10^{-6}$ | 0.6752 | 0.2 | 86 | 133 | 0.65 |
| 21 | 27 | 21 | $9.5941 \cdot 10^{-8}$ | 0.4514 | 0.02 | 100 | 150 | 0.67 |

Табеле 8.7, 8.8, 8.9 и 8.10 приказују вредности добијене при обучавању прве мреже, тј. мреже која треба да препозна препреке са 20 [mm] удаљености. На основу добијених вредности закључује се да најбоље резултате даје мрежа са једним скривеним слојем и 30 неурона у њему. Такође су и мреже са 2 и 3 скривена слоја дале задовољавајуће резултате. Међутим узета је најбоља мрежа и покушано је да се резултат побољша варирањем параметра учења. Добијени резултати приказани су у табели 8.10.

Табела 8.10: Варирање параметра учења за мрежу са најбољим резултатима (растојање до препрека 20 [mm])

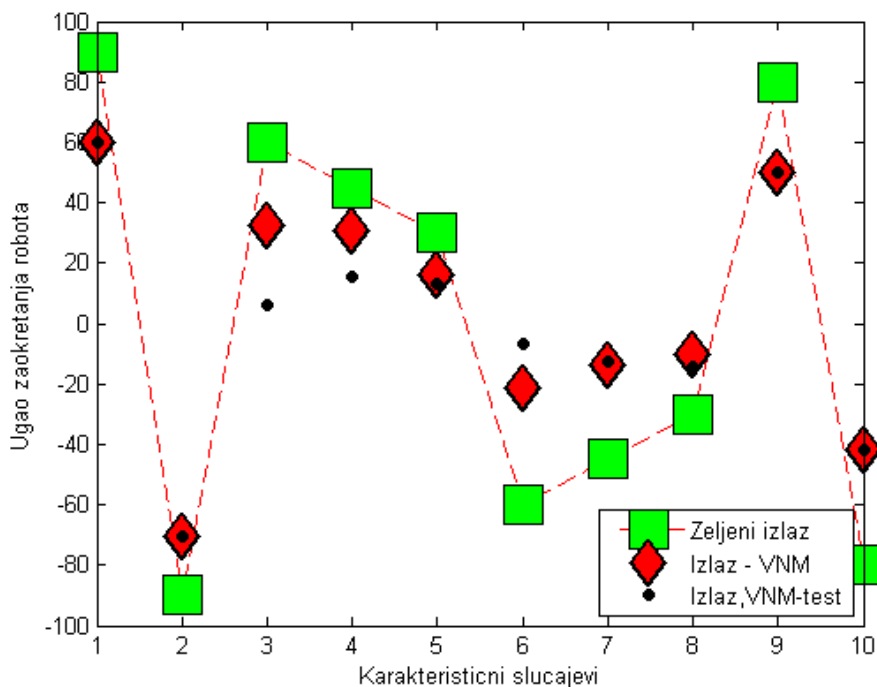
| Бр. неурона | rmse_train | rmse_test | Пар. учења | Бр. итерација | Vreme [s] | Итер./ [s] |
|-------------|-------------------------|-----------|------------|---------------|-----------|------------|
| 30 | $1.6927 \cdot 10^{-9}$ | 0.2764 | 0.01 | 27 | 7 | 3.86 |
| 30 | $4.8688 \cdot 10^{-10}$ | 0.2243 | 0.001 | 16 | 3 | 5.33 |
| 30 | $3.7592 \cdot 10^{-12}$ | 0.2181 | 0.0001 | 76 | 15 | 5.07 |
| 30 | $1.1770 \cdot 10^{-11}$ | 0.1934 | 0.9 | 41 | 8 | 5.13 |
| 30 | $4.7333 \cdot 10^{-11}$ | 0.3730 | 0.2 | 39 | 8 | 4.875 |

Из табеле 8.10 видимо да су за све параметре учења добијени задовољавајући резултати. Међутим, најбољи резултати су добијени када је вредност параметра учења износила 0.0001 иако су препоручене вредности у многим литературама од 0.2-0.4. У само 76 итерација у трајању од 15 секунди, односно 5.07 итерације по секунди, обучена је мрежа чији корен средње квадратне грешке износи $3.7592 \cdot 10^{-12}$ за обучавајући, односно 0.2181 за тест скуп. Графички приказ тестирања мреже дат је на слици 8.3.



Слика 8.3: Приказ излаза неуронске мреже након завршеног процеса обучавања

На слици 8.3, X оса представља десет карактеристичних случајева који су описани у поглављу 5.3, док Y оса представља излаз из мреже за одређени случај, тј. угао заокретања робота ради избегавања препреке. Зелени квадрати представљају жељене вредности излаза, црвени ромбови представљају излаз мреже за обучавајући скуп, док црне тачке представљају излаз из мреже за тест скуп. Ради компарације, дата је слика 8.4 на којој је приказан график излаза из мреже за неуспешно обучавање.



Слика 8.4: Приказ једног неуспешног обучавања VNM

Табеле 8.11, 8.12, 8.13 и 8.14 приказују вредности добијене при обучавању друге мреже, тј. мреже која треба да препозна препреке са 40 [mm] удаљености. Због ограничености ИЦ сензора, није имало смисла обучавати мрежу за случајеве 3 и 6, односно када робот наилази на препреку која се налази са леве стране под углом од 30°, и за препреку која се налази са десне стране под углом од исто 30°. Елиминисањем ова два случаја, остало их је 8 за које ће мрежа бити обучавана.

Табела 8.11: Приказ резултата тренирања вештачке неуронске мреже са једним скривеним слојем (растојање до препрека 40 [mm])

| Бр. неурона | rmse_train | rmse_test | Пар. учења | Бр. итерација | Време [s] | Итер./ [s] |
|-------------|------------------------|-----------|------------|---------------|-----------|------------|
| 5 | 1.9830 | 2.2812 | 0.4 | 200 | 5 | 40.00 |
| 5 | 1.6546 | 1.9740 | 0.1 | 113 | 3 | 37.67 |
| 5 | 1.8236 | 2.4546 | 0.02 | 200 | 5 | 40.00 |
| 15 | 1.7728 | 2.0196 | 0.4 | 83 | 5 | 16.60 |
| 15 | 1.4757 | 3.0781 | 0.1 | 200 | 11 | 18.18 |
| 15 | 0.7740 | 1.7533 | 0.02 | 171 | 9 | 19.00 |
| 30 | 1.7307 | 1.8638 | 0.4 | 112 | 19 | 5.89 |
| 30 | 0.3565 | 2.8274 | 0.1 | 200 | 32 | 6.25 |
| 30 | 0.1856 | 2.9473 | 0.02 | 89 | 15 | 5.93 |
| 50 | $2.0733 \cdot 10^{-7}$ | 1.5309 | 0.4 | 46 | 19 | 2.42 |
| 50 | 0.1474 | 2.3491 | 0.1 | 200 | 92 | 2.17 |
| 50 | 1.3968 | 1.8892 | 0.02 | 23 | 13 | 1.77 |

Табела 8.12: Приказ резултата тренирања вештачке неуронске мреже са два скривена слоја (растојање до препрека 40 [mm])

| Бр. н. с. с. 1 | Бр. н. с. с. 2 | rmse_train | rmse_test | Пар. учења | Бр. итерација | Време [s] | Итер. / [s] |
|-------------------|-------------------|------------|-----------|---------------|------------------|--------------|----------------|
| 10 | 5 | 1.5479 | 2.7581 | 0.4 | 200 | 7 | 28.57 |
| 10 | 5 | 1.9105 | 2.2229 | 0.1 | 200 | 7 | 28.57 |
| 10 | 5 | 1.7113 | 2.4929 | 0.02 | 200 | 8 | 25.00 |
| 10 | 10 | 2.2105 | 2.4677 | 0.4 | 200 | 12 | 16.67 |
| 10 | 10 | 1.1064 | 1.2727 | 0.1 | 200 | 12 | 16.67 |
| 10 | 10 | 1.8757 | 2.0188 | 0.02 | 19 | 2 | 9.50 |
| 10 | 20 | 1.4029 | 1.5090 | 0.4 | 39 | 6 | 6.50 |
| 10 | 20 | 0.9963 | 1.5656 | 0.1 | 200 | 27 | 7.41 |
| 10 | 20 | 0.9385 | 1.4230 | 0.02 | 192 | 26 | 7.38 |
| 20 | 10 | 1.5000 | 1.6084 | 0.4 | 78 | 10 | 7.80 |
| 20 | 10 | 1.6968 | 2.1175 | 0.1 | 116 | 15 | 7.73 |
| 20 | 10 | 1.6903 | 1.8224 | 0.02 | 200 | 26 | 7.69 |
| 20 | 20 | 1.4726 | 2.2386 | 0.4 | 200 | 66 | 3.03 |
| 20 | 20 | 1.3968 | 1.8276 | 0.1 | 63 | 21 | 3.00 |
| 20 | 20 | 1.6259 | 1.8174 | 0.02 | 200 | 68 | 2.94 |
| 30 | 30 | 1.6655 | 1.9283 | 0.4 | 74 | 96 | 0.77 |
| 30 | 30 | 1.5186 | 1.7557 | 0.1 | 60 | 84 | 0.71 |
| 30 | 30 | 0.7727 | 0.8679 | 0.02 | 62 | 87 | 0.71 |

Табела 8.13: Приказ резултата тренирања вештачке неуронске мреже са три скривена слоја (растојање до препрека 40 [mm])

| Бр. н. с. с. 1. | Бр. н. с. с. 2. | Бр. н. с. с. 3. | rmse_train | rmse_test | Пар. учења | Бр. Итер. | Време [s] | Итер. / [s] |
|--------------------|--------------------|--------------------|------------------------|-----------|---------------|--------------|--------------|----------------|
| 4 | 8 | 4 | 2.0269 | 2.2518 | 0.2 | 200 | 8 | 25.00 |
| 4 | 8 | 4 | 2.0398 | 2.2921 | 0.02 | 144 | 5 | 28.80 |
| 10 | 5 | 5 | 1.7515 | 2.0795 | 0.2 | 104 | 5 | 20.80 |
| 10 | 5 | 5 | 0.8259 | 1.2088 | 0.02 | 120 | 6 | 20.00 |
| 6 | 6 | 12 | 2.1065 | 2.3369 | 0.2 | 200 | 12 | 16.67 |
| 6 | 6 | 12 | 1.5686 | 1.7816 | 0.02 | 200 | 13 | 15.38 |
| 10 | 10 | 10 | 2.4768 | 2.6269 | 0.2 | 13 | 2 | 6.50 |
| 10 | 10 | 10 | 1.3957 | 1.5339 | 0.02 | 51 | 6 | 8.50 |
| 12 | 16 | 12 | 1.3643 | 2.1313 | 0.2 | 61 | 14 | 4.36 |
| 12 | 16 | 12 | 0.1757 | 1.7148 | 0.02 | 58 | 13 | 4.46 |
| 19 | 13 | 13 | 1.6716 | 1.8030 | 0.2 | 181 | 54 | 3.35 |
| 19 | 13 | 13 | 0.9342 | 2.2135 | 0.02 | 96 | 28 | 3.43 |
| 17 | 11 | 23 | 1.7179 | 1.8208 | 0.2 | 28 | 10 | 2.80 |
| 17 | 11 | 23 | 0.0756 | 1.2359 | 0.02 | 200 | 76 | 2.63 |
| 20 | 20 | 20 | 1.2616 | 1.8048 | 0.2 | 112 | 95 | 1.18 |
| 20 | 20 | 20 | $4.4952 \cdot 10^{-7}$ | 1.6461 | 0.02 | 200 | 178 | 1.12 |

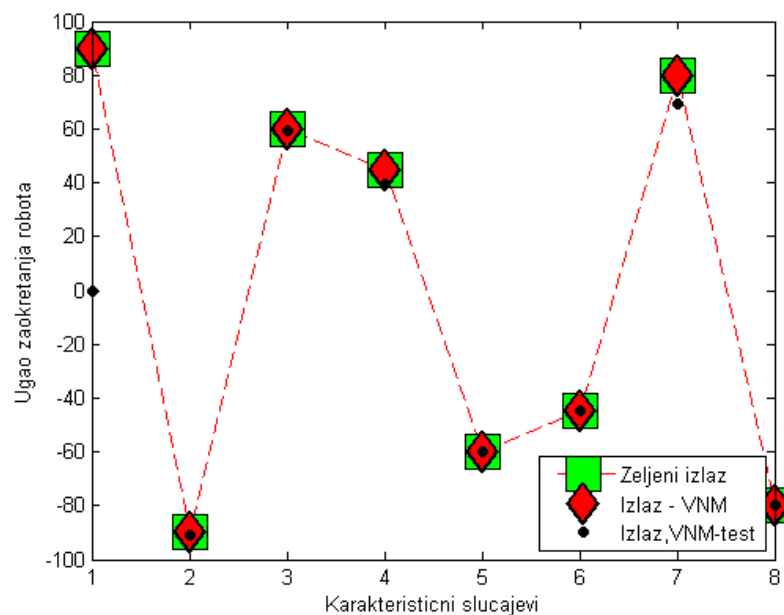
Као што се види, мрежа је дала задовољавајуће резултате само у два случаја, за мрежу са једним скривеним слојем и 50 неурона у њему и за мрежу са три скривена слоја са по 20 неурона у сваком од њих. Међутим, у оба случаја корен средње квадратне грешке, за тест скуп, није био задовољавајући. Зато је у претходним поглављима речено да мобилни робот не може да доноси одлуке само на основу ове мреже, разлог овоме је што су улазни подаци коришћени за обучавање били на граници сензорских могућности. Из овог разлога и постоје две мреже, а ова служи само за неки вид раног

детектовања препрека и тиме припреми главну⁷ мрежу на евентуалну препреку. Самим тим повећава се шанса да главна мрежа да боље резултате, односно донесе правилну одлуку. Иако излаз из мреже није задовољавајући, покушано је варирањем параметра учења да се побољша излаз из мреже. Резултати су дати у табели 8.14.

Табела 8.14: Варирање параметра учења за мрежу са најбољим резултатима (растојање до препрека 40 [mm])

| Бр. неурона | rmse_train | rmse_test | Пар. учења | Бр. итерација | Vreme [s] | Итер./ [s] |
|-------------|-------------------------|-----------|------------|---------------|-----------|------------|
| 50 | $1.2901 \cdot 10^{-8}$ | 2.2981 | 0.9 | 77 | 35 | 2.20 |
| 50 | 1.3963 | 1.6742 | 0.7 | 200 | 91 | 2.19 |
| 50 | 0.4152 | 2.8764 | 0.2 | 200 | 92 | 2.17 |
| 50 | 1.1127 | 1.8774 | 0.01 | 200 | 90 | 2.22 |
| 50 | 0.8272 | 1.9588 | 0.001 | 13 | 6 | 2.17 |
| 50 | $1.9298 \cdot 10^{-11}$ | 0.7573 | 0.0001 | 66 | 28 | 2.35 |

И код ове мреже најбољи резултати су добијени за вредност параметра учења од 0.0001 што наводи на закључак да за овакве улазне податке и за овакав проблем, најбоље је параметру учења доделити вредност 0.0001. На слици 8.5 дат је приказ излаза за ову мрежу.



Слика 8.5: Приказ излаза неуронске мреже након завршеног процеса обучавања

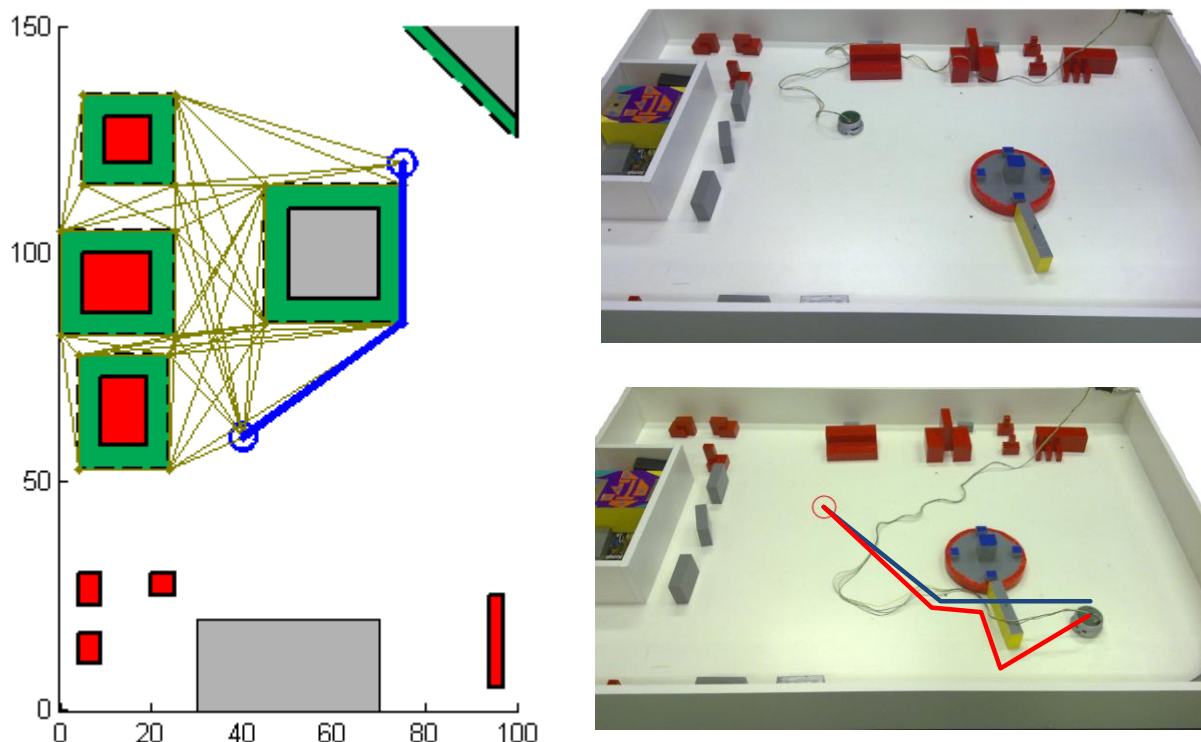
Мреже су тестиране и у реалним условима, при раду мобилног робота. Добијени резултати представљени су матрицом одлучивања и могу се видети у табели 7.1.

⁷ С обзиром да друга мрежа углавном служи само за детектовање препрека, прва (она која детектује препреке на 20 [mm]) се назива главном мрежом.

8.3 Резултати експеримента праћења путање са избегавањем препрека

Приказани су резултати за сваки сегмент посебно, сада ће бити дати резултати за целокупан процес кретања робота по моделу технолошког окружења са избегавањем препрека. Неопходно је и утврдити, да ли се грешка и колико повећава са појавом непредвиђених препрека у окружењу.

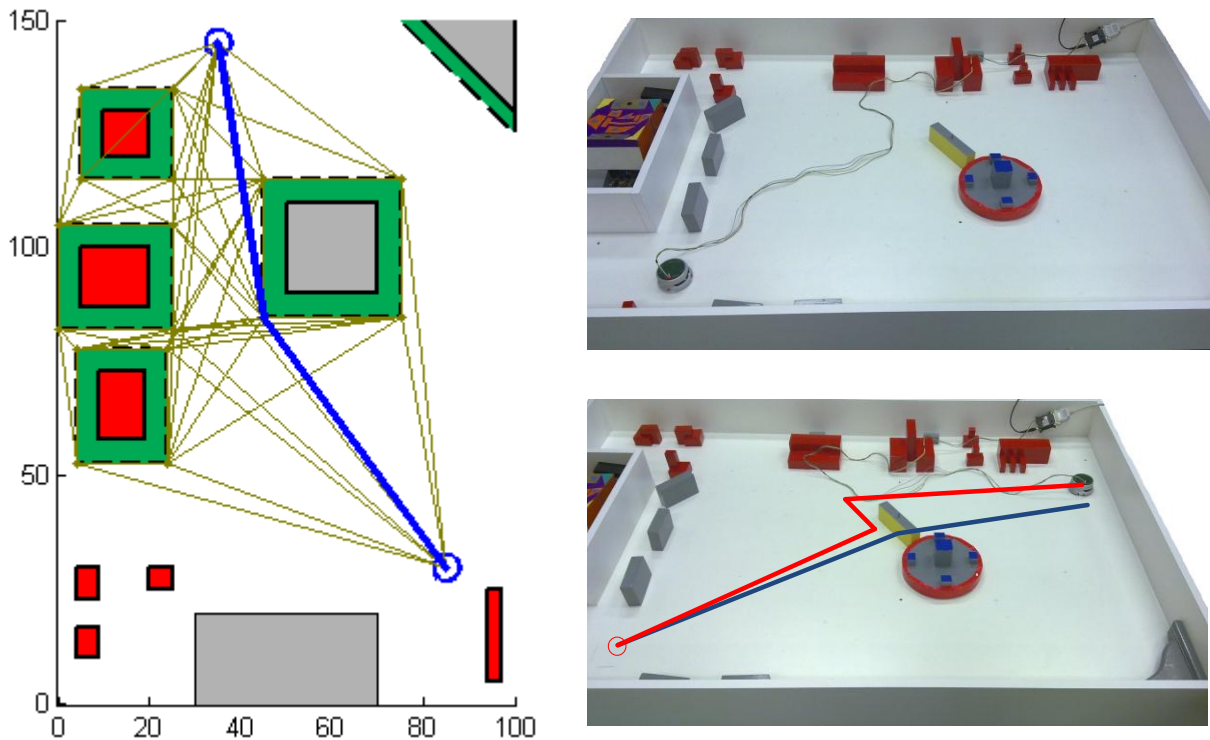
Задатак се састоји у праћењу путање од тачке А до тачке В. На путању се у току кретања постављају препреке о којима робот нема никакве претходне информације. Брзина робота је $30 [mm/s]$. Координате старта $[40 60]$, а циља $[75 110]$.



Слика 8.6: Са леве стране приказ генерисане путање, са десне стране остварена путања у току кретања (експеримент 1)

На слици 8.6 приказана је путања која је добијена у *matlab-u*, као и остварена путања у току кретања. Остварена грешка износи $5.5 [cm]$ по X оси и $0.5 [cm]$ по Y оси.

Урађен је још један експеримент са различитим координатама. Координате старта су $[85 30]$, а циља $[35 135]$.



Слика 8.7: Са леве стране приказ генерисане путање, са десне стране остварена путања у току кретања (експеримент 1)

На слици 8.7 приказана је путања која је добијена у *matlab-u*, као и остварена путања у току кретања. Остварена грешка износи 6.5 [cm] по X оси и 1 [cm] по Y оси. Разлог што је грешка по X оси већа од грешке по Y оси је непрецизност при заокретању. Грешка је готово идентична у оба случаја, што наводи на закључак да грешка није производ лоше обучене ВММ или непрецизности робота, јер и када нема препрека у окружењу добијена грешка се готово поклапа са грешкама у горе наведеним експериментима.

До те грешке долази јер морају паралелно да раде и ИЦ сензори и мотори, а информације се шаљу преко серијског кабла брзином 38400 *bps*⁸. Тако да се робот не заустави у правом тренутку већ се окрене за пар степени више, а затим настави да се креће за израчунато растојање. Ово доводи до тога да се јавља већа грешка по X оси у односу на Y осу. Остварени резултати су ипак задовољавајћи, али правилном оптимизацијом кода и неким видом паралелног процесирања могуће је драстично смањити грешку. Из овога закључујемо да модел кретања као и модул за планирање путање раде веома успешно. Модул за избегавање препрека се такође показао као врло успешан. Препреке су на време уочене и успешно избегнуте.

⁸ *bps* – bits per second

9. Закључак

Примарни задатак овог дипломског рада је да покаже могућности и да докаже употребљивост интелигентног мобилног робота у модерним технолошким окружењима. Циљ који треба постићи је да интелигентни мобилни робот, који има унапред дефинисану путању и редослед опслуживања машина, буде у могућности да избегне евентуалне препреке у окружењу и настави са обављањем свог задатка.

Да би ово могло да се оствари неопходно је било развити и имплементирати модел кретања. Овим је одговорено на једно од главних питања које интелигентни мобилни роботи „морају” да знају, а то је „где сам?”. Такође, неопходно је да робот има одговоре и на питања „где сам био?” „где ми је циљ?” и „како да дођем до циља?”. Одговоре на ова питања обезбедили су алгоритми за планирање путање кретања, формирање графа видљивости као и његова претрага ради утврђивања оптималне путање. Такође је неопходно обезбедити робусност система у случају непредвиђених догађања. Правовремено реаговање на поремећаје у окружењу изискивало је употребу вештачких неуронских мрежа. Сврха ВМ у овом случају је да препозна поремећаје који су настали у систему и одреди начин на који да их превазиђе. Примена вештачких неуронских мрежа има широку заступљеност код интелигентних мобилних робота, највише због њихове способности да нађу корелацију између улазних и жељених излазних података. Метода која је коришћена базира се на машинском учењу, применом система вештачких неуронских мрежа, због њихових могућности да се прилагоде променама, које у систему изазивају поремећајни фактори и да на основу тога уче и минимализују грешку после сваке следеће итерације.

У поглављу 8 приказани су добијени резултати процеса обучавања ВМ као и резултати добијени у реалним условима. На крају су урађени експерименти где је робот морао да дође до одређеног циља, укључујући и избегавање препреке. Грешка износи око 5.5-6.5 [cm] по X оси и 0.5-1 [cm] по Y оси. С обзиром на пређени пут, процентуално изражено, грешка по X оси износи од 13 - 15%, док је по Y оси грешка 1%. Грешка по X оси је резултат лоше оптимизованог програмског кода као и споре комуникације између рачунара и решавањем ових проблема смањили би и грешку по X оси на 1%. Дакле, може се констатовати да је експеримент успео, робот са задовољавајућом тачношћу прати задату путању, детектује препреке и избегава их.

Унапређење система могуће је постићи на више начина. Иако је грешка мала при кретању мобилног робота по технолошком окружењу, како се повећава пређени пут тако и грешка расте, па је неопходно увести додатне сензоре у циљу побољшања локализације, јер само одометријска локализација није довољна. Ово се може урадити коришћењем светлосних сензора или камере за препознавање карактеристичних објеката у окружењу и тиме повећати вероватноћу успешности обављања одређеног технолошког задатка.

Реаговање робота на поремећаје у окружењу могуће је побољшати коришћењем камере, као основног сензора, за препознавање препрека. Ако постоји захтев за манипулацију са предметима, неопходне су сензорске информације са камере ради препознавања предмета.

Мапирање простора се обично користи у непознатим окружењима, али у овом случају се може искористити за мапирање препрека. Тако би било могуће имати комплетне информације о објекту који се налази на путањи робота и самим тим избегавање препрека би било много прецизније и успешније одрађено.

10. Литература

- [1] Sigwert, R., Nourbakhsh, R., Illah, *Introduction to Autonomus Mobile Robots*, Massachusetts Institute of Technology, 2004.
- [2] Lozano-Perez, T., Wesley, M., A., *An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles*, *Communications of the ACM*, Vol. 22, No. 10, October 1979.
- [3] Велагић, Ј., *Увод у Мобилну Роботику*, Универзитет у Сарајеву - Електротехнички факултет, 2009.
- [4] Subir, K., Ghosh, *Visibility-based Robot Path Planning*, School of Technology & Computer Science, Tata Institute of Fundamental Research, Mumbai, India.
- [5] Reinhard, D., *Graph Theory*, SpringerVerlag New York, 2000.
- [6] Zhan, F., B., Noon, C., E., *Shortest Path Algorithms: An Evaluation Using Real Road Networks*, February, 1998.
- [7] Миљковић, З., *Предавања са предмета методе одлучивања*, дипломске академске студије, Универзитет у Београду - Машински факултет, Београд, 2009.
- [8] Миљковић, З., *Системи вештачких неуронских мрежа у производним технологијама*, Научна монографија (ISBN 86-7083-455-3), Серија монографских дела Интелигентни технолошки системи (Уредник серије: Проф. др Владимир Милачић), Књига 8, Универзитет у Београду -Машински факултет, VI+185 стр., Београд, 2003.
- [9] Миљковић, З., Александрић, Д., *Вештачке Неуронске Мреже – Збирка решених задатака са изводима из теорије*, Универзитет у Београду – Машински факултет, 2009.
- [10] Park, J., Sandberg, J. W. *Universal approximation using radial basis functions network*, *Neural Computation*, vol. 3, pp. 246-257, 1991.
- [11] Poggio, T., Girosi, F., *Networks for approximation and learning*, *Proc. IEEE*, vol. 78, no. 9, pp. 1481-1497, 1990.
- [12] Haykin, S. *Neural Networks: A comprehensive Foundation*. Upper Saddle River, NJ: Prentice Hall, 1994.
- [13] Adrian, G., Bors, *Introduction of the Radial Basis Function (RBF) Networks*, Department of Computer Science, University of York, YO10 5DD, UK.
- [14] Christopher, R., Bishop, *Neural Networks for Pattern Recognition*, Department of Computer Science and Applied Mathematics, Aston University, Birmingham, UK, 1995.
- [15] Chen, S., Hong, X., Harris., C., J., *Fully complex-valued radial basis function networks for orthogonal least squares regression*, *IJCNN* 2008
- [16] Chen, S., Hong, X., Harris., C., J., *Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design*, *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 1029-1036, Jun, 2003.

[17] Haykin, S., *Neural Networks, A Comprehensive Foundation*, McMaster University, Hamilton, Ontario, Canada, 1999.

[18] Бабић, Б., Предавања са предмета аксиоматске методе, дипломске академске студије, Универзитет у Београду – Машински факултет, Београд, 2009.

11. Прилог

11.1 Главни fajl

```
clear all, clc

%=====
%VAZNO!!!
load net6
load net7
%Load net mora da stoji pre "ref = kopen([0, 38400, 1]);" inace dolazi do
%problema i gresaka pri radu.
%=====
ref = kopen([0, 38400, 1]);
%=====
run RBF_NN_IR_V1;
%Tezinski odnosi iz RBF mreze; RBF mreza je ostala za odredjivanje
%rastojanja robota do prepreka, dok su FF za prepoznavanje prepreka
%=====

disp('Start')

%=====
Robot = [20 20]; % pocetne koordinate robota
Cilj = [30 50]; % cilj :)
%=====
[A,Cilj,Robot] =
Funkcija_za_proveravanje_ispravnosti_unetih_koordinata(Cilj,Robot);
% Proverava da li robot fizicki moze da ostvari zadate pozicije
%=====
[path1] = Funkcija_za_odredjivanje_preseka (Robot,Cilj);
% Generisanje putanje, Visibility graph i sve vezano za to
%=====
Robot = Robot.*10; % prebacivajne u mm
x = [Robot 90/57.3]'; %[mm]
dist = 45; % [mm] Rastojanje do prepreke kada treba da se zaustavi
qa = 0.4623; % obim kruga 166.42mm/360 stepeni, koliko milimetra po stepenu
%=====

path1 = path1(:,3:4); % koristimo samo 3 i 4. kolonu
path1 = path1.*10; % prebacivanje u mm

kSetEncoders(ref);

%=====
[x,path1,E,E2] =
Funkcija_sa_svim_funkcijama(ref,x,dist,qa,Cilj,net6,net7,path1);
% funkcija u kojoj se nalaze sve ostale funkcije za kretanje,skretanje,
% izbegavanje prepreka, prepoznavanje prepreka itd...
%=====
x
kclose(ref)
```

11.2 Kod sa funkcijama za kretanje robota

```
function [x,path1,E,E2,wp1,wp2] =
Funkcija_sa_svim_funkcijama(ref,x,dist,qa,Cilj,net6,net7,path1);

%=====
path1 = [x(1:2,:)';path1];
% Matrica sa koordinatama kroz koje robot treba da prodje bi dosao do cilja
%=====
f = size(path1);

for k = 1:f(1)-1
    % disp ' u for petlji u funkciji sa svim funkcijama'
x
    E = 0; E2=0;
    %=====
    wp1 = x(1:2,:)'; % wp1 - trenutna pozicija robota
    %wp1 = path1 (k,:)
    wp2 = path1 (k+1,:); % wp2 - pozicija u koju treba da dodje
    %=====
    [d,fi] = compute_offsets(x,wp1,wp2);
    % fi = round(fi);
    % Funkcija koja racuna predjeni put i ugao(d,fi)
    %=====
    [n,m,Vl,Vr] = Funkcija_za_n_i_m(fi);
    % n i m odredjuju smer, Vl i Vr odredjuju brzinu, levog i desnog tocka
    %=====
    fi=round(fi);
    fil = abs(fi)*qa
    % Put koji tockovi treba da predju da bi se robot zaokrenuo za ugao fi
    %=====
    if fi ~= 0
        [E,E2,x,path1] = Funkcija_za_skretanje
(Vl,Vr,ref,x,fil,net6,dist,E,d,E2,path1,Cilj,net7,wp1,wp2);
        end
        [x,E,path1] = Funkcija_za_kretanje
(Vl,Vr,ref,x,d,dist,net6,E,E2,path1,Cilj,net7,wp1,wp2);

    % Funkcije koje racunaju i izvrsavaju skretanje i kretanje robota
    %=====
end
```

11.3 Код за скретање робота

```
function [E,E2,x,path1,wp1,wp2] = Funkcija_za_skretanje
(Vl,Vr,ref,x,fil,net6,dist,E,d,E2,path1,Cilj,net7,wp1,wp2)

C = eye(3)*0;kl = 10^-3; kr = 10^-3;b = 53; enc_old2 = 0; E2=0;

kSetEncoders(ref);

while 1

    kSetSpeed(ref,Vl/8, Vr/8);          % Send wheel speeds to robot [mm/s]
    enc2 = kGetEncoders(ref); % encoder's reading in pulses
    e2 = (enc2 - enc_old2)*.0833 ; % in [mm]
    E2 = E2 + (abs(e2(1))+abs(e2(2)))/2;
    [x,C] = Khe_odometry(x,C,e2,b,kl,kr); % x in [mm]
    enc_old2 = enc2;

    % drawmob_robot(x,3.5,'r');

    if (E2+1)>= (fil/2);

        kSetSpeed(ref,(Vl/2)/8,(Vr/2)/8); % Send wheel speeds to
robot [mm/s]
        enc2 = kGetEncoders(ref); % encoder's reading in pulses
        e2 = (enc2 - enc_old2)*.0833 ; % in [mm]
        E2 = E2 + (abs(e2(1))+abs(e2(2)))/2;
        [x,C] = Khe_odometry(x,C,e2,b,kl,kr); % x in [mm]
        enc_old2 = enc2;

        % drawmob_robot(x,3.5,'r');

        E2 = E2+0.3;

        if (E2) >= fil

            [x,E,path1] = Funkcija_za_prinudno_skretanje_test
(Vl,Vr,dist,ref,net6,x,d,E,E2,path1,Cilj,net7,wp1,wp2);

            break
        end
    end
end
return
```

11.4 Код за pravolinijsko kretanje

```
function [x,E,path1,wp1,wp2] = Funkcija_za_kretanje
(Vl,Vr,ref,x,d,dist,net6,E,E2,path1,Cilj,net7,wp1,wp2)

enc_old = 0; C = eye(3)*0;kl = 10^-3; kr = 10^-3;b = 53;

fi = 0;
kSetEncoders(ref);

speeds = [30 30];
Vl = speeds(1);    % left wheel
Vr = speeds(2);    % right wheel
while 1

    p = kProximity(ref);
    [yy] = Khepera_IR_RBF_NN(p);

    if yy(3) < dist || yy(4) < dist || yy(2) < dist-30 || yy(5) < dist-30
    || yy(1) < dist-35 || yy(6) < dist-35

        [x,E,path1] = Funkcija_za_prinudno_skretanje_test
(Vl,Vr,dist,ref,net6,x,d,E,E2,path1,Cilj,net7,wp1,wp2);

        break
    end

    kSetSpeed(ref,Vl/8, Vr/8);    % Send wheel speeds to robot [mm/s]
    enc = kGetEncoders(ref); % encoder's reading in pulses
    e = (enc - enc_old)*.0833;    % in [mm]
    E = E + (e(1)+e(2))/2;
    [x,C] = Khe_odometry(x, C, e,b,kl,kr); % x in [mm]
    enc_old = enc;

    if E >= d;

        kStop(ref);

        break
    end

end

return
```

11.5 Kod za zaobilazjeње prepreka i praćenje zida

```
function [x,E,path1,wp1,wp2] = Funkcija_za_prinudno_skretanje_test
(Vl,Vr,dist,ref,net6,x,d,E,E2,path1,Cilj,net7,wp1,wp2)

qa = 0.4623;

p = kProximity(ref);

[yy] = Khepera_IR_RBF_NN(p);
if yy(3) < dist || yy(4) < dist || yy(2) < dist-30 || yy(5) < dist-30

%=====
% Plan B - da mi mreza odredi koji je slucaj u pitanju pa da na osnovu
toga
% izracunam ugao
% p = kProximity(ref);
% y = sim(net4,p);
% % kSetencoders(ref);
% t = find(y == max(y));
% [fi] = Funkcija_za_t_za_prepreke (t)

%=====
% Plan A - da mi mreza odredi odmah ugao za koji treba da se zaokrene
robot
[fi,x,E] = funkcija_za_odredjivanje_ugla (ref,net6,net7,dist,x);

%=====

fi=round(fi);
fil = abs(fi)*qa;
[n,m,Vl,Vr] = Funkcija_za_n_i_m(fi);

[E,E2,x,path1,wp1,wp2] = Funkcija_za_skretanje
(Vl,Vr,ref,x,fil,net6,dist,E,d,E2,path1,Cilj,net7,wp1,wp2);

kStop(ref);

Robot = round(x(1:2))';
Robot = Robot./10;

[path1] = Funkcija_za_odredjivanje_preseka (Robot,Cilj);

path1 = path1(:,3:4);
path1 = path1.*10;

[x,path1,E,E2,wp1,wp2] =
Funkcija_sa_svim_funkcijama(ref,x,dist,qa,Cilj,net6,net7,path1);

end

kStop(ref);
p = kProximity(ref);
[yy] = Khepera_IR_RBF_NN(p);
```

```
zlj = yy(8)*(0.8987); % cos od 26
zdj = yy(7)*(0.9335); % cos od 21

lj = (yy(1)*(0.9397));
dj = (yy(6)*(0.8192));

if (7 < lj & lj < 50) || (7 < dj & dj < 50)

    [x,E,path1,wp1,wp2] = Funkcija_za_kretanje_test
(Vl,Vr,ref,x,E,dist,net6,d,E2,path1,Cilj,net7,wp1,wp2);

    kStop(ref);

    return
end
return
kStop(ref)
```