



УНИВЕРЗИТЕТ У
БЕОГРАДУ
МАШИНСКИ ФАКУЛТЕТ

Предмет: **КОМПЈУТЕРСКА СИМУЛАЦИЈА
И ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА**

Завршни рад В.Сс.

Лист/листова: 1/57

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



Увод

Вештачка интелигенција (Artificial Intelligence - AI) представља мешавину конвенционалне науке, физиологије и психологије, све у циљу да се направи машина која би се, по људским мерилима могла сматрати "интелигентном". Знање и AI. AI се често повезује са концептом знања. Мада рачунар не може имати искуства или проучавати и учити као што чине људи, он може користити знање које му дају људски експерти. Такво знање се састоји од чињеница, концепта, теорија, хеуристичких метода, процедура и односа. Знање је информација која је организована и анализирана да би се учинила разумљивом и примењивом за решавање проблема или доношење одлука. Да би знања која се односе на одређени проблем била примењена у интелигентном систему, она се организују и чувају у бази знања. Скуп знања у вези са активношћу у организацијама назива се организациона база знања. Могућност стварања интелигентних машина заокупља људску машту још од древних времена, али тек сада, са брзим темпом развоја рачунара и већ педесетогодишњим искуством на пољу истраживања технике AI програмирања, сан о паметним машинама почео је да остаје стварност. Научници су коначно способни да створе системе који разумеју говор и могу да победе најбоље играче шаха, док је хардвер са елементима вештачке интелигенције већ испробан у оружаним сукобима (систем крстарећих пројектила које је америчка војска користила у пустињској олуји). Са одређеним програмима за IBM и Apple Macintosh персоналне рачунаре, вештачка интелигенција је почела да полако, али неуморно задире и у свакодневни живот. Међутим поред свег напретка, научници се још увек нису ухватили у коштац са основним изазовом у развоју интелигентне машине – прављењем система који опонашају људски мозак. Није ни чудо, јер људски мозак са више милијарди неурона представља можда најсложенију творевину у универзуму.

Историја вештачке интелигенције

Од давнина људе привлачи интелигенција, како сам појам интелигенције тако и могућност конструисања „интелигентних машина“ које би могле самостално да раде. Такве примере можемо пратити кроз историју.

- У старој Грчкој људи су се бавили питањима интелигенције, знања и правилног закључивања, а међу њима се нарочито истакао *Аристотел*.
- У XIII веку је *Рамон Лул* (1235-1316) описао систем *Ars Magna* којим је покушао да помоћу механичког комбиновања, симболичке нотације и комбинаторних дијаграма оствари „интелигентан“ систем.
- Током XVII века, *Г.В. Лајбниц* (1646-1716) и *Блазе Паскал* (1623-1662) покушавали су да конструишу механичку рачунску машину за сабирање.
- Цифарска рачунска машина коју је конструисао *Чарлс Бабаж*, била је у стању да по одређеном алгоритму извршава операције са декадним бројевима.
- Средином 19-ог века *Џорџ Бул* разрађује алгебру логике у којој се алгебарска симболика користи за оперисање појмова при логичком извођењу.
- Енглеz *Алан Туринг* и Американац *Пост*, 1936, године, независно један од другог објављују радове из области математичке логике и износе могућности конструисања универзалног трансформатора информација.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



- Међутим, тек појавом првог рачунара „Electronic Numerical Integrator And Computer” (ENIAC) кога су 1945. године измислили *Маучли* и *Ј. Преспер Екерт*, може се говорити о интелигентним машинама.

Термин „вештачка интелигенција” употребљен је први пут 1956. године на конгресу у Дартмуту, у Новој Шкотској (Канада). Конгрес је организовао *Џон Мек Карти*, којег многи данас сматрају оцем вештачке интелигенције. Иако конференција сама по себи није представљала велики успех, пионери на пољу вештачке интелигенције окупили су се на једном месту ради размене идеја и ударени су темељи за будући рад на пољу. Годину дана касније, тестирана је прва верзија новог програма под називом *The General Problem Solver (GPS)*. Програм је био способан за решавање већег броја општих проблема. Пар година касније, IBM је створио тим за истраживање вештачке интелигенције. *Херберт Гелернетер* провео је три године на развоју програма за решавање геометријских теорема. Упоредо са појавом нових програма, *Мек Карти* је спремао велико откриће у историји вештачке интелигенције – рачунарски језик LISP, који се још увек користи.

Напредак у шездесетим годинама прошлог века обележила је појава експертских система који изводе предвиђања вероватноће решења у задатим условима. Десетак година касније, експертски системи употребљавани су за прогнозе на тржишту вредносних папира и у медицинској дијагностици.

Током осамдесетих година прошлог века, вештачка интелигенција је развијана много бржим темпом. Поље истраживања је такође све чешће залазило у корпорацијски сектор. У 1986. години, продаја програма и хардверске опреме заснованих на AI износила је само у SAD око 425 милиона долара. Компаније као *Digital Electronics* користиле су XCON, експертски систем пројектован за програмирање великих VAX рачунара. Компаније *DuPont*, *General Motors* и *Boeing* су такође у великој мери користиле експертске системе.

Појава персоналних рачунара убрзала је развој AI и донекле променила смер улагања. Осамдесетих година прошлог века појавило се и ново поље на истраживању AI – вештачки визуелни системи. Тако је рад научника са MIT-а омогућио постављање камера и рачунара у индустрији, као средстава контроле квалитета. Иако груби, ови системи могли су да разликују облике и предмете. До 1985. године, преко стотину компанија нудило је вештачке визуелне системе само у SAD, а продаја је достигла износ од 80 милиона долара. Међутим, крајем осамдесетих година, потражња за AI системима је опала, а индустрија је изгубила готово пола милијарде долара. Због тога су и улагања у истраживање AI система опала.

У Јапану је развијана нова технологија. У SAD је развијена тзв. „нејасна логика”, која је намењена обради непрецизних или променљивих података. За разлику од традиционалних бинарних вредности, ова логика користи велики опсег вредности ради веће флексибилности. На тај начин, рачунар је могао да донесе закључке у неизвесним условима. Као могући начин за постизање AI разматране су и неуронске мреже. Осамдесете године прошлог века су увеле AI на корпорацијско тржиште и показале да технологија има конкретне примене. Многи очекују да ће AI технологије бити кључне у 21. веку.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



Интелигентни сиситеми и вештачка интелигенција

Израз *интелигентни системи* описује различите комерцијалне апликације вештачке интелигенције. Вештачка интелигенција је израз који обухвата неколико дефиниција. Већина експерата се слаже да се AI односи на две основне идеје. Прво, она подразумева проучавање процеса размишљања код људи; друго, она се бави представљањем тих процеса преко машина (рачунара, робота...).

AI је „понашање машине које би било названо интелигентним када би се тако понашало људско биће”. AI је грана рачунарске науке која се бави начинима представљања знања, коришћењем симбола пре него бројева, и хеуристиком, односно искустваним правилом пре него алгоритмима за обраду информација.

Способности које се сматрају знацима интелигенције су следеће:

- учење или схватање на бази искуства,
- уочавање смисла у неодређеним или контрадикторним порукама,
- брзо и успешно реаговање на нове ситуације,
- решавањем проблема размишљањем,
- решавањем сложених ситуација,
- схватање и закључивање на уобичајен, рационалан начин,
- примењивање знања при деловању у окружењу,
- препознавање релативног значаја разних елемената у датој ситуацији.

Коначни циљ AI је саградити машине које ће опонашати људску интелигенцију. Како време пролази, интелигентни системи постају све бољи и корисни су већ данас због ефикасности многих послова који захтевају људску интелигенцију.

Предности AI. Упркос ограничењима, методи AI могу бити изузетно вредни. Они могу чинити рачунаре једноставнијим за употребу и допринети да знање буде широко доступно. Главне потенцијалне предности AI се састоје у томе да:

- значајно повећава брзину и доследност појединих поступака у решавању проблема;
- помаже у решавању проблема који не могу бити решени или су компликовани за решавање конвенцијалном обрадом на рачунару;
- помаже при решавању проблема са некомплетним или нејасним подацима;
- помаже у регулисању информационог преоптерећења резимирањем и интерпретирањем информација и у претраживању велике количине података;
- знатно повећава продуктивност при обављању многих послова;
- чини врло једноставном употребу неких рачунарских апликација.

Конвенционална обрада на супрот AI, обради на рачунару. Конвенционални рачунарски програми су засновани на алгоритмима. Алгоритам је математичка формула или секвенцијална процедура која води до решења. Алгоритам се конвертује у рачунарски програм који прецизно говори рачунару које операције да изврши да би решио проблеме, азатим користи податке као што су бројеви, слова или речи.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



AI софтвер заснован на симболичкој обради знања. У AI је симбол слово, реч или број који представља објекте, процесе и њихове односе. Објекти могу бити људи, ствари, идеје, концепти, догађаји или констатовање чињеница. Употребом симбола могуће је формирати базу знања која садржи чињенице, концепте и њихове међусобне односе. Потом се различити процеси могу користити за употребу симбола ради генерисања савета или препоруке за решавање проблема.

Основне разлике између AI обраде на рачунару и конвенционалне употребе рачунара приказане су у следећој табели:

Табела 1.1. Основне разлике између AI обраде на рачунару и конвенционалне употребе рачунара.

Вештачка интелигенција	Конвенционално програмирање
Претежно симболичка обрада	Претежно нумеричка обрада
Хеуристичко претраживање	Алгоритми
Управљчке структуре најчешће издвојене од знања	Информације и управљање интегрисани заједно
Једноставно модификовање, ажурирање и проширивање	Тешко модификовање
Толеришу се погрешни одговори	Неопходни су тачни одговори



1. Интелигентни агенти у технолошким системима 21. века – улога и значај

1.1. Дефиниција Интелигентних агената

Изненађујуће или не, не постоји званична дефиниција агента, пошто су и даље отворене дебате о томе и и даље постоје контроверзе везане за појам агената. Ипак постоје неке основе на којима се агенти заснивају и око којих нема спорова, тако да ћемо њих искористити за дефинисање појма.

Било који агент се може посматрати као систем који помоћу сензора опажа своју околину и на основу тих података делује (одговара) ефекторима. Ова сажета представа агента на први поглед сугерише да је он састављен искључиво од физичких делова, али да би они могли да функционишу складно, потребна им је веза у облику меког медијума која пропушта и процесира информације кроз агента. То могу бити разни алгоритми, софтверска решења, математички модели. Потребно је дефинисати и објаснити још неке појмове како би било јасније шта представља **интелигентни агент**.

1.2. Појам интелигенције

У једној од дефиниција интелигенције, како њени главни чиниоци, између осталог се наводе способност планирања, решавање проблема, брзо учење на основу искустава и коришћење наученог садржаја. Данашњи интелигентни агенти су способни да подрже све поменуте појмове, а оно што их тренутно разликује од људи по питању интелигенције јесте недостатак емоционалне интелигенције, апстрактног размишљања, развијање комплексних идеја.

1.3. Аутономност

Интеллигентни агент има позитиван учинак ако обавља посао за који је намењен што је могуће боље. Ако свој задатак извршава и правовремено онда он има одређену дозу аутономности што је један од битних чинилаца интелигентних агената.

Аутономност се протеже толико колико и реаговање агента зависи од стеченог искуства. Уколико агент испољава активност на основу неког „уграђеног“ знања онда он представља програм који круто врши неку функцију. Ако реагује на основу искуства онда је интелигентан. Поставља се питање да ли је интелигентни агент направљен тако да поседује неко иницијално знање? Одговор је позитиван, то знање је способност учења односно неки од алгоритама учења који омогућавају да оно са чиме се агент сусретне не буде само прошлост следећег тренутка, већ искуство које може да се искористи приликом будућих активности. Овакав агент је у могућности да приликом промене у његовом окружењу промени своје понашање и не дође у ситуацију да испољи понашање које резултира потпуним неуспехом. Агенти који на овај начин одговарају промени окружења су флексибилни.

Лошија страна могућности постизања интелигенције, аутономности и флексибилности је време које је потребно да агент научи нешто и оно зависи од комплексности проблема или промене окружења.

1.4. Структура интелигентних агената

Софтверски део агента (програм), се покреће и ради на рачунару или некој рачунарској машини. Овде се већ примећује подела на хардверски и софтверски део.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

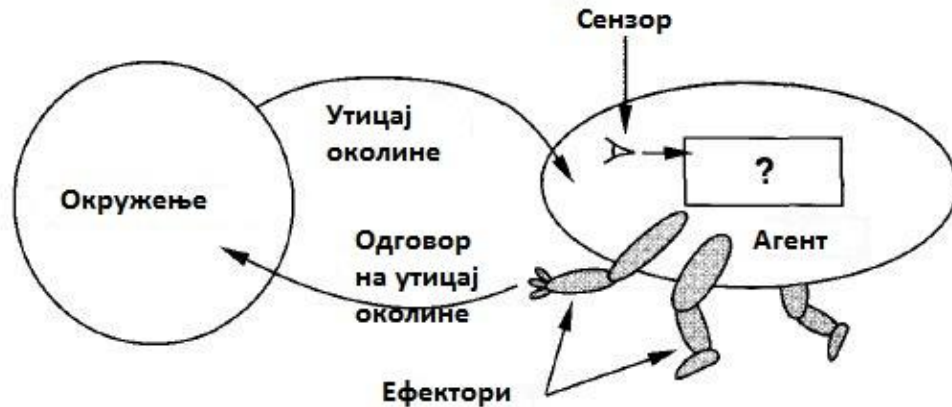
Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



Слика 1.1 Графички приказ интелигентног агента

Хардверски део агента је компјутер као први који се налази у контакту са програмом. Даљим усложњавањем проблема хардвер може да се разграна на сензоре и ефекторе, а сви они на групе и подгрупе у зависности од начина на који сигнал из околине треба да је предочен агенту. Формализована структура интелигентног агента би изгледала као збир архитектуре и програма, где је архитектура носилац опипљивих вредности. Ради избегавања збуњивања и погрешне интерпретације термина, треба напоменути да се појам архитектуре појављује и у чисто софтверским оквирима као структура вештачке неуронске мреже, о чему ће касније бити речи.

1.5. Окружење

Интелигентни агенти имају различиту архитектуру у зависности од тога какав поремећај из окружења треба да приме и какав одговор треба да дају. Различите врсте поремећаја потичу од различитих окружења, с тога треба навести поделу окружења по неколико критеријума.

Окружење може бити приступачно или неприступачно са становишта сензора. Приступачно је ако агент може да сагледа цело окружење или макар да сагледа све релевантне параметре.

Детерминистичко или недетерминистичко. Ако окружење прелази из једног стања у друг према неко закону и ако је свако следеће стање познато на основу претходног онда је окружење детерминистичко.

Епизодно или неепизодно. Код епизодног окружења, агент не мора да води рачуна о претходним или следећим епизодама јер су неповезане, јављају се дискретно. Реакција агента зависи само од квалитета епизоде у којој тренутно ради. Статичко или динамичко. Окружење је динамичко ако агент мора да врши опажање док доноси одлуку о даљем наставку рада и будућим активностима због тога што се окружење непрекидно мења. У супротном, окружење је статичко.

Дискретно или континуално. Ако се парови опажања и дејствовања агента одвијају сукцесивно онда је окружење дискретно, а ако је потребно да агент опажа и одговара на опажаје стално, без неког временског разграничења, онда је окружење континуално. Јасно је, да је неприступачно, неепизодно, динамичко, континуално окружење најтеже за сналажење интелигентних агената и при томе највећи изазов за оне који се баве осмишљавањем концепата за решавање таквих задатака.



1.6. Интелигентни системи

Интеллигентни системи су системи који раде самостално или полу-аутономно у неизвесном окружењу са минималним надзором и интеракцијом са људским оператерима. Такви системи су вођени контролама са посебним карактеристикама, техникама и методама познатим као интелигентно управљање. Интелигентне машине су у основи дефинисане као и хијерархијска структура у циљу постизања интелигентног понашања. Ова интелигенција се зове Машинска интелигенција која је уграђен у хијерархијску структуру машине за потребе извршавања задатака са минималним учешћем људских оператера.

Интеллигентне машине захтевају употребу генерализованих контролисаних акција ради обављања функције, као што је истовремено корисћење меморије, адаптације на околину и само-организацију, реагујући на корисничке захтеве. Већина покушаја дизајнирања интелигентних система и машина су у почетку биле концентрисане у коришћењу људске логике и хеуристике (засноване на логичком приступу) у настојању да имитирају функције и активности људског мозга. Алтернативни приступ је конвенционалнији аналитички приступ.

Интеллигентно управљање је развијено за спровођење функција хијерархијског система и машина и могу се сматрати спојем математичке и језичке методе и алгоритама који се примењују на системе и процесе. Интелигентно управљање је хијерархијски устројено, у складу са принципом повећања интелигенције, евидентно у свим хијерархијским структурама. Широко прихваћена структура се састоји од три нивоа: контроле, организације, координације и извршења.

1.7. Примена, улога и значај

Примена интелигентних агената у технолошким системима је уведена ради неутралисања утицаја људске грешке, убрзања производње, повећања продуктивности и побољшања квалитета. Примењују се у транспорту, манипулацији, монтажи, класификација, обради, провери квалитета итд.

У транспорту се роботи као ИА користе на местима где није могуће поставити тракасте транспортере, у окружењима која се динамички мењају. За транспорт у окружењима неповољним и опасним за рад људи, као што су радиоактивна или хемијски загађена окружења.

Приликом манипулације роботи убрзавају производни процес, уклањају опасност од повреда оператера, лако манипулишу деловима велике масе. Због своје сложене конфигурације која је условљена добром покретљивошћу робота, некада је потребно да се у кратком временском периоду управљање робота врши узастопним решавањем диференцијалних једначина, ради решавања инверзног кинематичког проблема, што може бити проблематично чак и за рачунар. Интелигентни агент омогућава решавање тог проблема на основу претходно стечених искустава тако што користи нека генералисана правила везана за понашање окружења у коме се налази.

Класификација у интелигентним технолошким системима се између осталог врши на основу геометријских и технолошких сличности делова. Оваква класификација нас уводи у групну технологију која је веома продуктивна у комбинацији са нумерички управљаним машинама алаткама. Класификацијом према технолошким или геометријским карактеристикама омогућава боље искоришћење производних средстава тако што делове групише у фамилије и класе које се могу обрађивати на истим обрадним системима. Вештачком интелигенцијом се врши технолошко препознавање, одређивање репрезента класе делова, утврђивање сличности, а затим грубо и фино пројектовање технологије које се на крају свде само на типске захвате што убрзава цео производни процес.

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.



ИА у обради су агилне машине алатке које подешавају режиме обраде у зависности од чинилаца који утичу на квалитет обраде. Оваква концепција машина алатки спречава колизије обрадка и алата, обрадка и дела машине као и могуће повреде оператера. Омогућавају максимално искоришћење енергетског биланса машине, повећавају продуктивност и квалитет обраде.

Архитектура ИА у технолошким системима се у главном састоји од сензора и ефектора.

Сензори могу бити :

- светлосни
- ултразвучни
- ласерски
- сензори за додир
- сензори за температуру
- камере

Особина свих сензора је да поремећај из околине региструју и претварају у електричне имполсе које шаљу софтверском делу ИА.

Ефектори су :

- мотори
- хваталке
- вентили
- пригушивачи

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



2. Мобилни LEGO робот – функција оптичких (светлосних) сензора, итд.

2.1. LEGO MINDSTORMS комплет

Комплет **LEGO MINDSTORMS** компаније **LEGO** састоји се од више модуларних елемената од којих је могуће креирати различите роботе, и софтвера NXT-G за њихово програмирање.

У комплету се налазе:

- 619 модуларних елемената, за склапање носеће конструкције и система преноса.
- NXT управљачка јединица - микро-контролер који активира сензоре и моторе са одговарајућим параметрима добијеним од софтвера и врши читавања вредности са сензора.
- Сензор за додир
- Ултразвучни сензор – за мерење удаљености препреке
- Светлосни сензор– за мерење контраста осветљења
- Звучни сензор – за мерење нивоа буке
- 3 Серво мотора са уграђеним енкодерима за мерење ротације
- 7 каблова за повезивање мотора и сензора са управљачком јединицом

Веза између софтвера и управљачке јединице остварује се преко USB (Universal Serial Bus) или Bluetooth (бежичног) интерфејса. Могуће је такође докупити додатне сензоре произведене од стране компаније **LEGO**, или других компанија, као што су: жироскоп, компас, термометар, RFID читач, акцелерометар и др.

2.2. NXT управљачка јединица

Главна компонента у комплету је рачунар у облику цигле назван NXT Intelligent Brick. Може да преузима одзиве са и до четири сензора и контролише до три мотора, преко RJ12 кабла, врло сличног, али некомпатибилног са RJ11 телефонским каблом. Има 100 x 64 пиксела монохроматски ЛЦД екран, резолуције 100 x 64 пиксела и четири тастера која се могу користити за навигацију кроз кориснички интерфејс помоћу хијерархијских менија. Она такође има звучник и може да репродукује аудио датотеке фреквенције до 8 kHz. Напаја се са 6 AA батерија (1,5 V свака).



Слика 2.1.1 NXT управљачка јединица

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

2.3. Сензор за додир

Сензор за додир детектује да ли је тренутно притиснут или пуштени. Наранџасти и сиви (десни и леви) тастери на управљачкој јединици могу се програмирати да служе као сензори за додир. NXT-G софтверу за програмирање, дата вредност 0 је кад није притиснут, а вредност 1 када је притиснут.



Слика 2.1.2 Сензор за додир

2.4. Ултразвучни сензор

Ултразвучни сензор може да измери растојање од сензора до објекта ка коме је оријентисан, а да детектује покрет. Максимална раздаљина коју може да измери је 233 цм, са прецизношћу од 3 цм. Ултразвучни сензор ради тако што шаље таласе звука (високе фреквенције) који се одбијају од објекта испред њега, а затим назад враћају назад. Мерењем времена које је потребно за повратак таласа индиректним путем се одређује раздаљина.



Слика 2.1.3 Ултразвучни сензор

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

2.5. Светлосни сензор

Светлосни сензор детектује ниво јачине светла у једном правцу, а такође укључује и ЛЕД за осветљавање објекта. Светлосни сензор може да детектује вредности светла одбијеног од објекта (помоћу уграђеног црвеног ЛЕД-а), или амбијенталног светла. NXT-G софтверу за програмирање сензор даје вредности јачине светла на скали од 0 до 100, 100 је веома светло а 0 веома тамно. Ако се калибрише за одређене услове, сензор може да се користи за одређивање раздаљине.



Слика 2.1.4 Светлосни сензор

2.6. Звучни сензор

Звучни сензор мери ниво јачине звука. NXT-G софтверу за програмирање сензор даје вредности јачине звука на скали од 0 до 100, 100 је веома гласно, 0 је потпуна тишина.

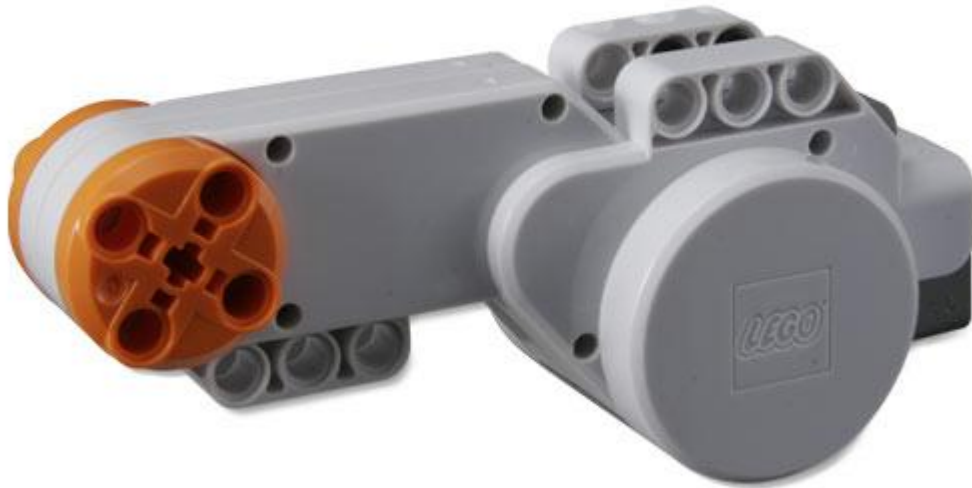


Слика 2.1.5 Звучни сензор

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

2.7. Серво мотор

Комплет садржи 3 идентична серво мотора са уграђеним редуктором и оптичким енкодером ротације, прецизности $\pm 1^\circ$. Мотори раде на напону од 9V и имају уграђен замајац за одржавање константне брзине при променама оптерећења (таква употреба може довести и до оштећења мотора).



Слика 2.1.6 Серво мотор

2.8. Конфигурације мобилних робота

Помоћу **LEGO MINDSTORMS** комплета могуће је склопити велики број различитих мобилних конфигурација. Основна подела мобилних конфигурација може се направити на основу начина остваривања кретања. У њој можемо издвојити две главне групе. У прву спадају роботи који позајмљују дизајн природе, као што су: роботи који остварују кретање помоћу ногу (антропоморфни, квадрипод, хексапод и др.), роботи који остварују кретање пузањем и др. Друга група су роботи који остварују кретање помоћу једног од најбитнијих изума човечанства – точка.

Док су роботи прве групе много прилагодљивији различитим подлогама, главне мане које их чине непогодним за примену у индустријским условима затвореног погона су: компликована конструкција (самим тим скупља и сложенија производња) и мања брзина коју остварују на равном терену у односу на роботе који се крећу помоћу точкава.

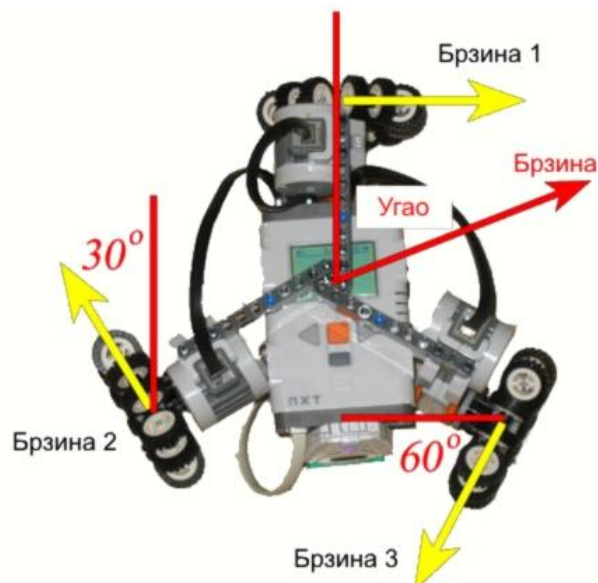
Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



Слика 2.2.1 Пример хексаподне конфигурације

Најпогодније конфигурације за мобилног робота који се креће помоћу точкова, оствариве помоћу комплета компаније **LEGO**, су роботи са диференцијалним погоном тј. роботи код којих сваки точак има мотор који га покреће. Диференцијални погон омогућава једноставнију конструкцију, знатно мањих габарита и флексибилније управљање.

Неке од конфигурација са диференцијалним погоном су:



Слика 2.2.2 Омнидирекциона конфигурација са три точка

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

Омнидирекциона конфигурација са три точка омогућава кретање у свим правцима уз минималне потребе за ротацијом робота, задавањем одговарајуће брзине сваком од мотора за жељени правац кретања. Захтева посебне омнидирекционе тачкове.



Слика 2.2.3 Конфигурација са два точка и задњим ослонцем

Конфигурација са два точка и задњим ослонцем омогућава транслаторно кретање када се оба точка окрећу истом брзином и у истом смеру, кружно кретање када се точкови окрећу у истом смеру различитим брзинама или ротацију око своје осе када се точкови окрећу истом брзином али у супротном смеру. Као ослонац се користи точак због мањег трења, али могуће је користити и другачије ослонце. Такође, постоје и конфигурације са 2 ослонца или конфигурације код којих се трећи точак користи за управљање.

2.9. Функција сензора

Функција сензора код различитих конфигурација мобилних **LEGO** робота огледа се претежно у прикупљању података из окружења који су неопходни за неометано кретање робота до његовог циља.

Ултразвучни сензори најбољу примену проналазе у детектовању стационарних или динамичких (покретних) препрека. Такође се могу користити за одређивање раздаљине између робота и препреке, а ротацијом сензора или самог робота може се доћи до детаљнијих података о окружењу који се могу искористити за формирање мапе простора и избор најбоље трајекторије. Међутим њихова употребна вредност ограничена је наведеним степеном прецизности и несигурношћу читавања у случајевима када се звучни таласи одбијају од препреке неравних површина или површина које су закошене под углом већим од 45° у односу на раван сензора.



Светлосни сензори проналазе примену у читавању контролних тачака на подлози при чему је неопходно да је контраст између боје тачака и боје подлоге што већи (нпр. црна и бела боја). Могу се, до одређеног нивоа, примењивати и за класификацију објеката уз претходно наведене услове (велика контрастна разлика између боја различитих објеката) и углавном само за две врсте објеката (две различите боје). Примена сензора за одређивање раздаљине могућа је уз одговарајуће калибрисање, у окружењу са константним контролисаним нивоом јачине светлости или у потпуном мраку. У оваквим условима и прве две функције дају знатно боље резултате (препознавање више боја, мања могућност грешке и др.).

Ротациони енкодери на моторима су кључни за одређивање позиције и оријентације робота у односу на неки референтни координатни систем. Уз употребу одговарајућег математичког модела за употребљену конфигурацију робот је у стању да прорачуна сопствену позицију. Грешка мерења код енкодера повећава се са већим бројем обртаја а код комплекснијих трајекторија долази до акумулације грешке. У циљу смањивања ових грешака препоручљиво је да робот кретање обавља у виду мањих инкремената путање са поништавањем читавања по завршетку сваког инкремента. Такође се могу применити и напреднији модели са корекцијом грешке када она достигне одређени ниво.

Поред свих наведених недостатака сензори, адекватном применом, могу дати задовољавајуће резултате, у складу са првенствено едукационом применом робота. Употребом вештачких неуронских мрежа и сложенијих алгоритама управљања резултати сензора се могу значајно побољшати.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



3. Вештачке неуронске мреже у интелигентном управљању мобилним роботима

Контрола покрета у биоолошким и вештачким системима захтева присутност сензомоторних трансформација које претварају сигнале са сензора у команде мотора код машина, а информације из чула у покрете мишића. Ове трансформације су комплексне, нелинеарне, и јако их је тешко изразити у строго аналитичкој форми, пошто се неретко извршавају у системима где су појаве стохастичке. Вештачке неуронске мреже могу бити коришћене за учење једне или више сензомоторних трансформација у функцији да извршавају задатке без прецизних уграђених знања о тренутним параметрима окружења и робота, што касније резултира памћењем и коришћењем оптималних управљачких параметара. Овакав концепт управљања може редуковати значајну количину израчунавања која су присурна приликом програмирања и извршавања програма, тако да имплементација робота у технолошки систем може да се изврши брже и јефтиније.

3.1. Разлика између надгледаног и ненадгледаног учења

Са теоријског становишта, надгледано и ненадгледано учење се разликују само према узрочно последичној структури модела. Код надгледаног учења се дефинише утицај једног сета података на други сет података, који су добијени посматрањем појаве која се моделује. Први сет података су улази, а други излази. Другим речима претпоставља се да се улази налазе на почетку а излази на крају узрочног ланца.

У случају ненадгледаног учења, сматра се да су излази узроковани неким потенцијалним променљивима. Ненадгледано учење оставља могућност моделовања проблема и решења без тачне претпоставке шта утиче на исход рада система. Овакав модел учења није неопходан све док постоје макар статистички подаци о утицајима који делују на систем. Са ненадгледаним учењем је могуће решавати много комплексније проблеме него са надгледаним. Узрок томе је експоненцијални пораст компликованости учења са повећањем нивоа учења, код надгледаног, дефинисног сетом улазних и излазних података. Ненадгледано учење се врши помоћу жељеног понашања система као излазних параметара и потенцијалних променљивих параметара на улазу. Сваки виши ниво учења линеарно отежава учење, што је повољнији случај у односу на надгледано учење.

3.2. Комбиновано учење

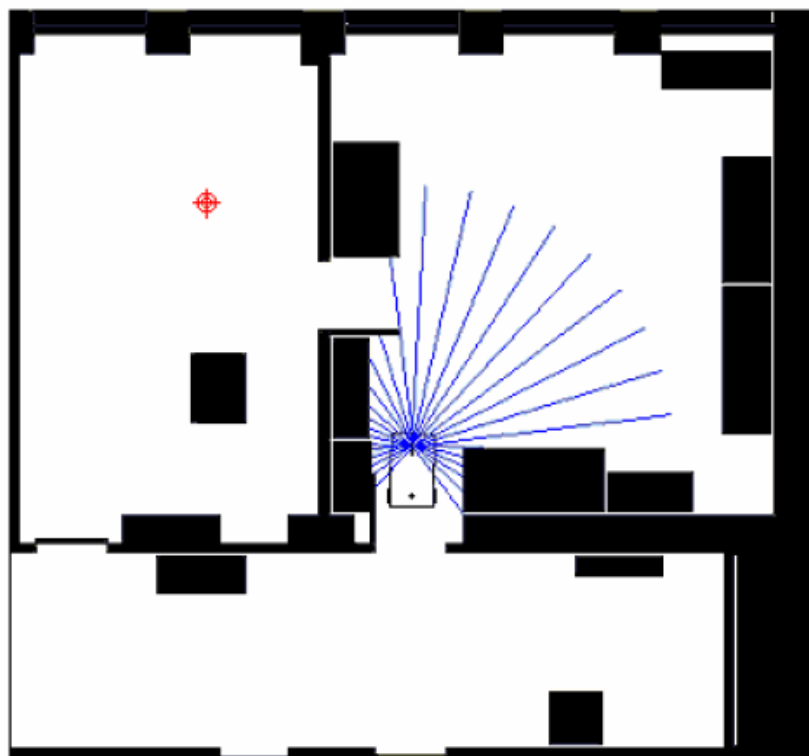
Један од конкретних проблема је планирање и налажење путање мобилног робота у окружењу које има насумично постављене препреке где су задати почетни услови у облику почетне и крајње позиције и оријентације. Један од начина за решавање овог реалног проблема је коришћење две вештачке неуронске мреже. Једна мрежа је са ненадгледаним учењем, а друга је са надгледаим. Основни хардвер овог мобилног робота чине мотори и ултразвучни сензори који могу радијално да читавају препреке у односу на робота. О моторима и детаљнијој конфигурацији овде не ће бити речи јер је за сада битније објаснити како интелигентни агент учи од свог окружења него како шаље повратне сигнале ка њему. Фазе овог алгоритма су следеће:

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



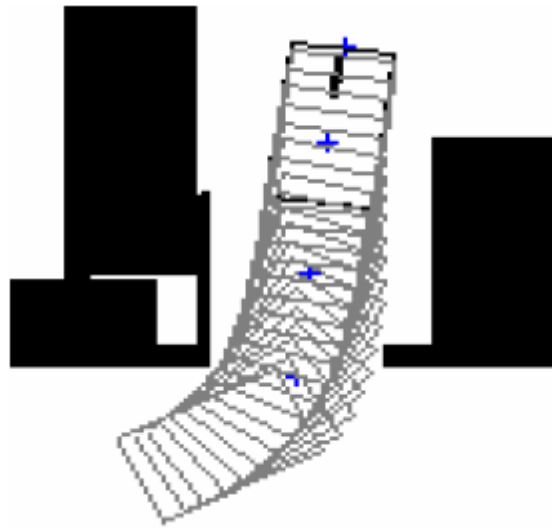
- Одредити стартну и циљну позицију
- Изједначити тренутну и циљну оријентацију
- Активирати сензор и утврдити локални изглед окружења
- Позвати прву неуронску мрежу
- Позвати другу неуронску мрежу
- Робот се креће са датом оријентацијом
- Вратити се на трећу ставку

Једна неуронска мрежа се користи за мапирање слободног простора, а друга има задатак да на основу рада прве мреже испланира сигурну путању до циља. Прва неуронска мрежа на свом улазу прима податке о простору од ултразвучног сензора. Улазних података има колико има смерова у којима се мери раздаљина од робота. Дакле, подаци су бројне вредности удаљености објеката од робота. На излазу се посматрају правци кретања на којима се налазе или не налазе препрека и помоћу њих се фаворизују или инхибирају утицаји ордеђених улаза у зависности да ли омогућавају или не остваривање излаза. Излаз прве мреже је један улаз за другу мрежу. Друга мрежа на улазу добија још и циљну позицију тако да може да генерализује правило по коме треба реаговати окретањем за одређени угао који у исто време представља излазну вредност за обучавање.

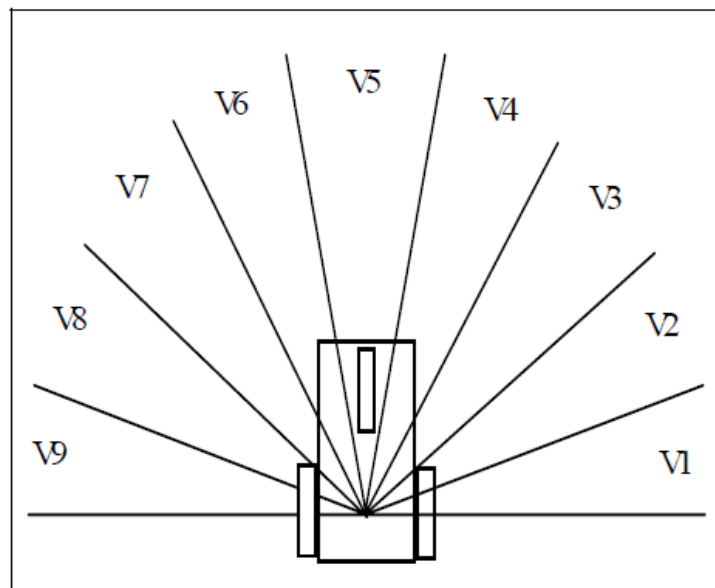


Слика 3.1 Мапирање слободног простора

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



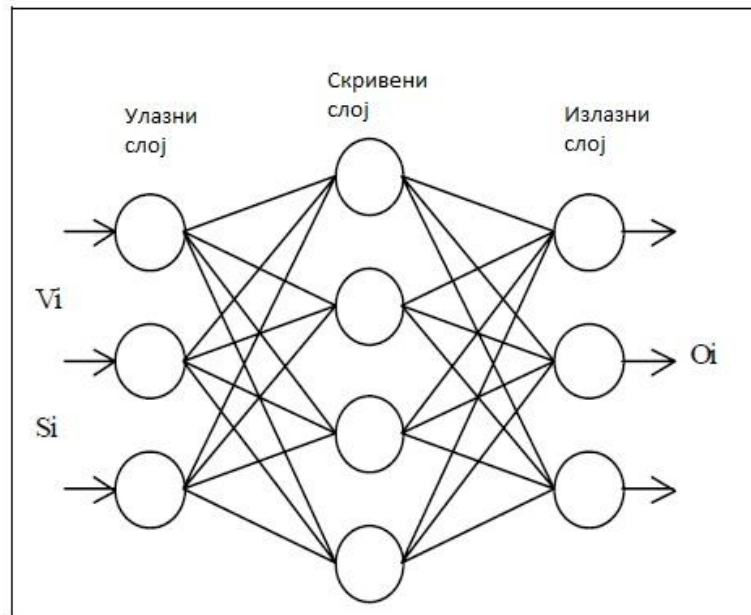
Слика 3.2 Сукцесивно окретање робота и транслаторно померање по испланираној путањи (више итерација)



Слика 3.3 Правци снимања околине

На основу снимљеног простора формира се n -димензиони вектор могућих кретања односно, дефинише се слободна и препречена путања у зависности од раздаљине препрека у односу на мобилног робота. Тако дефинисана једна компонента улаза. Друга компонента улаза је циљна тачка. Идеја је да се мобилни робот креће ка циљу најкраћим путем ако је слободан тако што ће извршавати транслаторно кретање када год се оријентише ка циљу. Оријентација ка циљној тачки се извршава кад год је могуће. Ако постоји препрека, прелази се на следећи итерацију и тако до коначног циља.

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



Слика 3.4 Упростићени приказ коришћене мреже где је V_i - вектор могућих праваца кретања, S_i - позиција циљне тачке, O_i - излазни угао ротације

Коришћењем више различитих или истих сензора, усложњава се процес обраде информација које постају доступне, али се повођавају могућности управљања и контроле мобилних робота. Један од сензора погодних за коришћење код мобилних робота је камера. Развој рачунара и дигиталне технике омогућили су декомпозицију дигиталне слике као и анализу њени појединих делова. Само процесирање слике је доста сложен софтверски проблем чији се принцип овде само укратко помиње. У комбинацији са сензором даљине препрека може бити корисно средство оријентације мобилног робота у простору. Сензори за мерење даљине објеката спречавају могућност колизије мобилног робота и делова из околине, док се заједничким коришћењем слека камере и неуронских мрежа може вршити препознавање облика, одређивање њихове оријентације, специфичности простора и изградња мапе окружења за потпуну оријентацију у радном простору. Ако се претпостави да је дигитализована слика бинарна онда је онда састављена само од нула и јединица и таква је погодна за коришћење у векторском или матричном облику. Применом надгледаног и ненадгледаног учења, матричне и векторске улазе, мобилни робот, односно, неуронска мрежа преводи у управљачке сигнале мотора који условљавају артикулисано кретање у датим условима.



4. Роботизовани унутрашњи транспорт и манипулација деловима у интелигентном технолошком систему

Глобализација светске економије и раст конкурентности на тржишту, намеће све веће захтеве модерног предузетништва. Тренутно, управљање и контрола предузећа је немогућа без примене одговарајућих алата за одлучивање у свим фазама од пројектовања до експлоатације производа. CIM (Computer Integrated Manufacturing) системи су пример тако доступних алата који омогућавају композитна аутоматизација технолошких и организационих припрема за производњу, тренутни надзор, контролу технолошког процеса, организације и управљања. Развој CIM система је у последњих неколико година, била усмерена на примену метода вештачке интелигенције да подржи процесе у одлучивању и контроли производње, као и праћење, симулацију и технолошке процесе дијагнозе. IM (Intelligent Manufacturing) је најновија идеја у развоју аутоматизације и компјутерској интеграцији система за производњу. Дефиниција IM: Скуп метода и поступака CAD, CAP, CAM алата опремљених са вештачком интелигенцијом за подршку пројектовања, планирања и производње. Између осталог, основни елементи *Intelligent Manufacturing* система су:

- Интелигентне машине и алати, односно нумерички контролисане машине и роботи
- Интелигентни производни системи
- Интелигентни системи за управљање.

4.1. Основне петпоставке везане за интелигентне технолошке системе

Интелигентни технолошки системи између осталог представља информациони систем у коме се чувају неопходне информације и у коме се врши њихова анализа. Садржи основу метода одлучивања у циљу асистирања приликом доношења одлука у свакој фази производње, као и могућност учења и адаптације на динамичке промене окружења. Другим речима то је систем који подржава одлучивање на основу примењених метода из домена вештачке интелигенције, који решавају комплексне проблеме или процесира непоуздане, контрадикторне или тешко формализујуће информације.

Основни захтеви интелигентних технолошких система су:

- Могућност сакупљања и процесирања различитих типова информација из свих извора, унутрашњих и спољашњих, у циљу сачињавања модела знања потребног за доношење одлука. У исто време, потребно је да може сачинити модел знања и на основу људског размишљања.
- Треба постојати могућност анализе претходно стечених знања новим интелигентним методама.
- Постојање детектора критичних ситуација, обавештавање и правовремено реаговање. Анализа датих ситуација мора се извршити у реалном времену.
- Узимање у обзир недостатка стабилности, динамичких промена и учење система у фази пројектовања.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

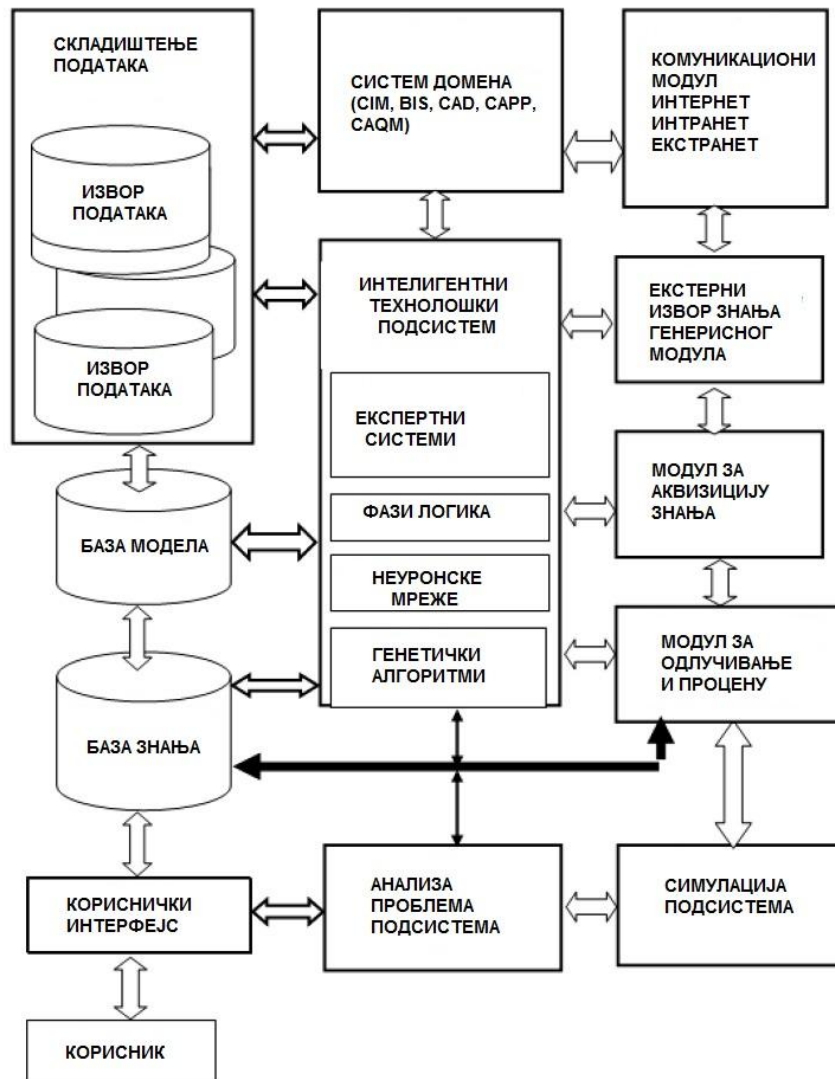
Прегледао:

4.2. Модел интелигентног технолошког система

У подстављеном приступу интелигентним технолошким системима захтева се да буде отворне архитектуре са модуларном структуром која омогућава неколико метода примене знања као и неколико метода учења. Треба да има интегрисане савремене методе и технике које чине:

- Вештачке неуронске мреже – моделују компликоване функције високог степена
- *Fuzzy logic*
- Генетички алгоритми

Комбиновањем ових алата, у којима је знање представљено симболички, са традиционалним експертским системима омогућавају креирање комплексних програмских алата за налажење оптималних решења и доношење правих одлука.



Слика 4.1 Структурни дијаграм интелигентног технолошког система

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



Транспорт у интелигентном технолошком систему мора да одговара бара на неке захтеве који су постављени целом систему. На основу тога може се закључити да се транспорт материјала у таквом систему мора вршити помоћу неке интелигентне машине. Ако претпоставимо да је то аутономни мобилни робот онда је његово кретање подређено стању у целом систему. Он је у стању да обезбеди сигурно кретање кроз радни простор уз испуњавање захтева који се тичу прецизности доласка на циљ и временских ограничења како не би дошло до кашњења у извршавању операција чије започињање зависи од времена завршетка транспорта. Пошто је интелигентни технолошки систем уједно и информациони систем, способан је да дистрибуира битне информације својим подсистемима. На тај начин, роботу који врши транспорт су доступне информације које нису њему директно доступне кроз сензорски систем корисне за његов континуални рад и за рад целог система као што су почетак или завршетак циклуса производње, загушење неког дела постројења, отказ неке машине... Интелигентност и аутономност транспорта се огледа у примени интелигентних агената помоћу којих се генерализују правила понашања у радном окружењу како не би било „изненађујућих“ ситуација током транспорта. Нове ситуације се користе за стицање искустава и остају забележене међу подацима и информацијама које се чувају у информационом систему. Комуникација између агената је битна за оптимално искоришћење могућности постројења, у временском и енергетском смислу, као и у смислу постизања квалитета. То се постиже планирањем и доношењем одлука бирањем најбољих могућности, за шта су интелигентни агенти способни.

Сигурност у оваквим системима обезбеђује се сензорским подсистемима мобилног робота и одговарајућим правилима којих би требало да се придржавају запослени. Брзина кретања и сигурност су у контрадикцији, па треба наћи компромисно решење. У критичним ситуацијама активирају се режими рада који смањују вероватноћу колизије са машином или повреде људи који раде у близини мобилних робота. Простији систем смањивања могућности појаве нежељених ситуација је таква конфигурација робота који имају мању брзину него људи и она се примењује у мање аутоматизованим системима где има знатно више људи него мобилних робота. У таквим системима мобилни роботи обично немају високоодговорну функцију јер не могу да искористе своје могућности у погледу повећања функционалности, па је питање да ли на таквим местима постоји боља алтернатива. Овде се мисли на првенствено на окружења где мобилни роботи постоје као атракција, а не као оптимално решење. У производним постројењима мобилни роботи могу бити опремљени сигурносним браником који реагује када већ дође до нежељеног контакта, тако што тренутно прекида све функције мобилног робота.

Манипулација у интелигентном технолошком систему се врши помоћу робота различите зглобне конфигурације. Интелигентни агенти могу бити искоришћени на начин као и код мобилних робота. Планирање покрета, прикупљање искустава, препознавање технолошких форми и адаптивност су такође одлике једног манипулатора у интелигентном технолошком систему. Ако си уграђене дикиталне камере, које фиксне у односу на робота, дају му увид у стање околине само у домену резолуције слике, што често задовољава потребе процеса производње. Ако је потребна, комуникација и синхронизација са транспортерима се може обављати бежичним путем уз максимални комфор у раду. Класификација различитих делова је готово обавезна ако је потребна јер је та операција решена на више начина, а битна је са становишта организације и планирања целокупне производње. Манипулациони роботи морају бити флексибилни, односно не смеју застаревати и губити вункцију због немогућности испуњавања захтева производње. Поред интелигентних агената могу се користити и механичка својства робота као носиоци флексибилности, пре свега коришћењем редувантних робота (више од 6 степени слободе), који су физички погодни за рад у променљивим условима.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



5. Концепт примене интелигентних мобилних робота у функцији пешавања проблема унутрашњег транспорта у интелигентном технолошком систему – примена софтвера *AnyLogic*, *BPnet* и *Matlab*®. Рад обухвата концепцијско пешавање проблема унутрашњег транспорта и манипулације деловима, одређивање положаја мобилног робота у технолошком окружењу коришћењем сензорског подсистема (утицај окружења, препознавање објеката и заобилажење препрека – промена боје и/или осветљења објекта), кроз примену система вештачких неуронских мрежа

(Петровић Милош)

Основни проблем транспорта унутар технолошких система јесте динамичка промена окружења. Због тога је у таквим технолошким системима неопходно употребити интелигентног агента у виду интелигентног мобилног робота.

У овом раду представљен је концепт унутрашњег транспорта у интелигентном технолошком систему заснован на употреби вештачке интелигенције и машинског учења робота за транспорт и робота за манипулацију деловима. Модел концепта је реализован применом LEGO MINDSTORMS модуларних робота и софтверских пакета MATLAB и ANYLOGIC.

5.1. Концепт решења унутрашњег транспорта

Концепт који је овде приказан развијен је за потребе транспорта у технолошком систему производног погона замишљеног предузећа. Транспорт делова који пристижу у технолошки систем обавља се помоћу аутономног мобилног робота. Класификацију делова и њихово распоређивање на одговарајуће производне линије обавља стационарни робот. Пошто је предвиђено да оба робота раде у динамичном окружењу неопходно је да поседују способност прилагођавања новонасталим условима. У том циљу управљање мобилног робота је реализовано тако да је у стању да прилагоди своју путању стално променљивим положајима објеката у окружењу. Управљање стационарног робота је реализовано тако да омогућава преузимање делова из различитих положаја и оријентација мобилног робота. Класификација делова у моделу је извршена према њиховој боји.

5.2. Конструкцијска решења робота

Мобилни робот



Слика 5.2.1 Мобилни робот



Слика 5.2.2 Мобилни робот

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

Подсистеми мобилног робота :

Механички систем

Управљачка јединица *NXT* је искоришћена као носећа структура која повезује остале елементе у целини. Сви елементи робота су круто спојени *Lego* коцкама. На конструкцију механичког система су повезана два сервомотора која независно погоне два точка. Са задње стране управљачке јединице налази се ослонац.

Управљачки систем је *NXT* управљачка јединица.

Погонски систем су два сервомотора.

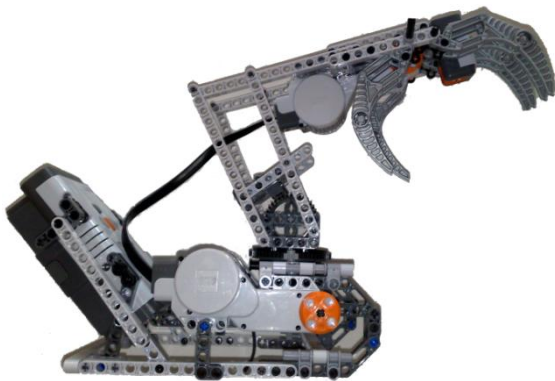
Мерни системи су унутрашњи сензори који мере број обртаја мотора, енкодери.

Сензорски систем се састоји од два ултразвучна сензора и једног оптичког сензора.

Техничке карактеристике

Мобилни робот поседује диференцијални погон на два точка којима постиже кружно и транслаторно кретање у равни.

Стационарни робот



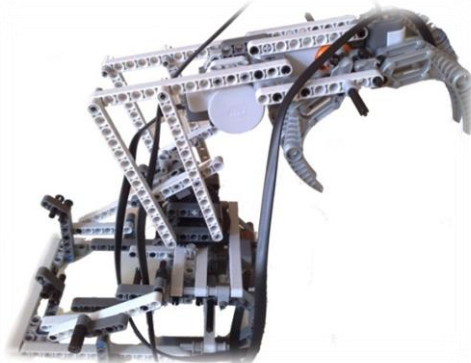
Слика 5.2.3 Стационарни робот



Слика 5.2.4 Стационарни робот

Подсистеми стационарног робота

Механички систем је састављен од *Lego* коцки. Носећи део је фиксиран и на њему се налазе управљачка јединица и мотори за ротацију у бази и ротацију у рамену. Рука је за базу везана преко планетарног преносника којим се преноси кретање са мотора за обе ротације. Мотор за покретање *end*-ефектора и оптички сензор се налазе на руци.



Слика 5.2.5 Машински подсистем стационарног робота

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

Енд-ефектор је хватач који се састоји из четири сегмента.



Слика 5.2.6 Енд ефектор стационарног робота

Управљачки систем је *NXT* управљачка јединица.

Погонски систем су три сервомотора.

Мерни системи су унутрашњи сензори који мере број обртаја мотора, енкодери.

Сензорски систем је један оптички сензор који врши класификацију делова.

Техничке карактеристике

Има два степена слободе, радни простор му је полукружни радијуса приближно 250 милиметара, носивост му је врло мала до 20 g. Програмирање се врши у *off-line* режиму, применом *Matlab* софтверског пакета.

Радни простор вештине стационарног робота је појас радијуса 220 милиметара са толеранцијом од ± 10 милиметара. Та површина је обележена црном бојом као оријентација за мобилног робота.

5.3. Софтверско решење управљања роботима

Управљање роботима се врши из софтверског пакета *Matlab* помоћу *RWTH NXT toolbox* -а додатног модула за *Matlab* развијеног за управљање *LEGO* роботима од стране студената и професора немачког *RWTH Aachen* универзитета. Веза између робота и рачунара на коме се покреће *Matlab* остварена је преко *Bluetooth* бежичног интерфејса. Такође су искоришћене могућности *Neural Network Toolbox*-а модула за рад са неуронским мрежама у *Matlab*-у.

Програм за управљање роботима може се посматрати као две одвојене целине интегрисане у један програм: програм за управљање мобилним роботом и програм за управљање стационарним роботом.

Програм за управљање мобилним роботом дели простор предвиђен за маневрисање мобилног робота у мрежу квадрата дужине страница 10 cm. Темена тих квадрата представљају контролне тачке помоћу којих робот формира своју трајекторију. Координате тачака су одређене у односу на сталну почетну позицију робота са које он започиње сваки циклус преношења дела до стационарног робота. Укупне димензије радног простора робота су 100 x 60 cm и он садржи 77 контролних тачака. Робот остварује кретање бирајући једну од суседних тачака, ротира се тако да му је предњи део са ултразвучним сензорима оријентисан ка изабраној тачки,

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.



проверава да ли се на путу до изабране тачке налази препрека и у зависности од одзива сензора наставља ка изабраној тачки или бира нову. Овај циклус се понавља све док мобилни робот не уђе у радни простор стационарног робота. Током кретања од тренутне до изабране тачке робот константно читава вредности са светлосног сензора усмереног ка подлози на којој се креће. Улога светлосног сензора је да региструје контролну линију у облику полукруга која ограничава радни простор стационарног робота. Када се то догоди мобилни робот зауставља своје кретање док стационарни робот не преузме део. Вештачка неуронска мрежа која руководи избором тачака обучена је тако да као прву тачку изабере ону тачку која је на најмањој раздаљини од координата стационарног робота. Ако је та тачка недоступна због препреке која се налази на правцу између изабране и тренутне тачке, мрежа узима за следећу тачку ону са првом већом раздаљином. Тако је омогућено да мобилни робот увек бира најкраћу слободну путању до стационарног робота. Систем са контролним тачкама је изабран јер робот остварује већу прецизност кретањем преко више мањих деоница путање. По завршетку сваке транслације и ротације програм прорачунава координате тренутне позиције робота и поништава стање на енкодерима, тако да кад мобилни робот заврши кретање програ је у стању да његове координате пренесе стационарном роботу са максималном грешком од ± 3 цм. Приликом повратка на почетну позицију робот се враћа путањом коју је меморисао у доласку.

Програм за управљање стационарним роботом као улазне податке користи излазе из програма за управљање мобилним роботом. Његови улази су вектор положаја мобилног робота $p(X1, Y1)$ и вредност координате $Y1$.

У програму се налази вештачка неуронска мрежа која на основу вектора положаја мобилног робота одређује потребан број обртаја мотора С, и програмска петља која на основу вредности координате $Y1$ одређује смер обртања ради извршења првог дела позиционирања руке. Други део позиционирања извршава мотор В који фиксним бројем обртаја доводи енд ефектор изнад дела. Затварање енд ефектора и хватање дела извршава мотор А, такође са фиксним бројем обртаја. Након затварања енд ефектора укључује се оптички сензор који читава боју дела. На основу очитане боје са сензора и броја обртаја мотора С очитаног са енкодера програм прорачунава потребан број обртаја за крајњу леву или крајњу десну позицију у зависности од боје дела. Након остављања дела рука се по фиксним бројевима обртаја мотора С и В враћа у првобитан положај.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

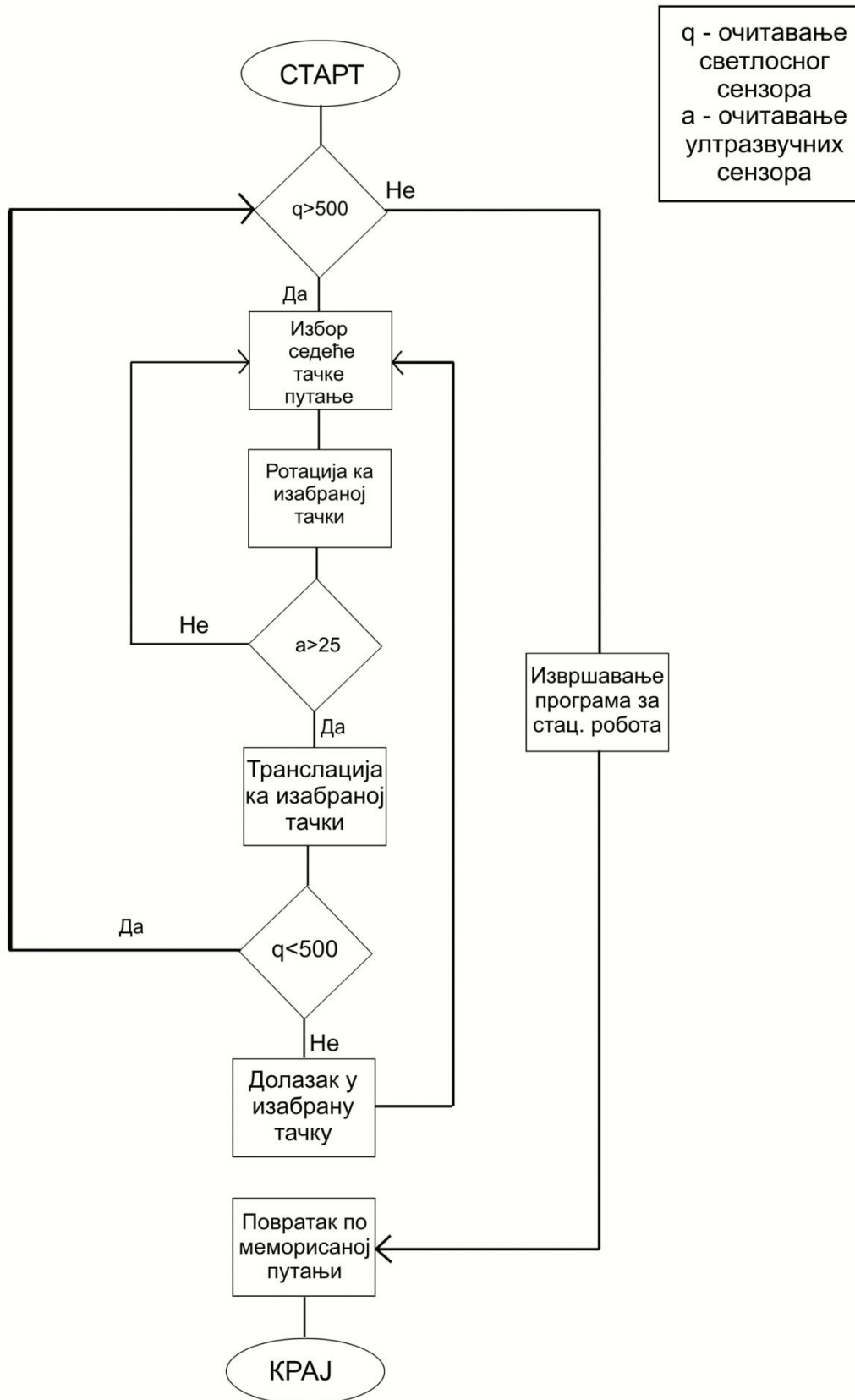
Шк. год.

2009/2010

Датум:

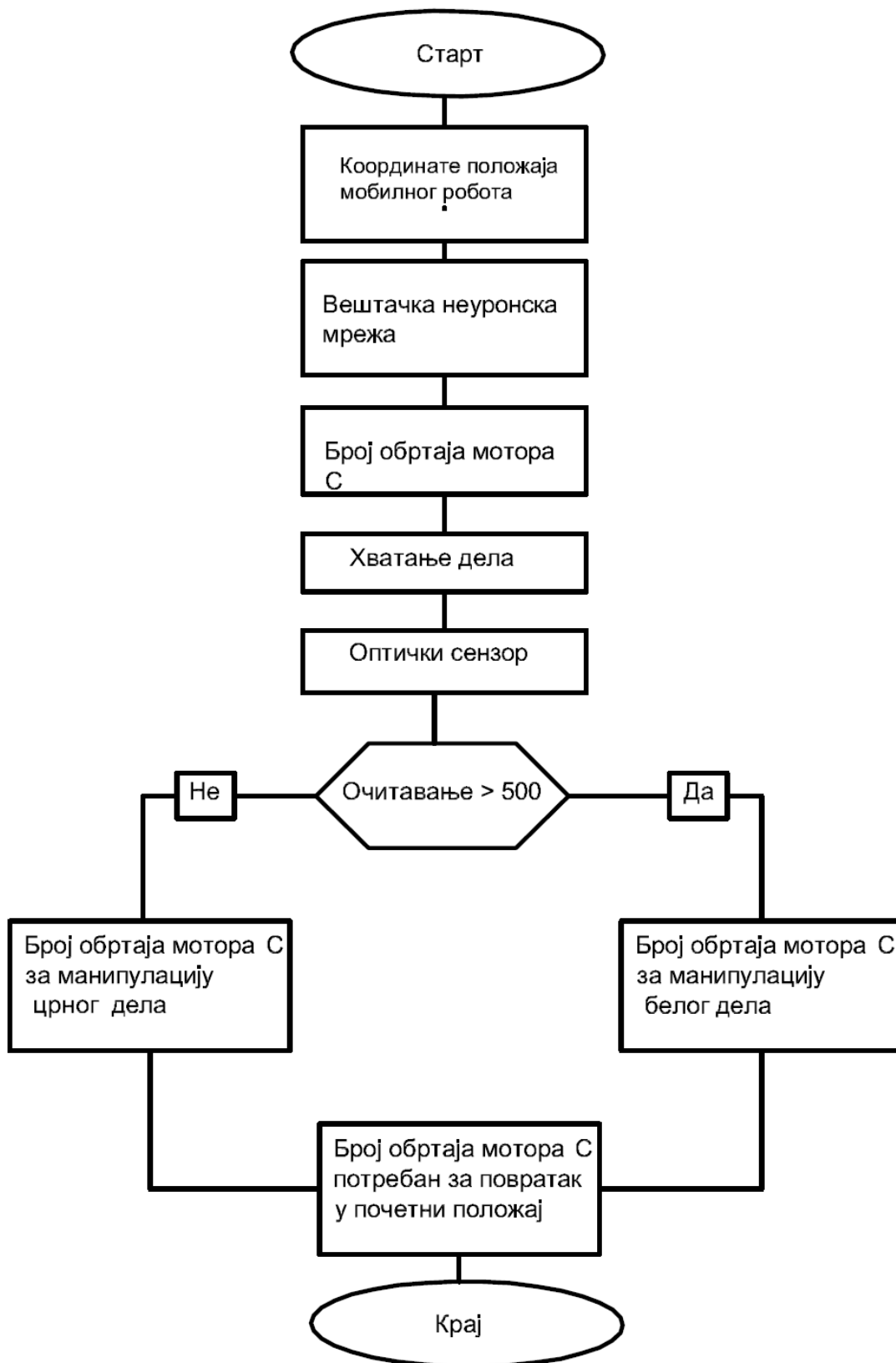
21.06.2010.

Прегледао:



Слика 5.3.1 Алгоритам програма за управљање мобилним роботом

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

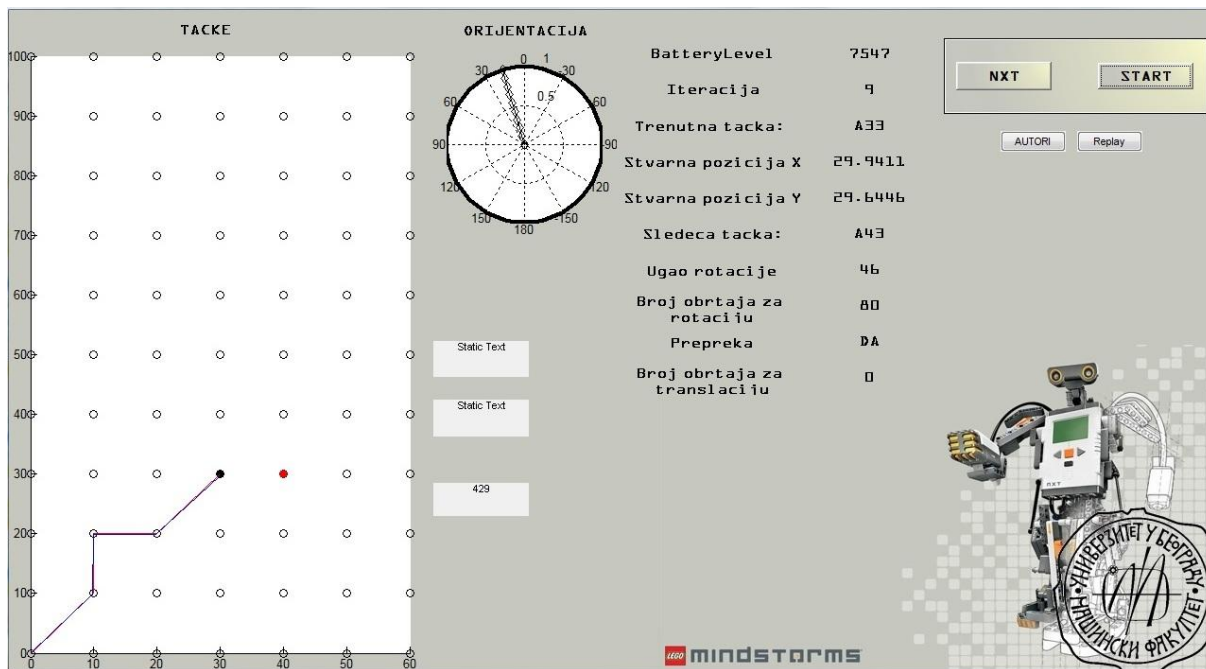


Слика 5.3.2 Алгоритам програма за управљање стационарним роботом

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

Ради лакшег управљања роботима у програмском језику *Matlab* написан је програм са сопственим управљачким интерфејсом на коме се приказују: график остварене и жељене путање са приказом избора следеће тачке, график са приказом оријентације мобилног робота и различити релевантни подаци. Управљачки интерфејс има четири дугмета:

- NXT - за остваривање везе са роботом,
- START - за покретање циклуса преноса једног дела
- Replay - за поновно приказивање путање робота
- AUTORI - за приказивање података о ауторима програма



Слика 5.3.3 Управљачки интерфејс програма



5.4. Комплетан код програма за управљање роботима

```
function varargout = program1(varargin)
% PROGRAM1 M-file for program1.fig
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @program1_OpeningFcn, ...
                  'gui_OutputFcn',  @program1_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

function program1_OpeningFcn(hObject, eventdata, handles, varargin)
axes(handles.axes4)
background=imread('background.jpg');
image(background);
axis off;
guidata(hObject, handles);
dugme = importdata('dugme.jpg')
axes(handles.axes5)
prozor1=imread('prozor1.jpg')
image(prozor1);
axis off;
guidata(hObject, handles);
set(handles.pushbutton1, 'CDATA', dugme);
set(handles.pushbutton2, 'CDATA', dugme);
handles.output = hObject;
guidata(hObject, handles);

function varargout = program1_OutputFcn(hObject, eventdata, handles)
set(gcf, 'Units' , 'Pixels');
set(gcf, 'Position', [10, 50, 1279, 699]);
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
axes(handles.axes1)
connecting=imread('connecting.jpg');
image(connecting);
axis off;
guidata(hObject, handles);
pause(1)
COM_CloseNXT('All');
mobilni = COM_OpenNXTEx('Bluetooth', '001653087014', 'bluetooth.ini');
COM_SetDefaultNXT(mobilni);
bb=NXT_GetBatteryLevel
bb1=num2str(bb);
set(handles.text1, 'String', bb1);
guidata(hObject, handles);
```

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.



```
pause(0.3)
axes(handles.axes1)
connected=imread('connected.jpg');
image(connected);
axis off;
guidata(hObject, handles);
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
load mreza_stacionarni net1
load mreza1 net;
load radijalna;
tacke;
```

подпрограм tacke.m

```
p=10;
q=linspace(0,p*10,p+1);
for i=1:p-3;
    for j=1:p+1;
eval(['A' num2str(i-1) num2str(j-1) '=[q(1,i),q(1,j)]']);
    end;
end;
```

```
S=razdaljina(A00,A66);
```

```
v=1;
```

```
alfa=0;
```

```
t=0;
```

```
TETA=0;
```

```
GAMA=0;
```

```
a1=255;
```

```
a=255;
```

```
q=1000;
```

```
X=0;
```

```
Y=0;
```

```
i=0;
```

```
j=0;
```

```
GG=0;
```

```
b=9.5;
```

```
mB=NXTMotor('B');
```

```
mB.ResetPosition;
```

```
mC=NXTMotor('C');
```

```
mC.ResetPosition;
```

```
trenutna=eval(['A' num2str(i) num2str(j)]);
```

```
E(1,1)=trenutna(1,1);%TACKE za grafik putanje
```

```
E(1,2)=trenutna(1,2);
```

```
D(1,1)=trenutna(1,1);
```

```
D(1,2)=trenutna(1,2);
```

```
%+++++++NXT1+++++++
```

```
while q>400
```

```
    t=t+1
```

```
    txt2 = int2str(t)
```

```
set(handles.text2,'String',txt2);%ITERACIJA
```

```
guidata(hObject, handles);
```

```
txt3=(['A' int2str(trenutna(1,1)/10) int2str(trenutna(1,2)/10)])
```

```
    set(handles.text3,'String',txt3);%TRENUTNA
```

```
guidata(hObject, handles);
```

```
    D(t,1)=trenutna(1,1);%TACKE za grafik putanje
```

```
    D(t,2)=trenutna(1,2);
```

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



%=====IZBOR_TACKE=====

okolne tackel

функција okolne_tackel.m

```
[ c1,c2,c3,c4,c5,c6,c7,c8,l1,l2,l3,l4,l5,l6,l7,l8 ] = komsije(trenutna)
P1=eval(['A' num2str(c1) num2str(l1)]);
P2=eval(['A' num2str(c2) num2str(l2)]);
P3=eval(['A' num2str(c3) num2str(l3)]);
P4=eval(['A' num2str(c4) num2str(l4)]);
P5=eval(['A' num2str(c5) num2str(l5)]);
P6=eval(['A' num2str(c6) num2str(l6)]);
P7=eval(['A' num2str(c7) num2str(l7)]);
P8=eval(['A' num2str(c8) num2str(l8)]);
```

```
ww(1,1)=trenutna(1,1)/10
ww(2,1)=trenutna(1,2)/10
GG=sim(radijalna,ww)
sledeca=eval(['P' num2str(GG(v))]);
txt6=(['A' num2str(sledeca(1,1)/10) num2str(sledeca(1,2)/10)]%sledeca
pov1=eval(['A' num2str(trenutna(1,1)/10) num2str(trenutna(1,2)/10)]
povratak(t,1)=pov1(1,1);
povratak(t,2)=pov1(1,2);
    set(handles.text6,'String',txt6);
guidata(hObject, handles);
```

%=====GRAFIK1=====

```
txt33=(['TACKE']);
set(handles.text33,'String',txt33);
guidata(hObject, handles);
axes(handles.axes1)
grafik3
```

подпрограма grafik3.m

```
p=10;
for i=1:p-3;
    for j=1:p+1;
        rx=eval(['A' num2str(i-1) num2str(j-1)]);
scatter(rx(1,1),rx(1,2),'ok'); hold on;
    end;
end;
p2 = scatter(trenutna(1,1),trenutna(1,2),'ok','filled'); hold on;
p3 = scatter(sledeca(1,1),sledeca(1,2),'or','filled'); hold on;
plot(D(:,1),D(:,2),'r');hold on;
plot(E(:,1),E(:,2),'b');hold off;
put(t)=getframe;
xlabel(' ');
ylabel(' ');
```

```
guidata(hObject, handles);
```

%=====UGAO=====

```
mB.ResetPosition;
mC.ResetPosition;
ugao
```

подпрограма ugao.m

```
t1=trenutna(1,1)/10;
t2=(trenutna(1,2)/10)+1;
tackica=eval(['A' num2str(t1) num2str(t2)]);
kat1=razdaljina(trenutna,tackica);
kat2=razdaljina(trenutna,sledeca);
hipo=razdaljina(tackica,sledeca);
alfa=rad2deg(acos((kat1^2+kat2^2-hipo^2)/(2*kat1*kat2)));
if trenutna(1,1)<sledeca(1,1)
    alfa=-alfa;
else
    alfa=alfa;
end
```

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



```
функција razdaljina. М
function [ d ] = razdaljina( x,y )
d=sqrt((x(1,1)-y(1,1))^2+(x(1,2)-y(1,2))^2);
end

alfa1=round(alfa-TETA)
%-----
tahoo=abs(round((9.5*deg2rad(alfa1)*180)/(2*pi*2.75))+1);
%-----
if alfa1<0
    powerC=-15;
    powerB=-powerC;
elseif alfa1>0
    powerC=15;
    powerB=-powerC;
else alfa1==0
    powerC=0;
    powerB=0;
end

mC = NXTMotor('C', 'Power',powerC, 'TachoLimit', tahoo);
mB = NXTMotor('B', 'Power',powerB, 'TachoLimit', tahoo);
mB.SendToNXT();
mC.SendToNXT();
mB.WaitFor();
mC.WaitFor();

%-----
txt7=num2str(alfa1)
    set(handles.text7,'String',txt7);
guidata(hObject, handles);
txt8=num2str(tahoo)
    set(handles.text8,'String',txt8);
guidata(hObject, handles);
%=====STVARNA_POZ=====
ugaoB=mB.ReadFromNXT();
    l=(ugaoB.TachoCount);
ugaoC=mC.ReadFromNXT();
    d=(ugaoC.TachoCount);
stvarna poz
подпрограм stvarna_poz.m
S1=(l*3.14/180)*2.75;
Sd=(d*3.14/180)*2.75;
b=9.5;
GAMA=GAMA+rad2deg((Sd-S1)/b);
X=X-(((Sd+S1)/2)*sind(GAMA));
Y=Y+(((Sd+S1)/2)*cosd(GAMA));
E(t,1)=X;
E(t,2)=Y;
%=====
TETA=orijentacija(GAMA)
Функција orijentacija.m
function [ TETA ] = orijentacija( GAMA )
e1=180;
f1=abs(GAMA);
a1=(mod(f1,e1));
if GAMA>0;
    a1=a1;
else
    a1=-a1;
end;
TETA=a1;
if TETA>180;
    TETA=-180-(TETA-180);
```

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



```
elseif TETA < (-180);  
    TETA = 180 + (TETA + 180);  
else  
    TETA = TETA;  
end;  
end
```

```
%=====
```

```
mB.ResetPosition;  
mC.ResetPosition;  
txt4 = num2str(E(t,1))  
    set(handles.text4, 'String', txt4); %X  
guidata(hObject, handles);  
txt5 = num2str(E(t,2))  
    set(handles.text5, 'String', txt5); %Y  
guidata(hObject, handles);  
%=====GRAFIK2=====
```

```
txt34 = (['ORIJENTACIJA']);  
set(handles.text34, 'String', txt34);  
guidata(hObject, handles);  
axes(handles.axes2)  
grafik2
```

```
подпрограм grafik2.m  
for h=1:10;  
    FI(1,h) = TETA + pi/2;  
    r = linspace(0,1,10);  
end;  
polarLabels(FI, r, '-kd', -90, 0);  
ph = findall(gca, 'type', 'patch');  
    set(ph, 'facecolor', [1,1,1], 'edgecolor', [0,0,0], 'linewidth', 3);  
guidata(hObject, handles);  
%=====TRANSLACIJA1 (ima PREPREKE)=====
```

```
OpenUltrasonic(SENSOR_2)  
a = GetUltrasonic(SENSOR_2);  
CloseSensor(SENSOR_2);  
OpenUltrasonic(SENSOR_3)  
a1 = GetUltrasonic(SENSOR_3);  
CloseSensor(SENSOR_3);  
if (a < 20) | (a1 < 20)  
    txt9 = (['DA']);  
set(handles.text9, 'String', txt9);  
guidata(hObject, handles);  
    txt10 = (['0']);  
set(handles.text10, 'String', txt10);  
guidata(hObject, handles);  
v = v + 1;  
trenutna = trenutna;  
%=====TRANSLACIJA1 (nema PREPREKE)=====
```

```
else  
    if trenutna(1,2)/10 < 6  
        v = 1;  
    elseif trenutna(1,2)/10 == 6  
        v = 2;  
    elseif trenutna(1,2)/10 > 6  
        v = 6;  
    end  
    txt9 = (['NE']);  
set(handles.text9, 'String', txt9);  
guidata(hObject, handles);  
%=====
```

```
SP(1,1) = E(t,1);  
SP(1,2) = E(t,2);
```

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.



```
u=razdaljina(SP,sledeca);
    taho=abs(round(sim(net,u)));
    txt10=num2str(taho);
    set(handles.text10,'String',txt10);
    guidata(hObject, handles);
OpenLight(SENSOR_4, 'ACTIVE');
    q=GetLight(SENSOR_4);
    mBC = NXTMotor('BC', 'Power',30, 'TachoLimit', taho);
    mBC.SendToNXT();
    for e=1:30
        pause(0.1)
        q=GetLight(SENSOR_4);
        txt39=num2str(q)
        set(handles.text39,'String',txt39);
        guidata(hObject, handles);
        if q<400
            mBC.Stop('off')
            CloseSensor(SENSOR_4);
            break;end
    end
        CloseSensor(SENSOR_4);
    mB.ResetPosition;
    mC.ResetPosition;
    mB.WaitFor();
    mC.WaitFor();
    v=1
    clear GG

=====STVARNA_POZ-2=====
ugaoB=mB.ReadFromNXT();
    l=(ugaoB.TachoCount);
ugaoC=mC.ReadFromNXT();
    d=(ugaoC.TachoCount);
stvarna_poz1
подпрограм stvarna_poz1.m
S1=(l*3.14/180)*2.75;
    Sd=(d*3.14/180)*2.75;
    b=9.5;
    GAMA=GAMA+rad2deg((Sd-S1)/b);
    X=X-(((Sd+S1)/2)*sind(GAMA));
    Y=Y+(((Sd+S1)/2)*cosd(GAMA));
    E(t+1,1)=X;
    E(t+1,2)=Y;
mB.ResetPosition;
mC.ResetPosition;
txt4 = num2str(E(t+1,1))
    set(handles.text4,'String',txt4);%X
guidata(hObject, handles);
txt5 = num2str(E(t+1,2))
    set(handles.text5,'String',txt5);%Y
guidata(hObject, handles);
=====
SP(1,1)=E(t+1,1);
SP(1,2)=E(t+1,2);
trenutna=sledeca;
pov1=eval(['A' num2str(trenutna(1,1)/10) num2str(trenutna(1,2)/10)])
povratak(t+1,1)=pov1(1,1);
povratak(t+1,2)=pov1(1,2);
    end
end
```



```
%=====KOORDINATE ZA NXT2=====
D(t,1)=trenutna(1,1);
D(t,2)=trenutna(1,2);
SP(1,1)=E(t+1,1);
SP(1,2)=E(t+1,2);
txt4 = num2str(E(t+1,1))
    set(handles.text4,'String',txt4);%X
guidata(hObject, handles);
txt5 = num2str(E(t+1,2))
    set(handles.text5,'String',txt5);%Y
guidata(hObject, handles);
ZETA=180-GAMA-90;
ZETA1=deg2rad(ZETA);
X1=round((60-SP(1,1))*10)
txt37=(['X=' num2str(X1)])
    set(handles.text37,'String',txt37);
    guidata(hObject, handles);
if SP(1,2)>=60
Y1=round((60-SP(1,2))*10)
elseif SP(1,2)<60
    Y1=round((- (SP(1,2)-60))*10)
end
txt38=(['Y=' num2str(Y1)])
set(handles.text38,'String',txt38);
    guidata(hObject, handles);
%=====GRAFIK PP=====
axes(handles.axes1)
plot(D(:,1),D(:,2),'r');hold on;
plot(E(:,1),E(:,2),'b');hold off;
put(t+1)=getframe;
guidata(hObject, handles);
save put put;
%=====NXT2=====
COM_CloseNXT('All');
industr = COM_OpenNXTEx('Bluetooth','00165303EFC8','bluetooth1.ini');
COM_SetDefaultNXT(industr);
p=[X1;Y1];           %pozicija mobilnog robota%
EER=sim(net1,p)%broj obrtaja motora C%
T1=EER*950;
T2=EER*1000;
kp=8;               %korekcija snage%
kt=6;               %korekcija obrtaja%
if Y1 > 0;
    powerC=100;
    tahoC=round(T1);
    powerB=-round(powerC/kp);
    tahoB=round(tahoC/kt);
    mC = NXTMotor('C', 'Power', powerC, 'TachoLimit', tahoC);
    mB = NXTMotor('B', 'Power', powerB, 'TachoLimit', tahoB);
    mC.SendToNXT();
    mB.SendToNXT();
    mB.WaitFor();
    mC.WaitFor();
    ss=mC.ReadFromNXT();
    d=abs(ss.Position)
    mB = NXTMotor('B', 'Power', 80, 'TachoLimit', 3000);
    mB.SendToNXT();
    mB.WaitFor();
    mA = NXTMotor('A', 'Power', 15, 'TachoLimit', 60);
    mA.SendToNXT();
    mA.WaitFor();
```

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.



```
OpenLight (SENSOR_1, 'ACTIVE');
if GetLight (SENSOR_1) > 500
    B=GetLight (SENSOR_1)
    CloseSensor (SENSOR_1);
    mB = NXTMotor('B', 'Power', -80, 'TachoLimit', 3000);
    mB.SendToNXT();
    mB.WaitFor();
    pC=100;
    TCb=round(15000-d);
    pB=-round(pC/kp);
    TBb=round(TCb/kt);
    mC = NXTMotor('C', 'Power', pC, 'TachoLimit', TCb);
    mB = NXTMotor('B', 'Power', pB, 'TachoLimit', TBb);
    mC.SendToNXT();
    mB.SendToNXT();
    mB.WaitFor();
    mC.WaitFor();
    mA = NXTMotor('A', 'Power', -15, 'TachoLimit', 60);
    mA.SendToNXT();
    mA.WaitFor();
    pb=-round(-100/kp);
    tb=round(15000/kt);
    mC = NXTMotor('C', 'Power', -100, 'TachoLimit', 15000);
    mB = NXTMotor('B', 'Power', pb, 'TachoLimit', tb);
    mC.SendToNXT();
    mB.SendToNXT();
else
    B=GetLight (SENSOR_1)
    CloseSensor (SENSOR_1);
    mB = NXTMotor('B', 'Power', -80, 'TachoLimit', 3000);
    mB.SendToNXT();
    mB.WaitFor();
    pC=-100;
    TCc=round(15000+d);
    pB=-round(pC/kp);
    TBc=round(TCc/kt);
    mC = NXTMotor('C', 'Power', pC, 'TachoLimit', TCc);
    mB = NXTMotor('B', 'Power', pB, 'TachoLimit', TBc);
    mC.SendToNXT();
    mB.SendToNXT();
    mB.WaitFor();
    mC.WaitFor();
    mA = NXTMotor('A', 'Power', -15, 'TachoLimit', 60);
    mA.SendToNXT();
    mA.WaitFor();
    pb=-round(100/kp);
    tb=round(15000/kt);
    mC = NXTMotor('C', 'Power', 100, 'TachoLimit', 15000);
    mB = NXTMotor('B', 'Power', pb, 'TachoLimit', tb);
    mC.SendToNXT();
    mB.SendToNXT();
end
else
    powerC=-100;
    tahoC=round(T2);
    powerB=-round(powerC/kp);
    tahoB=round(tahoC/kt);
    mC = NXTMotor('C', 'Power', powerC, 'TachoLimit', tahoC);
    mB = NXTMotor('B', 'Power', powerB, 'TachoLimit', tahoB);
    mC.SendToNXT();
    mB.SendToNXT();
```

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.



```
mB.WaitFor();
mC.WaitFor();
ss=mC.ReadFromNXT();
d=abs(ss.Position)
mB = NXTMotor('B', 'Power', 80, 'TachoLimit', 3000);
mB.SendToNXT();
mB.WaitFor();
mA = NXTMotor('A', 'Power', 15, 'TachoLimit', 60);
mA.SendToNXT();
mA.WaitFor();
OpenLight(SENSOR_1, 'ACTIVE');
if GetLight(SENSOR_1) > 500
    B=GetLight(SENSOR_1)
    CloseSensor(SENSOR_1);
    mB = NXTMotor('B', 'Power', -80, 'TachoLimit', 3000);
    mB.SendToNXT();
    mB.WaitFor();
    pC=100;
    TCb=round(15000+d);
    pB=-round(pC/kp);
    TBb=round(TCb/kt);
    mC = NXTMotor('C', 'Power', pC, 'TachoLimit', TCb);
    mB = NXTMotor('B', 'Power', pB, 'TachoLimit', TBb);
    mC.SendToNXT();
    mB.SendToNXT();
    mB.WaitFor();
    mC.WaitFor();
    mA = NXTMotor('A', 'Power', -15, 'TachoLimit', 60);
    mA.SendToNXT();
    mA.WaitFor();
    pb=-round(-100/kp);
    tb=round(15000/kt);
    mC = NXTMotor('C', 'Power', -100, 'TachoLimit', 15000);
    mB = NXTMotor('B', 'Power', pb, 'TachoLimit', tb);
    mC.SendToNXT();
    mB.SendToNXT();
else
    B=GetLight(SENSOR_1)
    CloseSensor(SENSOR_1);
    mB = NXTMotor('B', 'Power', -80, 'TachoLimit', 3000);
    mB.SendToNXT();
    mB.WaitFor();
    pC=-100;
    TCc=round(15000-d);
    pB=-round(pC/kp);
    TBc=round(TCc/kt);
    mC = NXTMotor('C', 'Power', pC, 'TachoLimit', TCc);
    mB = NXTMotor('B', 'Power', pB, 'TachoLimit', TBc);
    mC.SendToNXT();
    mB.SendToNXT();
    mB.WaitFor();
    mC.WaitFor();
    mA = NXTMotor('A', 'Power', -15, 'TachoLimit', 60);
    mA.SendToNXT();
    mA.WaitFor();
    pb=-round(100/kp);
    tb=round(15000/kt);
    mC = NXTMotor('C', 'Power', 100, 'TachoLimit', 15000);
    mB = NXTMotor('B', 'Power', pb, 'TachoLimit', tb);
    mC.SendToNXT();
    mB.SendToNXT();
```

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



```
end
end
%=====POVRATAK+NXT1=====
COM_CloseNXT('all')
mobilni = COM_OpenNXTEx('Bluetooth','001653087014','bluetooth.ini');
COM_SetDefaultNXT(mobilni);
NAZ=razdaljina(SP,A00);
while NAZ>5
    t=t-1;
    sledeca(1,1)=povratak(t,1);
    sledeca(1,2)=povratak(t,2);
%=====GRAFIK1=====
txt33=(['TACKE']);
set(handles.text33,'String',txt33);
guidata(hObject, handles);
axes(handles.axes1)
grafik3
guidata(hObject, handles);
%=====UGAO=====
mB.ResetPosition;
mC.ResetPosition;
ugao
alfa1=round(alfa-TETA)
%-----
tahoo=abs(round((9.5*deg2rad(alfa1)*180)/(2*pi*2.75))+1);
%-----
if alfa1<0
    powerC=-15;
    powerB=-powerC;
elseif alfa1>0
    powerC=15;
    powerB=-powerC;
else alfa1==0
    powerC=0;
    powerB=0;
end
mC = NXTMotor('C', 'Power',powerC, 'TachoLimit', tahoo);
mB = NXTMotor('B', 'Power',powerB, 'TachoLimit', tahoo);
mB.SendToNXT();
mC.SendToNXT();
mB.WaitFor();
mC.WaitFor();
%-----
txt7=num2str(alfa1)
set(handles.text7,'String',txt7);
guidata(hObject, handles);
txt8=num2str(tahoo)
set(handles.text8,'String',txt8);
guidata(hObject, handles);
%=====STVARNA_POZ=====
ugaoB=mB.ReadFromNXT();
l=(ugaoB.TachoCount);
ugaoC=mC.ReadFromNXT();
d=(ugaoC.TachoCount);
stvarna_poz
%=====
TETA=orientacija(GAMA)
%=====
mB.ResetPosition;
mC.ResetPosition;
txt4 = num2str(E(t,1))
```

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.



```
set(handles.text4,'String',txt4);%X
guidata(hObject, handles);
txt5 = num2str(E(t,2))
set(handles.text5,'String',txt5);%Y
guidata(hObject, handles);
%=====GRAFIK2=====
txt34=(['ORIЈENTACIЈA']);
set(handles.text34,'String',txt34);
guidata(hObject, handles);
axes(handles.axes2)
grafik2
guidata(hObject, handles);
%=====
SP(1,1)=E(t,1);
SP(1,2)=E(t,2);
NAZ=razdaljina(SP,A00);
u=razdaljina(SP,sledeca);
    taho=round(sim(net,u));
        txt10=num2str(taho);
    set(handles.text10,'String',txt10);
    guidata(hObject, handles);
    mBC = NXTMotor('BC', 'Power',30, 'TachoLimit', taho);
        mBC.SendToNXT();
        mB.ResetPosition;
        mC.ResetPosition;
        mB.WaitFor();
        mC.WaitFor();
%=====STVARNA_POZ-2=====
ugaoB=mB.ReadFromNXT();
    l=(ugaoB.TachoCount);
ugaoC=mC.ReadFromNXT();
    d=(ugaoC.TachoCount);
stvarna_poz1
mB.ResetPosition;
mC.ResetPosition;
txt4 = num2str(E(t+1,1))
    set(handles.text4,'String',txt4);%X
guidata(hObject, handles);
txt5 = num2str(E(t+1,2))
    set(handles.text5,'String',txt5);%Y
guidata(hObject, handles);
%=====
SP(1,1)=E(t+1,1);
SP(1,2)=E(t+1,2);
NAZ=razdaljina(SP,A00);
trenutna=sledeca;
end
%=====
mB.ResetPosition;
mC.ResetPosition;
ugao
alfa1=round(TETA)
%-----
tahoo=abs(round((9.5*deg2rad(alfa1)*180)/(2*pi*2.75))+1);
%-----
if alfa1<0
    powerC=-15;
    powerB=-powerC;
elseif alfa1>0
    powerC=15;
    powerB=-powerC;
```

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.



```
else alfa1==0
    powerC=0;
    powerB=0;
end
mC = NXTMotor('C', 'Power',powerC, 'TachoLimit', tahoo);
mB = NXTMotor('B', 'Power',powerB, 'TachoLimit', tahoo);
mB.SendToNXT();
mC.SendToNXT();
mB.WaitFor();
mC.WaitFor();

function pushbutton4_Callback(hObject, eventdata, handles)
load put
axes(handles.axes1)
movie(put,1,1);
axis off;
cla
guidata(hObject, handles);

function pushbutton5_Callback(hObject, eventdata, handles)
autori();
```

```
подпрограма autori.m
function varargout = autori(varargin)
% AUTORI M-file for autori.fig
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @autori_OpeningFcn, ...
                  'gui_OutputFcn',  @autori_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

function autori_OpeningFcn(hObject, eventdata, handles, varargin)
load autori.mat
load muzika
axes(handles.axes1)
wavplay(muzika,fs,'async')
movie(autori,-50,35)
axis off;
guidata(hObject, handles);
delete(gcf)
handles.output = hObject;
guidata(hObject, handles);

function varargout = autori_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
```

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.

6. Дистрибуирана симулација подсистема подсистема унутрашњег транспорта у дефинисаном радном простору датог модела технолошког система, базирана на примени AnyLogic софтвера

Технолошки систем који симулирамо састоји се од транспортног подсистема и обрадног подсистема. Транспортни подсистем се састоји од два робота, мобилног робота (транспортера) и стационарног робота (манипулатора). Обрадни подсистем се састоји од три машине које обрађују два производа, и то једна машина обрађује један производ а две машине обрађују други.

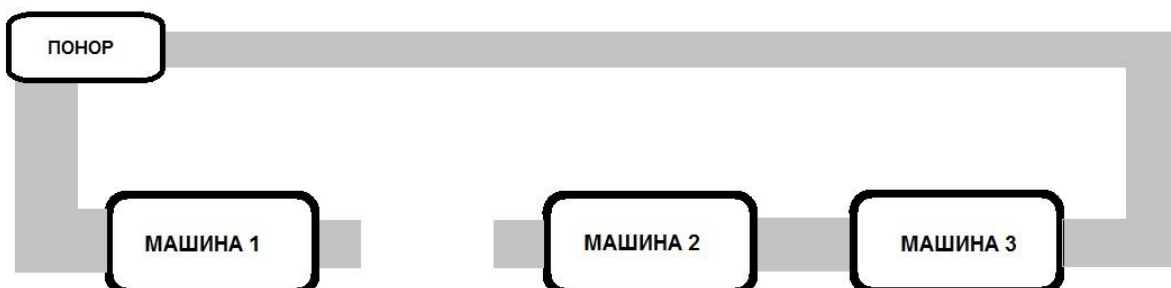
Одредити време потребно за операцију транспорта и повратка мобилног робота, као и време потребно за хатање, манипуацију и остављање дела стационарног робота. Које ће задовољити потребе обрадног подсистема да све машине раде са искоришћеношћу од преко седамдесет процента. Ако делови долазе до мобилног робота са два улаза и подједнаким интервалом долажења од 25 s.

Табела 6 Интервали времена потребни за обраду делова на машинама

Интервал времена	Машине обрадног система		
	M1 [s]	M2 [s]	M3 [s]
	25 ± 5	20 ± 5	15 ± 5

6.1. Опис обрадног подсистема

Обрадни подсистем технолошког система садржи три машине и пет тракастих транспортера, машине су постављене лево и десно од манипулатора. Десно од манипулатора се налази машина M1 која обрађује беле делове и она врши само грубу обраду на тим деловима. Лево од манипулатора се налазе машине M2 и M3 које обрађују црне делове, машина M2 врши грубу а машина M3 фину обраду на црним деловима. Делови се у обрадном систему преносе тракастим транспортерима.



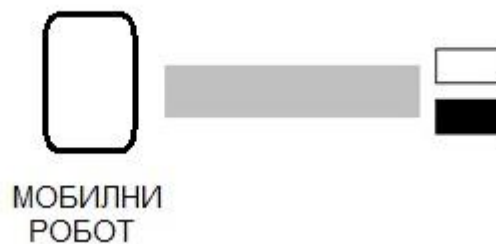
Слика 6.1 Шема обрадног подсистема

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

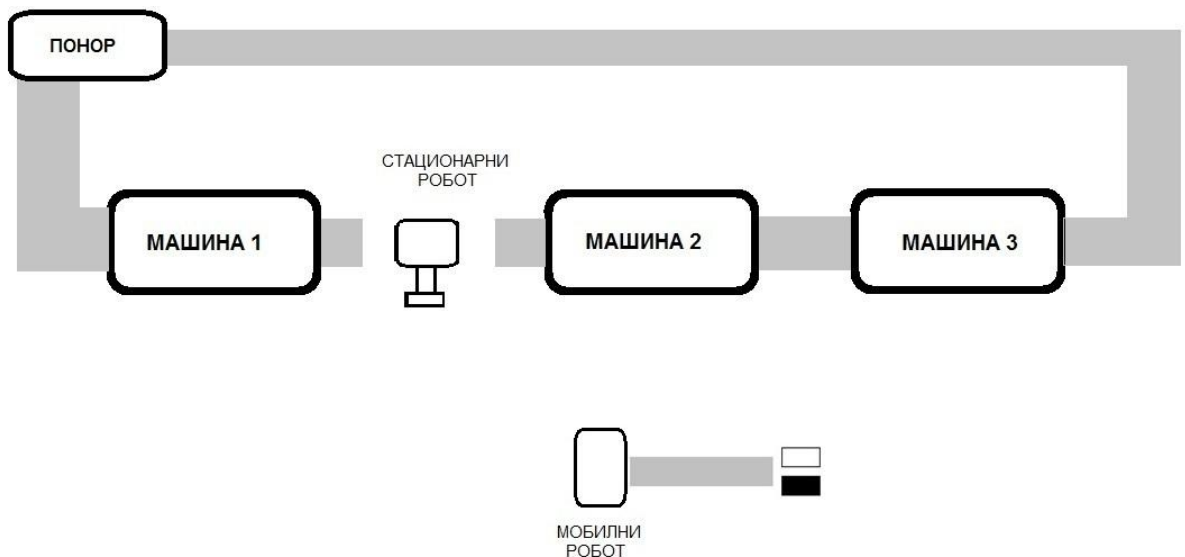
6.2. Опис транспортног подсистема

Транспортни подсистем обрадног система садржи два робота, мобилног и стационарног. Конфигурација мобилног робота је таква да у једном циклусу може да понесе само један део. Мобилни робот је опремљен ултразвучним и оптичким сензорима који му омогућавају сналажење у динамичком окружењу овог система.

Стационарни робот врши манипулацију делова као и њихово сортирање. Опремљен је светлосним сензором који му омогућава препознавање делова.

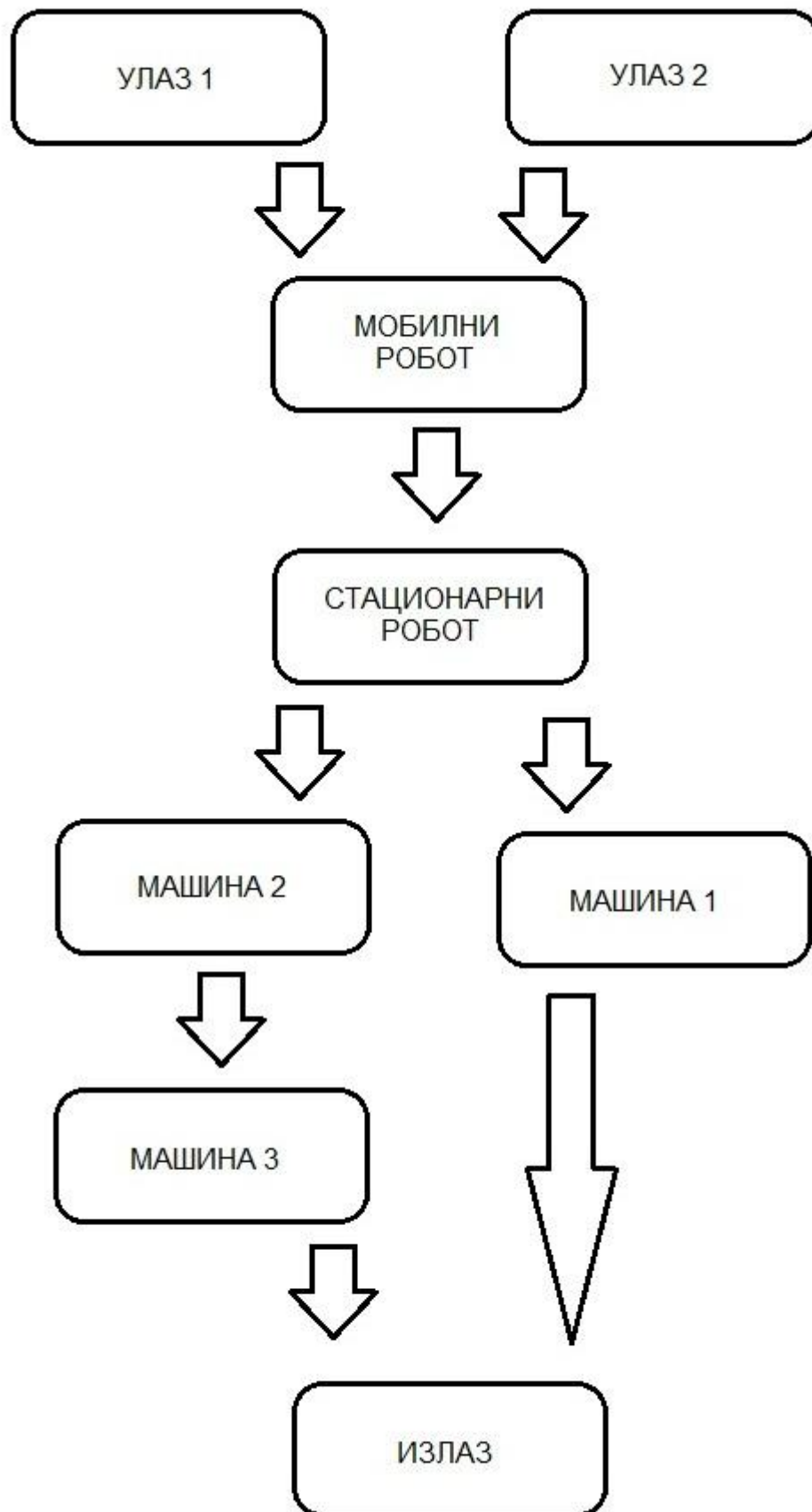


Слика 6.2 Шема транспортног подсистема



Слика 6.2.1 Шема целокупног технолошког система

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



Слика 6.2.2 Блок дијаграм симулационог система

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

6.3. Симулациони модел система у *AnyLogic* софтверском пакету

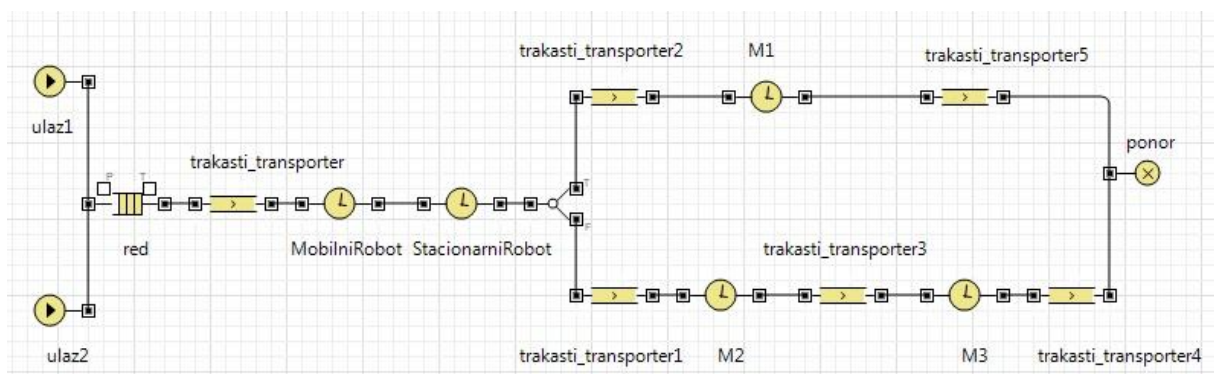
AnyLogic је софтверски пакет опште намене за симулацију и моделовање дискретних, континуалних и хибридних система. Његова област примене обухвата :

- Системе за контролу саобраћаја
- Производне системе
- Логистику
- Рачунарске системе
- Системе образовања итд.

AnyLogic омогућава симулацију рада система у реалном времену као и креирање анимације тог система.

Модел технолошког система

Елементи технолошког система су у симулацији приказани као елементи из *Enterprise Library* палете. Симулирање рада сваке машине и робота врши се елементима који њихов рад приказују помоћу временских интервала снимљених током њиховог рада у реалним условима.



Слика 6.3 Симулациони модел технолошког система

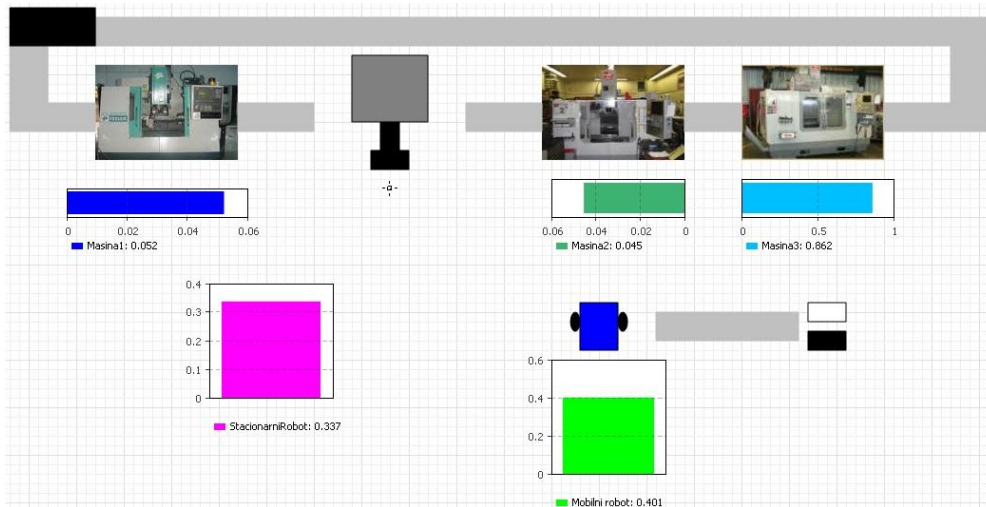
Обрада на машинама се представља као униформна расподела интервала времена обраде и то :

$M1 = \text{uniform}(20,30)$

$M2 = \text{uniform}(15,25)$

$M3 = \text{uniform}(10,20)$

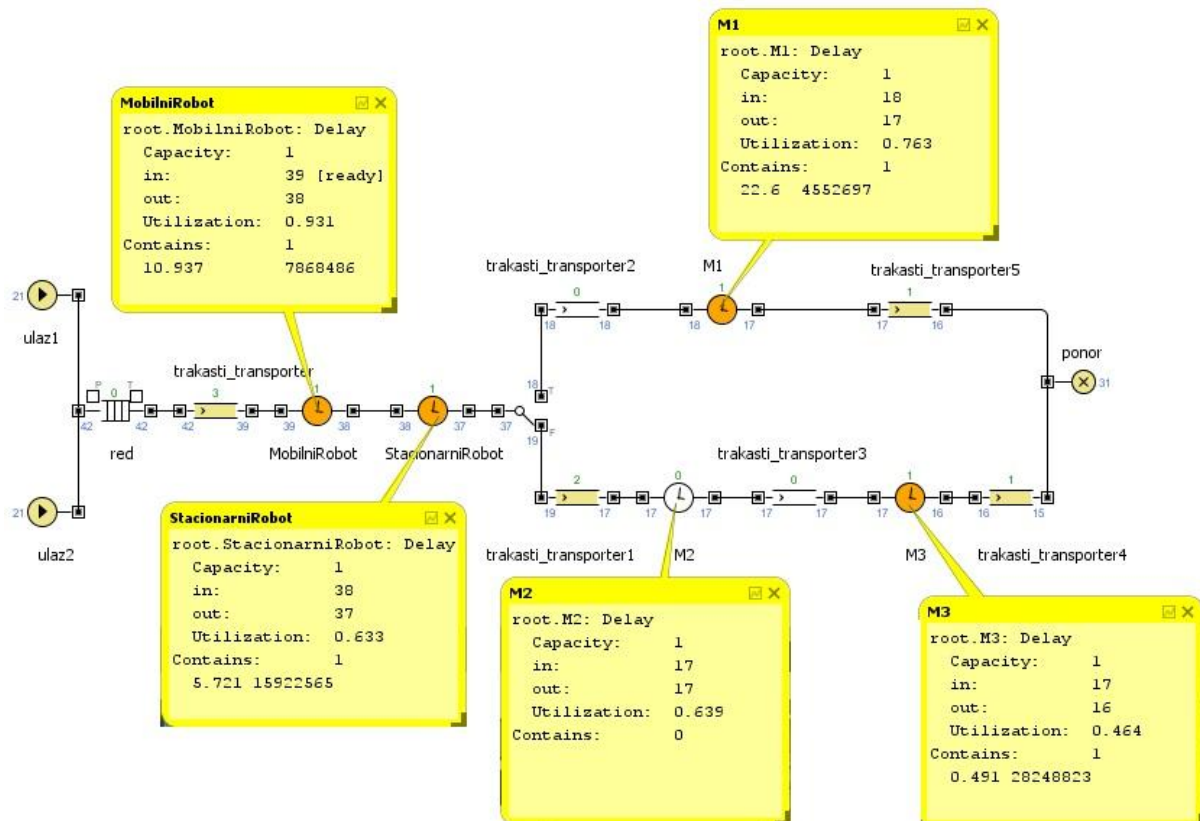
За овако изабране интервале времена обраде и интервале долажења делова, вршимо варирање интервала времена потребних за извршавање циклуса робота док не добијемо задовољавајућу искоришћеност машина.



Слика 6.3.1 Приказ анимираног модела технолошког система

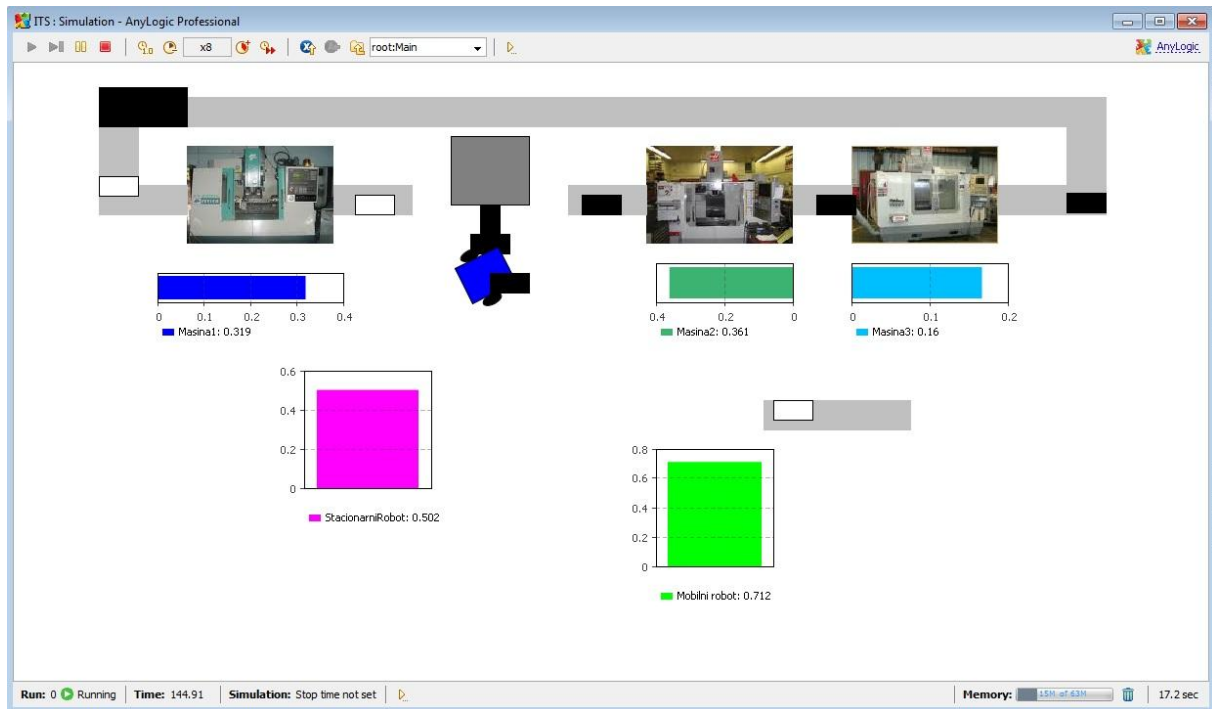
Извршење симулације

Након подешавања свих параметара модела симулација се пушта у рад и врши се снимање постигнутих резултата.



Слика 6.3.2 Извршење симулације, симулациони модел

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



Слика 6.3.3 Извршење симулације, анимациони модел

Анализа резултата

Табела 6.3 Табела анализе резултата

Време операције мобилног робота [s]	Време операције стационарног робота [s]	Време рада система до преоптерећења [s]	Искоришћеност машине М1 [%]	Искоришћеност машине М2 [%]	Искоришћеност машине М3 [%]
25 ± 5	15 ± 5	3796,88	50,1	39,4	29
17.5 ± 2.5	12.5 ± 2.5	6576,56	70,1	55,6	42
12.5 ± 2.5	10 ± 2	166,85	42,9	34,9	17
12.5 ± 2.5	9 ± 1	Неограничено	98,5	80,1	60

Испитивањем модела технолошког система са параметрима задатим у **Табели 6** дошли смо до резултата приказаних у **Табели 6.3** Анализом тих резултат утврђено је да захтевана искоришћеност машина не може бити постигнута, најближа достигнута искоришћеност са реалним трајањем циклуса робота је она у последњем случају из **Табеле 6.3** Ако то не задовољава потребе произвођача могућа су следећа решења :

1. Скраћење времена обраде на машини М2
2. Увођење још једног мобилног робота
3. Повећање капацитета мобилног роота
4. Уградња још једног транспортног подсистема

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

2009/2010

21.06.2010.

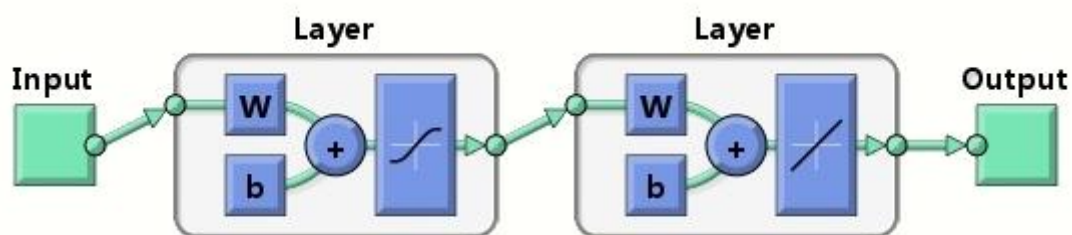
7. Машинско учење мобилног робота компаније LEGO – LEGO MINDSTORMS NXT у циљу едукације, тестирања и верификације предложеног концепта

У предложеном концепту роботизованог унутрашњег транспорта, манипулације и класификације делова у интелигентном технолошком систему примењене су 3 вештачке неуронске мреже у функцији одлучивања и одређивања параметара управљања, на основу утицаја променљивих фактора околине.

Две мреже су примењене код мобилног робота: мрежа која обавља избор следеће тачке трајекторије мобилног робота и мрежа која одређује број обртаја мотора потребан за транслацију робота до следеће тачке трајекторије. Код стационарног робота – манипулатора примењена је једна вештачка неуронска мрежа која одређује број обртаја мотора потребан да робот, ротацијом у бази, заузме положај за препознавање транспортованог објекта и да га преузме од мобилног робота.

Мреже примењене за одређивање броја обртаја мотора су типа *Feed Forward Backpropagation* и њихов задатак је да помоћу обучавајућих параметара „науче“ зависност између растојања и броја обртаја мотора потребних за његово прелажење тј. да изврше „уклапање криве“ (*Curve fitting*) математичких модела кинематике робота. *Backpropagation* мреже су веома погодне за овакве задатке и са релативно малим бројем обучавајућих параметара дају задовољавајуће резултате.

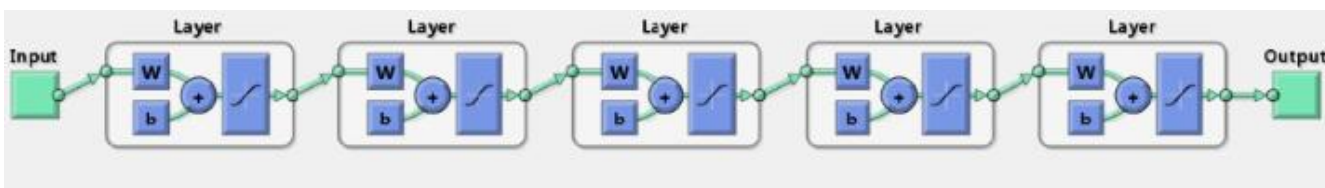
Мрежа за одређивање броја обртаја мотора мобилног робота има 2 скривена слоја са по 20 неурона. Улаз је једнокомпонентни вектор који представља раздаљину коју је потребно да робот пређе. Излаз је, такође, једнокомпонентни вектор чија вредност представља укупни угао за који је потребно да се вратило мотора окрене да би робот прешао улазну раздаљину. Активациона функција првог скривеног слоја је сигмоидна функција, а другог скривеног слоја је линеарна активациона функција.



Слика 7 Мрежа за одређивање броја обртаја мотора мобилног робота

Мрежа за број обртаја мотора у бази стационарног робота има 4 скривена слоја, прва два са по 8 неурона, а друга два са по 6 неурона. Ова неуронска мрежа успоставља функционалну везу између координата (x,y) , које описују положај мобилног робота у равни у координатном систему стационарног робота и потребног угла, односно броја обртаја вратила којим се роботска рука позиционира тачно на место изнад датих координата. На овај начин се решава један прост инверзан кинематички проблем. Прихваћена архитектура мреже је 2 [8-8-6-6] 1 која је дала прихватљиве резултате приликом обучавања *Scaled conjugate gradient backpropagation* алгоритмом учења.

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



Слика 7.1 Архитектура мреже за добијање броја обрта мотора базе стационарног робота на основу позиције мобилног

8. Анализа резултата машинског учења мобилног робота *LEGO MINDSTORMS NXT* и његове улоге у оквиру симулираног интелигентног технолошког система

Први технички систем је једноструко преносни, што значи да има идеалну улазну и једну излазну величину. У питању је функционална веза између раздаљине које пређе мобилни робот и угла вратила које се при том кретању обрне. Ова веза има врло једноставан аналитички облик, али је ипак моделирана вештачком неуронском мрежом у циљу едукације и експерименталног доказа њене употребљивости. Свих 100 обучавајућих парова су добијени амалитичким путем, а затим се приступило машинском учењу.

Табела 8.1 Улазни вектор

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100					

Табел 8.1.1 Излазни вектор

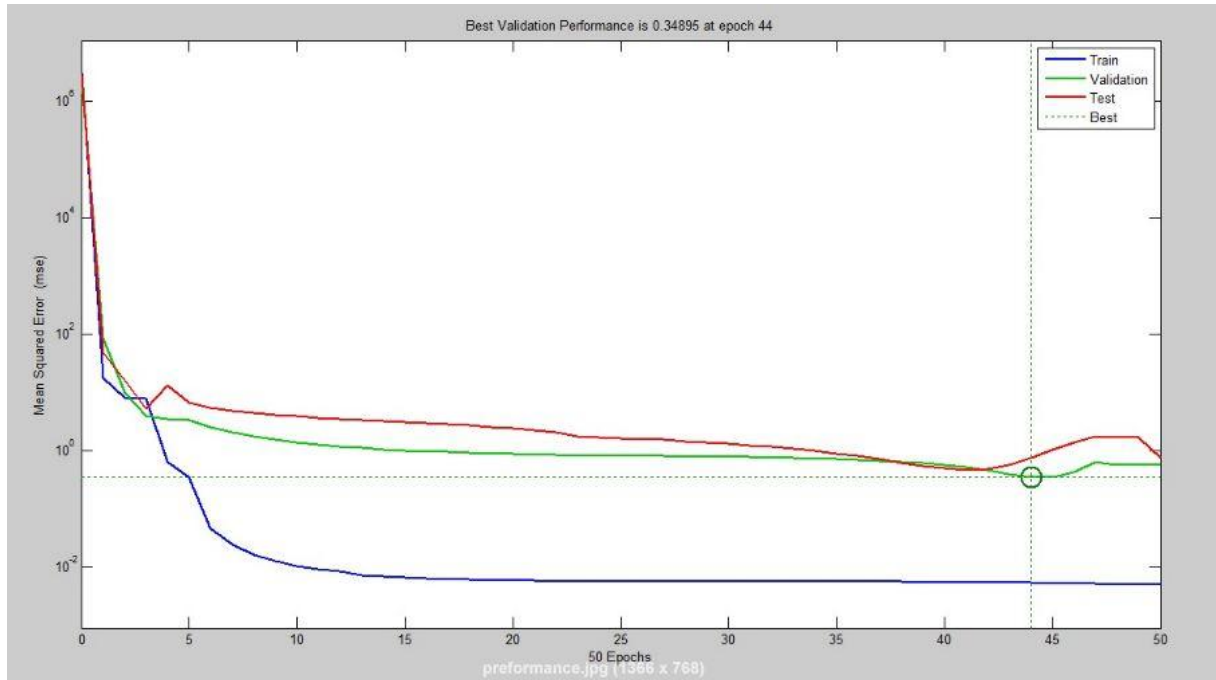
3	6	9	12	15	18	21	24	27	30	33	36	39	43	46		
49	52	55	58	61	64	67	70	73	76	79	82	86	89	92	93	94
95	98	101	104	107	110	113	116	119	122	125	129	132	135	138		
141	144	147	150	153	156	159	162	165	169	172	175	178	181	18	185	
186	187	190	193	196	199	202	205	209	212	215	218	221	224	225	226	
227	228	229	230	231	232	233	236	239	242	246	249	252	255	258	261	
264	267	270	273	276	279	283	286	289	292	295	298	301	304	307		



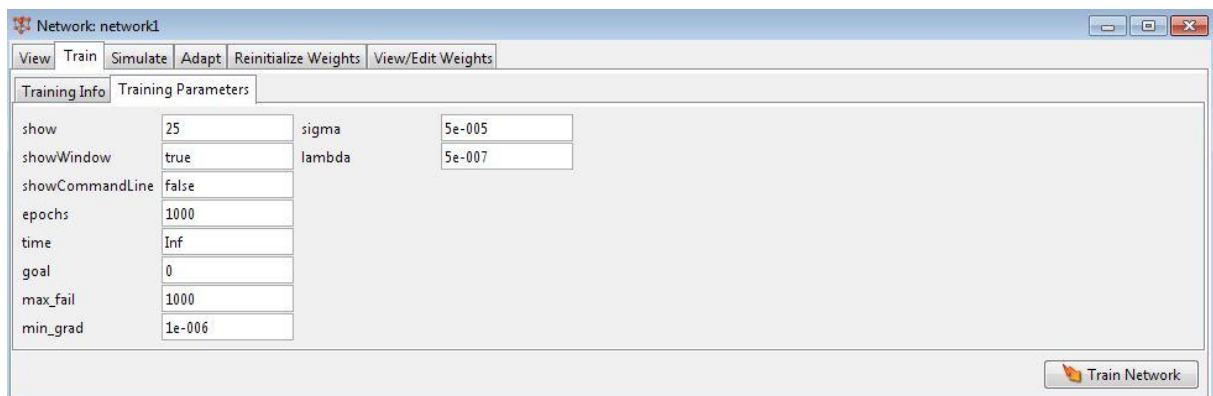
Matlab 2008b аутоматски бира параметре за обуку, валидацију и тестирање и то у односу 70:15:15.

8.1. Обука неуронских мрежа мобилног робота које генеришу потребан број обрта мотора на основу тражене раздаљине

Обука неуронске мреже је заустављена на 50 итерацији, јер је градијент достигао најнижу вредност, што значи да се достигао неки од минимума (глобални или локални) у равни грешака.



Слика 8.1 Графикон зависности најмање грешке квадрата у зависности од броја итерација 1-[20]-1

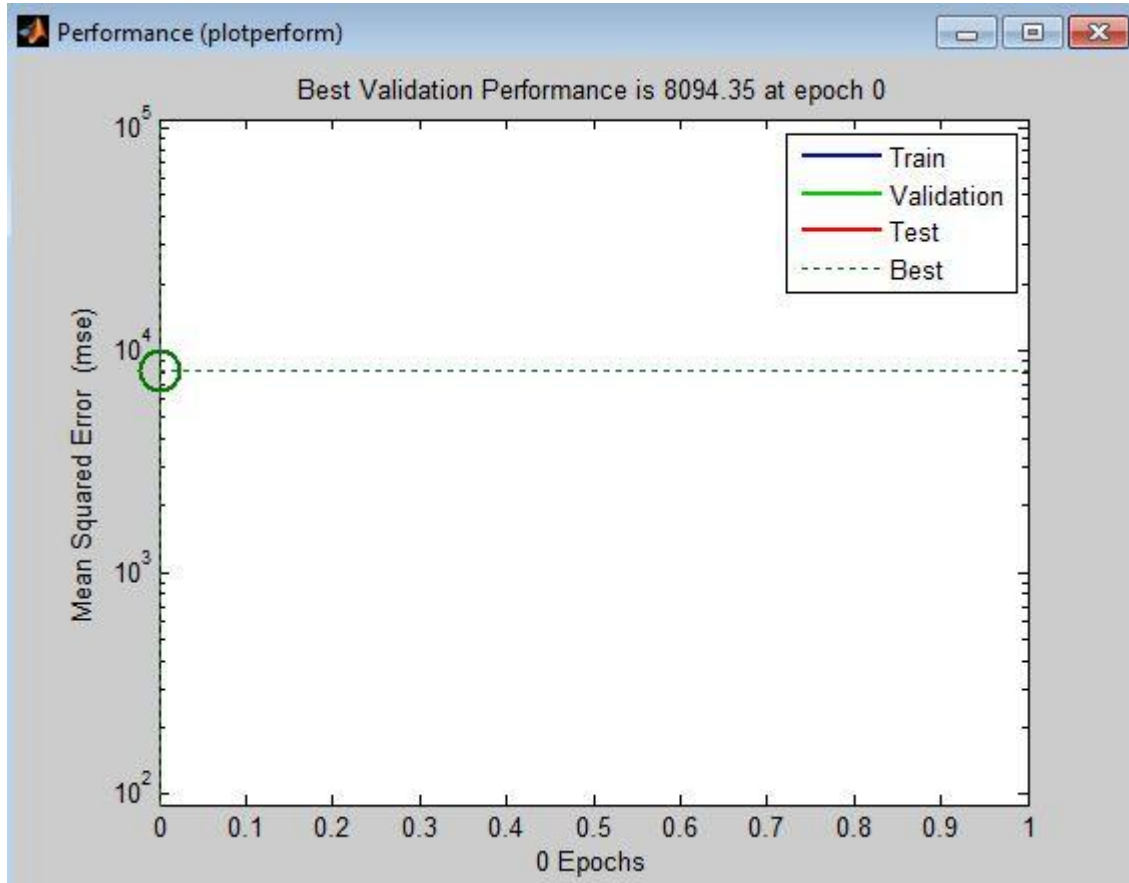


Слика 8.1.2 Параметри обучавања неуронске мреже 1-[20]-1

Најмања грешка обуке је 0,02, а најмања грешка валидације је 0,34895

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	

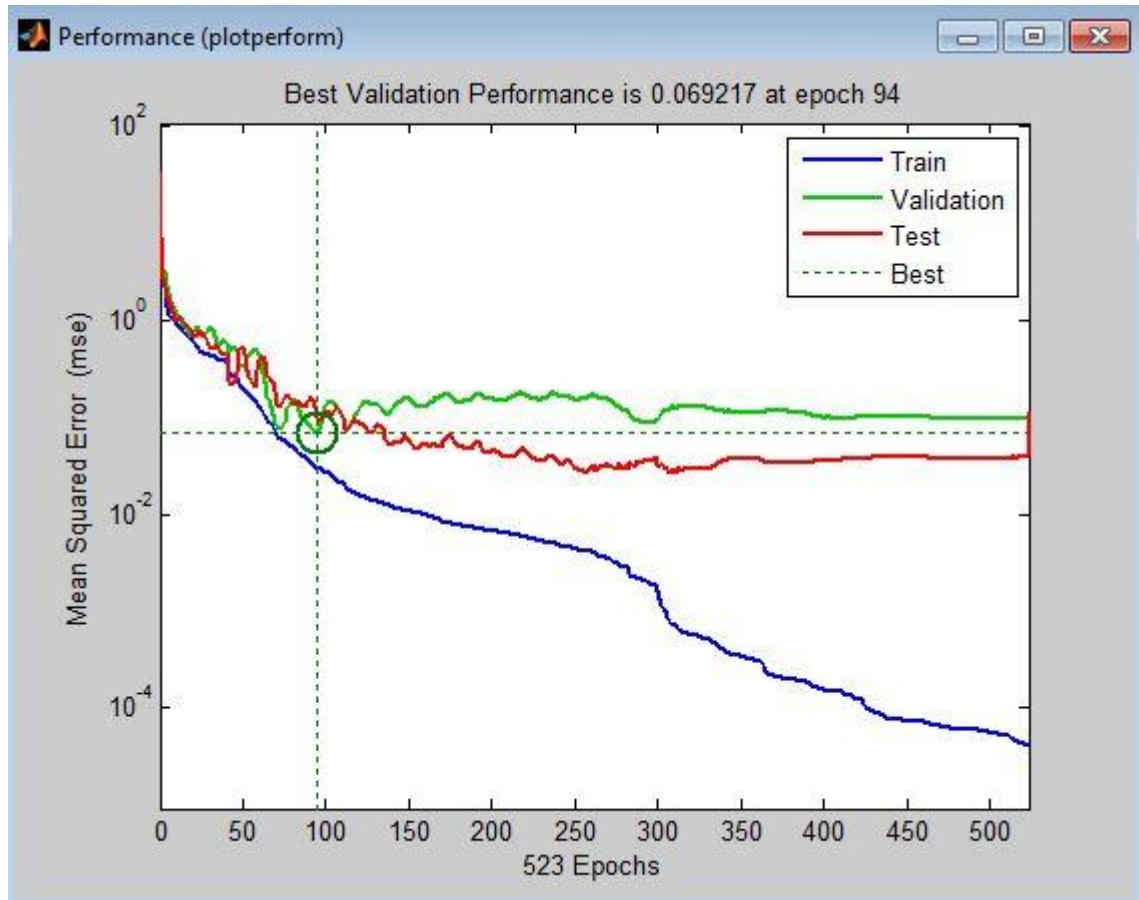
Архитектура 1 [4 6 6] 1 се показала као неупотебљива јер није ни могла да крене тренинг односно да изврши ни једну итерацију, због наглог пада градијента, а при томе је грешка била јако велика, око 1300, при истим почетним параметрима као код претходне архитектуре.



Слика 8.1.3 Графикон зависности најмање грешке квадрата у зависности од броја итерација 1 [4 6] 1

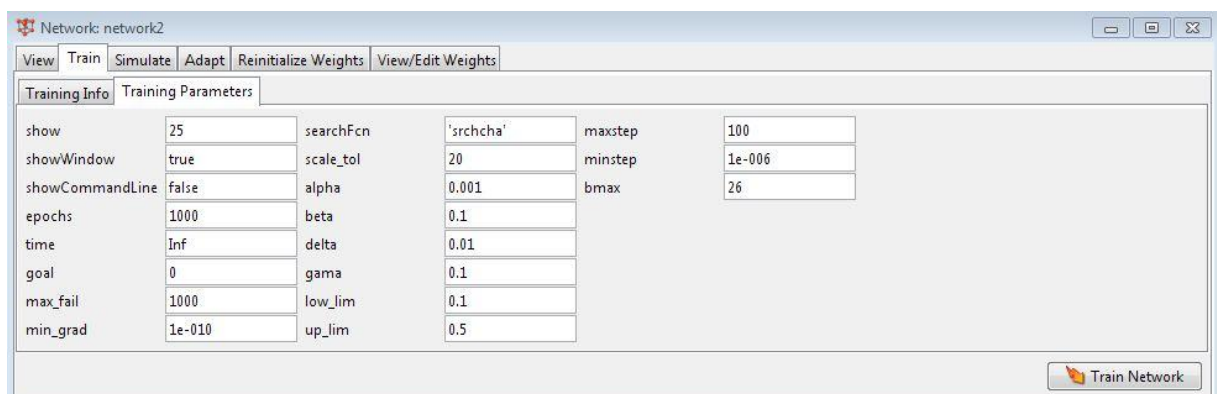
Архитектура 1 [10 10 10] 1 је накнадно тренирана мрежа са *Conjugate gradient backpropagation with Fletcher-Reeves updates* алгоритмом која је показала најбоље перформансе учења. Мрежа је урадила 500 итерација и при томе достигла минимални грешку од 0,0001, док је минимална грешка валидације достигнута у стотој итерацији и износи 0,069.

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



Слика 8.1.4 Графикон зависности најмање грешке квадрата у зависности од броја итерација 1 [10 10 10] 1

Парамери обучавања мреже



Слика 8.1.5 Параметри обучавања неуронске мреже 1 [10 10 10] 1

Бројеви индекса чланова групе:				Шк. год.	Датум:	Прегледао:
17/07	396/07	362/07	175/05	2009/2010	21.06.2010.	



8.2. Обука неуронских мрежа за позиционирање стационарног робота на основу координата мобилног робота

Везу између обртаја мотора за позиционирање и координата мобилног робота представља вишеструко преносна функција моделована помоћу неуронске мреже. Улази у мрежу су координате система који се налази у равни која је радни простор мобилног робота, а излаз је број обртаја мотора. Мотор је спрегнут са осом обртања стационарног робота преко пужног преносника чији су нам параметри непознати. Пошто је преносни однос непознат приступили смо прикупљању података који могу да укажу на то како се број обртаја мотора индиректно преноси на обртање осе робота и тако формирали обучавајуће парове.

Табела 8.2 Излазни вектор

3	3	3	3	6	6	6	6	9	9	9	9	1
12	12	12	0	0	0	0	12	12	12	12	9	9
9	9	6	6	6	6	3	3	3	3			

Табела 8.2.1 Улазни вектор

200	209	219	228	170	178	186	194	123	129	135	141	65	68	71	74	210	220	230
65	68	71	74	123	129	135	141	170	178	186	194	200	209	219	228	0	0	0

240	74	71	68	65	141	135	129	123	194	186	178	170	228	219	209	200
0	-228	-219	-209	-200	-194	-186	-178	-170	-141	-135	-129	-123	-74	-71	-68	-65

8.3. Мрежа за избор тачака

Мрежа за избор тачака за тренинг користи улазни вектор величине 2 x 77 и излазни вектор величине 16 x 77. Због великог броја улаза и излаза приказано је само по првих 10 улазних и излазних вектора. Грешка излаза износи 3,64 што је задовољавајуће за примену којој је мрежа намењена.

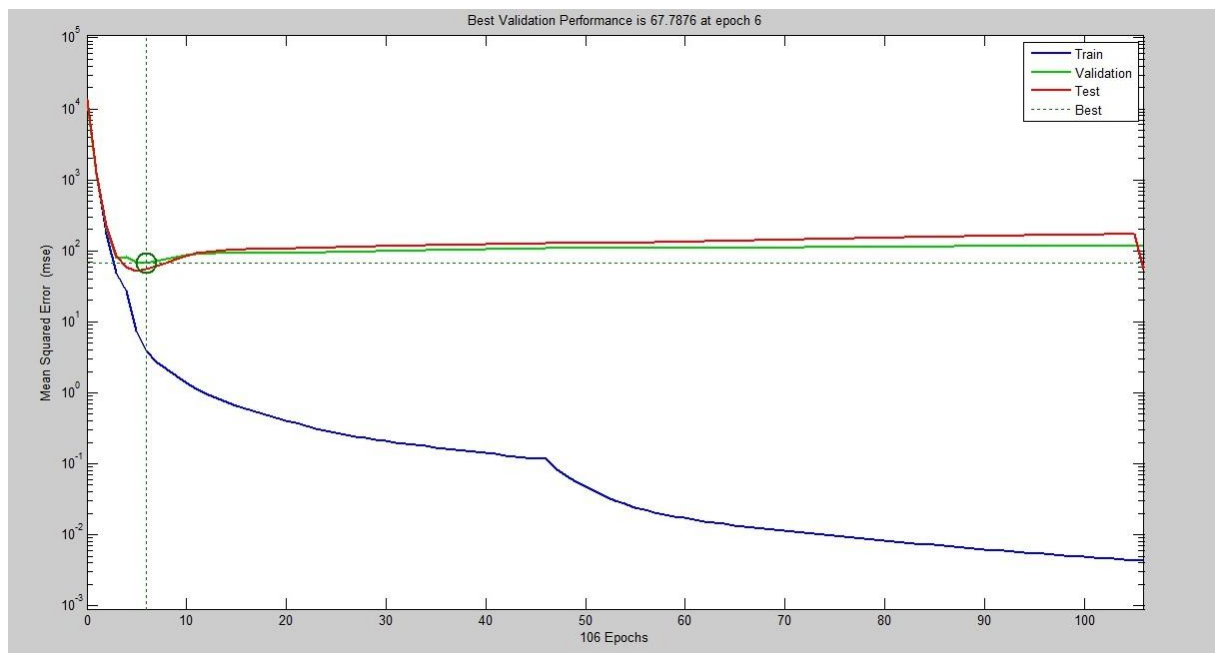
Табела 8.3.1 Излазни вектор

0	0	0	0	0	0	0	0	0	0
0	10	20	30	40	50	60	70	80	90



Табела 8.3.2 Улазни вектор

10	10	10	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10	10
10	0	0	0	10	10	10	10	10	0
0	0	0	0	0	0	0	0	0	0
10	0	10	10	0	0	0	0	0	10
0	10	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
10	20	30	40	50	60	60	60	70	80
10	10	20	30	40	50	70	70	80	90
0	20	30	40	30	40	50	80	90	80
10	20	30	40	50	60	60	60	70	80
0	10	10	20	50	60	70	60	70	100
0	0	20	30	40	50	70	70	80	90
0	0	10	20	30	40	50	80	90	100
0	0	10	20	30	40	50	80	90	100



Слика 8.3.1 Графикон зависности најмање грешке квадрата у зависности од броја итерација

Бројеви индекса чланова групе:

Шк. год.

Датум:

Прегледао:

17/07

396/07

362/07

175/05

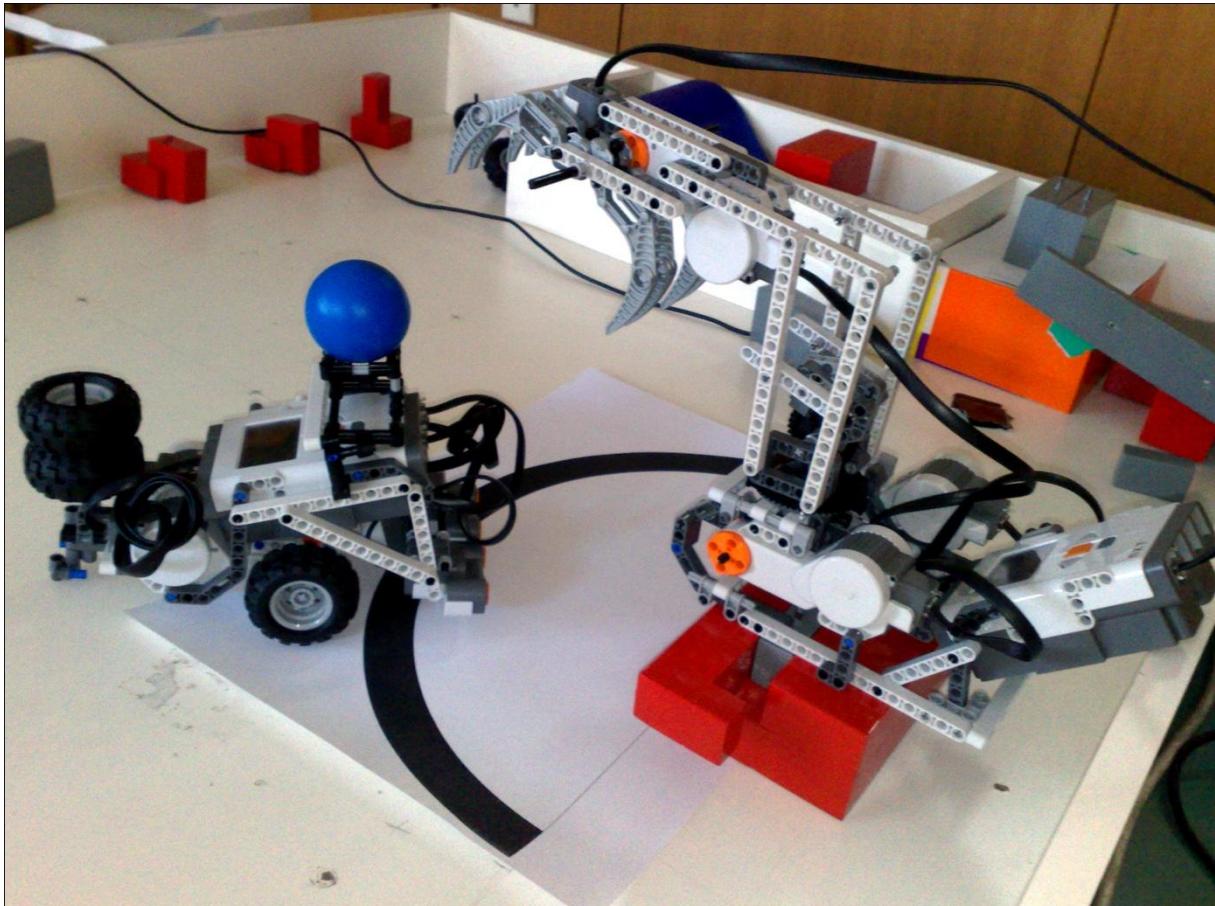
2009/2010

21.06.2010.



Закључак

Претходна анализа и време проведено у решавању проблема везаних за LEGO mindstorm NXT наводе на закључак да вешто имплементирани интелигентни агенти могу чак и од нечега што је првенствено замишљено као играчка нове генерације направити едукациони алат који може делимично да се понаша аутономно и успешно моделује и имитира компоненту неког реалног комплексног система. Даља импликација је окренута ка томе да су могућности озбиљних интелигентних система тренутно несагледиви.



Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао:



Литература

- Stuart J. Russell and Peter Norvig, *Artificial Intelligence A Modern Approach*, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1995
- Gerhard Weiss *Multiagent Systems A Modern Approach to Distributed Modern Approach to Artificial Intelligence*, Massachusetts Institute of Technology, 1999
- *Building Robots with LEGO MINDSTORMS*, PUBLISHED BY Syngress Publishing, Inc., 800 Hingham Street, Rockland, MA 02370, Copyright © 2002
- Danica Janglová, *Neural Networks in Mobile Robot Motion*, Janglová, D. / *Neural Networks in Mobile Robot Motion*, pp. 15-22, *International Journal of Advanced Robotic Systems*, Volume 1 Number 1 (2004), ISSN 1729-8806
- Aluizio F. R. Araújo, Christof Dücker, and Helge Ritter, Guilherme A. Barreto, *Student Member*, A Distributed Robotic Control System Based on a Temporal Self-Organizing Neural Network, *IEEE TRANSACTIONS ON SYSTEMS, MAN and CYBERNETICS—PART C: APPLICATIONS AND REVIEWS*, VOL. 32, NO. 4, NOVEMBER 2002
- Balic, J., *Advances in Production Engineering & Managment* 1, 13-22, *Intelligent CAD/CAM Systems for CNC Programing – an overview*, University Maribor, Faculty of Mechanical Engineering, Smetanova 17, Maribor, Slovenia, 2006, ISSN 1854-6250
- Mikell P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*, third edition, by Person Education Inc. Upper Saddle River, New Jersey 07458, 2008
- <http://en.wikipedia.org/wiki/Intelligence>
- http://www.cis.hut.fi/harri/thesis/valpola_thesis/node34.html
- Зоран Миљковић, *Системи вештачких неуронских мрежа у производним технологијама*, Машински факултет, Универзитет у Београду, 2008.
- Зоран Миљковић, Драган Александрић, *Вештачке неуронске мреже*, Машински факултет, Универзитет у Београду, 2009.

Бројеви индекса чланова групе:

17/07

396/07

362/07

175/05

Шк. год.

2009/2010

Датум:

21.06.2010.

Прегледао: