



Универзитет у Београду Машински факултет

Дипломске академске студије

Модул ЗА ПРОИЗВОДНО МАШИНСТВО

ИНТЕЛИГЕНТНИ ТЕХНОЛОШКИ СИСТЕМИ

ПРОЈЕКАТ

Оцена пројектног задатка:	Предметни наставници:	проф. др Зоран Миљковић и проф. др Бојан Бабић		
	Предметни сарадници:	др Божица Бојовић и Најдан Вуковић, дипл. маш.инж		
Потпис наставника:	Група:			
	РБ	Презиме и име:	Бр.инд.	Потпис:
	1.	Вујовић Бојан	1116/08	
	2.	Дерета Славомир	1125/08	
	3.	Дрљић Миодраг	1127/08	
	4.	Чолак Владимир	1239/08	
5.	Штрбац Бојан	1243/08		

Школска година: 2009/2010.

УНИВЕРЗИТЕТ У БЕОГРАДУ – МАШИНСКИ ФАКУЛТЕТ

Дипломске академске студије – 2. година

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

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

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

ПРОЈЕКТНИ ЗАДАТАК (1/2)

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

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

За дату диспозицију технолошког окружења „XY”:

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

Решењем пројектног задатка обухватити:

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

Напомене:

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

Датум издавања задатка: **16.10.2009.**

Рок завршетка задатка: **04.12.2009.**

Задатак издао: _____

(Најдан Вуковић)

Резиме

У првој декади 21. века, постали смо сведоци наглог процвата примене вештачке интелигенције у многим сферама живота: интелигентне, куће, аутомобили, бела техника само су неки од примера примене интелигентних технолошких система. За нас, као произвођаче, је ипак најинтересантнија област која се бави применом интелигентних мобилних робота у индустрији. Пренесено на конкретан проблем, наш задатак је био да се у експерименталном моделу технолошког окружења изврши имплементација интелигентног мобилног робота. За технолошко окружење је узет реалан модел једне фабрике, док је улогу интелигентног робота у тој фабрици преузео модел робота *LEGO Mindstorms NXT*. За саму интелигенцију робота су заслужне вештачке неуронске мреже које смо обучавали за различите задатке, а без којих би остварење циља пројекта било немогуће. Поред вештачких мрежа, кључну улогу је одиграла и математичка метода по имену "Калманов филтер", без које робот не би могао да се снађе у простору. Комбинацијом ова два елемента, добили смо жељено интелигентно кретање робота кроз постројење. Само кретање је у 80% било успешно, што је последица несавршености мерних система и конструкције робота.

Кључне речи: *интелигентни технолошки системи, интелигентни мобилни робот, вештачка интелигенција, интелигентни мобилни робот, вештачке неуронске мреже, Калманов филтер*

¹ **Бојан Вујовић 1116/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија
Е-пошта: bokiskroz@gmail.com

² **Славомир Дерета 1125/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија
Е-пошта: slavomir_dereta@hotmail.com

³ **Миодраг Дрљић 1127/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија
Е-пошта: diablo26002005@yahoo.com

⁴ **Владимир Чолак 1239/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија
Е-пошта: colakv@gmail.com

⁵ **Бојан Штрбац 1243/08**, Универзитет у Београду – Машински факултет, студент друге године
Дипломских академских студија
Е-пошта: strbacb@gmail.com

Списак слика

Слика 1.1.:	<i>LEGO Mindstorms NXT</i> робот	9
Слика 2.1.1.:	<i>Layout</i> производног монтажног предузећа "XY"	11
Слика 2.3.1.:	Принцип рада енкодера.....	13
Слика 2.3.2.:	Структура преносне гране мотора <i>LEGO</i> робота.....	15
Слика 2.3.3.:	Позиција енкодера у <i>LEGO</i> роботу.....	15
Слика 2.3.4.:	Изглед сензора за детекцију светлости.....	16
Слика 2.3.5.:	Препознавање боја у зависности од читавања са оптичког сензора	16
Слика 2.3.6.:	Принцип функционисања оптичког сензора.....	17
Слика 2.4.1.:	Приказ типичног процесуирајућег елемента – неурона	18
Слика 2.4.2.:	Вештачка неуронска мрежа.....	18
Слика 2.4.3.:	Сигмоидна активациона функција.....	20
Слика 2.4.4.:	Backpropagation - принцип функционисања.....	21
Слика 4.1.:	Базни модел кретања робота.....	25
Слика 4.2.:	Процес обучавања мреже.....	27
Слика 4.3.:	Резултати обучавања мреже.....	27
Слика 4.4.:	Резултати мерења црне и беле боје оптичким сензором	29
Слика 4.5.:	Процес обучавања мреже за препознавање боје.....	29
Слика 4.2.1.:	Приказ "троугаоне" матрице за предузеће "XY".....	31
Слика 4.2.2.:	Привремени резултати прве симулације у Anylogic окружењу	32
Слика 4.2.3.:	Коначни резултати прве симулације у Anylogic окружењу	32
Слика 4.2.4.:	Коначни резултати друге симулације у Anylogic окружењу	33
Слика 4.2.5.:	Друга диспозиција фабричког окружења.....	33

Списак табела

Табела 2.4.1.:	Разлике биолошких и неуронских мрежа.....	17
Табела 4.1.:	Једначине које одређују модел кретања.....	25
Табела 4.2.:	Вредности мерења AngleLimit у зависности од жељеног угла.....	26
Табела 4.2.1.:	Параметри машина и растојања између њих.....	30
Табела 4.2.2.:	Резултати симулације у <i>Anylogic</i> софтверу.....	34

Садржај

1. Увод	8
2. Поставка проблема.....	11
2.1. Модел технолошког окружења.....	11
2.2. Мобилна роботика.....	12
2.3. Мерни систем.....	14
2.4. Вештачке неуронске мреже.....	17
2.5. Калманов филтер.....	21
3. Концепсијско решење пројектонг задатка.....	24
4. Пројектно решење.....	24
4.1. Мобилни интелигентни робот.....	24
4.2. Anylogic окружење.....	30
5. Експериментални резултати.....	35
6. Анализа и дискусија.....	36
7. Закључак.....	36
8. Литература.....	37

1. Увод

Овај пројектни задатак обухвата представљање и решење проблема транспорта материјала, сировина, полуфабриката или готових делова у фабрици "XY". Решење проблема се заснива на идентично направљеном окружењу фабрике. Прецизније речено, сваки елемент је сразмерно смањен, како би симулација решавања проблема била потпуно идентична понашању система у реалном окружењу.

Сам задатак се односио на програмирање робота који ће кроз фабрику допремати већ поменуте материјале путањама које су јасно дефинисане, како не би дошло до судара са инвентаром фабрике. Као физички симулациони модел, коришћено је умањено радно окружење и мобилни робот *LEGO Mindstorms NXT*⁶ (слика 1.1.). За управљање роботом, користили смо софтверско решење развијано у "Matlab"⁷-у, које у себи садржи вештачке неуронске мреже које су кључни елемент понашања робота у окружењу.

Робот је за задатак имао пролаз кроз референтне тачке у окружењу које су задате са наше стране, како не би дошло до колизије са неком од машина у постројењу. Задатак вештачких неуронских мрежа је био да робота "научи" да препознаје те тачке као референтне, како би знао у ком правцу даље треба да се креће. Поред основних референтних тачака, ми као корисници робота смо морали да дефинишемо и међутачке кроз које би робот пролазио да не би дошло до наглог скретања робота. Међутачке смо могли да дефинишемо на два начина: применом алгоритма А* или ручно. Ми смо одабрали ручну методу.

Поред прорачуна путања које робот мора да прође, рађена је и симулација технолошког система у "Anylogic"⁸ окружењу. "Anylogic" нам је помогао да симулирамо реалан рад постројења фабрике, с тим што је уместо транспортера предвиђено коришћење мобилног робота. Сви параметри који се тичу рада система су унесени у софтвер и на основу тога смо добили реална решења колико је ефикасан тренутни распоред машина, а колико је ефикасан распоред машина предложен са наше стране, број делова у реду за чекање на машину и слично.

⁶ Званичан сајт фирме LEGO <http://mindstorms.lego.com/en-us/Default.aspx>

⁷ Званичан сајт софтверског пакета Matlab <http://www.mathworks.com/products/matlab/>

⁸ Званичан сајт софтверског пакета Anylogic <http://www.xjtek.com/>

Решење проблема је детаљно дато на наредним странама, а решењем је обухваћено:

1. Формирање конфигурације мобилног робота;
2. Развијање и имплементација модела кретања у *Matlab* окружењу
3. Развијање и имплементација опсервационог (сензорског) модела мобилног робота применом система вештачких неуронских мрежа
4. Примена Калмановог филтера у циљу одређивања положаја мобилног робота у окружењу ;
5. Верификација резултата у експерименталном моделу технолошког окружења
6. Развијање симулационог модела у *Anylogic* окружењу
7. Предлагање новог диспозиционог плана технолошког окружења и формирање "троугаоне" матрице за нови модел на основу резултата симулације
8. Упоредивање два модела диспозиционог плана на основу резултата симулације и закључак



Слика 1.1.: LEGO Mindstorms NXT робот

2. Поставка проблема

2.1. Модел технолошког окружења

Предузеће "XY" се бави производњом и монтажом SIVACON електро-ормана. Како би оптимизација плана распореда физичких средстава дошла до коначног решења, Siemens AG је анализирао пројектоване технолошке поступле обраде делова, као и анализу дефинисане технологије за монтажу SIVACON електро-ормана. Резултати анализе су показали да је решење распореда машина које је дато у наставку у складу са прописаним нормативима за пројектовање фабричких постројења.[2]

На слици 2.1.1. је дат *layout* монтажног погона предузећа "XY". Са слике се могу уочити положаји машина које се налазе у погону. Списак машина које се налазе у погону је следећи:

- M1 – маказе за сечење
- M2 – НУ машина за пробијање и просецање
- M3 – НУ хидраулична апкант преса
- M4 – машина за исецање профила
- M5, M6 – стубне бушилице
- M7 – Кружна тестера
- M8 – оштрилица
- M9 – линија за обраду делова од бакра

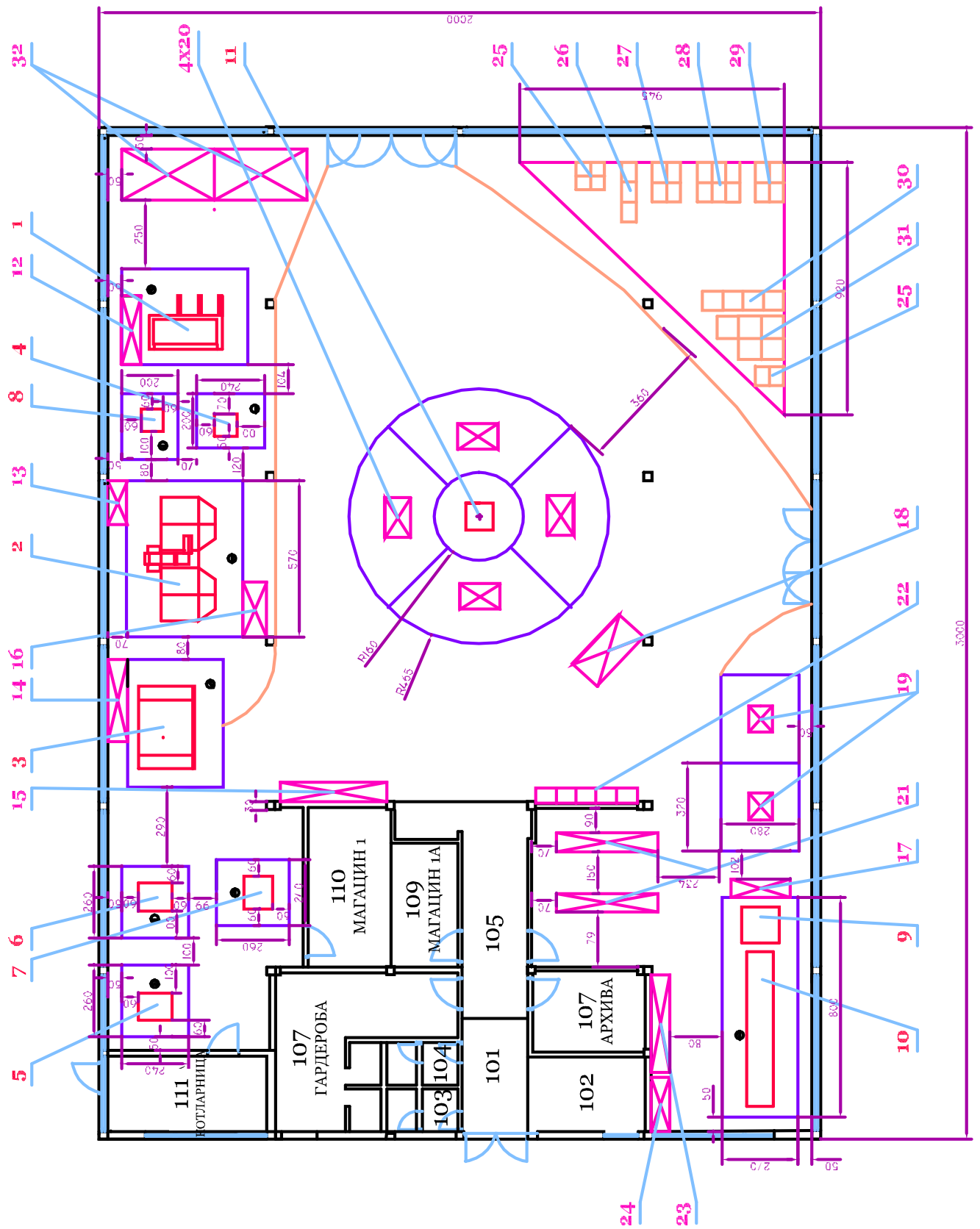
Распоред машина у постројењу је разврстан према редоследу технолошких операција, што подразумева да се машине и уређаји постављају у линијском распореду, где је редослед извођења операција условљен пројектованим технолошким поступком за сваки део који се израђује, а који директно утиче на квалитативне међузависности машина и опреме које се користе у том погону.

Током процеса пројектовања распореда машина и радних места коришћен је квалитативан критеријум међузависности, с обзиром на то да су у разматрање биле укључене све активности које су значајне за одвијање основног технолошког процеса, при чему је количина материјала која се креће кроз процес разматрана на предвиђеном глобалном једногодишњем нивоу производње и монтаже 300 SIVACON електро-ормана.

Према пројектованом *layout-u* датом на слици 2.1.1. види се да су у линијском распореду кључне машине за обраду челичног лима, машине алатке M1, M2, M3 и M4, а да су издвојене машине M5, M6 и M7, које су у мањој употреби, као и то да је линија за обраду бакра (M9) потпуно дислоцирана у односу на све друге обрадне системе, јер је и технолошки процес израде делова од бакра издвојен у односу на остале технолошке процедуре. Оштрилица алата, M8, највише се користи за оштрење алата који су везани за машину M2, тако да је позиционирана баш уз њу.

Тренутно стање у фабрици је такво да посао транспорта материјала, сировина, обрадака и осталог већ помеутог обавља виљушкар. Идеја пројекта је да се тај виљушкар замени мобилним роботом који би те послове обављао уместо њега. Према технолошком плану обраде, робот би се кретао од складишта материјала до машина, потом између машина и на крају би се враћао у складиште.

Токови материјала, односно путање којима се робот креће су дати у поглављу 4.2..



Слика 2.1.1.: Layout производног монтажног предузећа "XY"

2.2. Мобилна роботика

Реч *робот* долази од старословенске речи *робота (робота)* која значи рад, односно тешки рад. Први пут реч *робот* је употребио Карел Чапек у својој утопијској драми Р.У.Р. (Росумови Универзални Роботи) [3].

Према дефиницији ISO 8373 робот је:
Аутоматски управљан, репрограмабилан, вишенаменски манипулатор програмиран за кретање у три или више оса, који може бити стационаран или мобилан за примене у индустријској аутоматизацији. Ова дефиниција вреди само за индустријске роботе, а не и за мобилне, јер се за мобилне постављају нови, далеко сложенији захтеви.

Захтеви које мобилни робот треба да испуни су:

- Аутономно померање у непознатом простору
 - Где је? Где иде? Како до тамо доћи? Како избећи сударе са другима?
 - Робот аутономно моделира простор у коме се налази на темељу информација које прима са својих сензора
- Обављање задате мисије, односно задатка
- Интеракција са људима и/или са другим роботима у окружењу
- Способност учења и интелигентног закључивања
- ...

Могућа дефиниција мобилног робота може бити: *мобилан робот је мобилан и манипулативан физички систем који се аутономно помера кроз неструктурирани простор, остварујући притом интеракцију са људским бићима или аутономно обављајући неки посао уместо њих.*

Роботи у стварности представљају мехатронске системе и једног робота можемо дефинисати као скуп различитих ентитета:

- Механичка конструкција
- Енергетска електроника
- Управљачка електроника
- Програмски језици
- Навигација
- Комуникација човек-робот
- Интелигентно управљање
- Рачунска архитектура
- Сензорски системи
- Актуатори

Сама по себи мобилна роботика представља вишедисциплинарно подручје технике које се бави развојем и производњом аутономних мобилних робота и у себе интегрише велики број научних дисциплина (машинство, електронику, аутоматику, математику...)

Класификација мобилних робота се може извршити на неколико начина, али ћемо овде поменути само једну од њих. Роботи се према преносном медију деле на:

- Ваздушне
- Водене
- Свемирске
- Копнене

Поред ове најједноставније поделе, мобилни роботи се могу поделити и према другим својим особинама, као нпр.: према степену аутономности (полуоператорско или потпуно аутономно) , према облику тела робота (једноставан или сложен) , према врсти терена за који је предвиђен (унутрашњи простор или спољашњи) , према локомотивном механизму

(точкови, ноге, гусенице...), према врсти управљања (диференцијални погон, синхрони погон...), према општој класи (AGV, услужни роботи, истраживање, едукација...).

Мотиви развоја мобилних робота су разни и има их много, али су најбитнији:

- Смањење трошкова и повећање продуктивности рада уз мање трошкове рада, већу свеукупну расположивост робота, бржи рад у односу на човека, повећан квалитет производа/услуге
- могућност приступа местима која су за човека опасна по живот, превише удаљена или недоступна
- старење становништва у развијеним земљама

Могућности примене су такође многоструке и потребно је само мало проучити тренутно стање у свету и погледати у будућност. Резултати су фантастични:

- свуда где се користе возила са возачем, у будућности би могао бити мобилни робот (MP)
- Свуда где се данас користе или би се могле користити транспортне траке и/или манипулатори, у будућности би могао бити MP
- Свуда где се данас користе домаће животиње за рад, у будућности би могао бити мобилни робот
- ...

Главни физички подсистеми кор робота се могу сврстати у неколико категорија, но те се категорије могу даље поделити, па ћемо у овом случају све то представити великом саставницом:

- Механички подсистем
 - Кућиште MP-а (носи све компоненте MP-а)
 - Пропулзија (осигурава погонску енергију путем електричних, хемијских мотора...)
 - Суспензија (расподељује оптерећење MP-а ради одржавања стабилности)
 - Локомоција (механизам претварања енергије пропулзије у померање MP-а)
 - Помоћни механизми (руке, главе сензора...)
- Електрични подсистем
 - Енергетска јединица (генерише енергију пропулзијској јединици, типично дизел и плински генератори, соларне ћелије, акумулатори...)
 - Батерије и претварачи енергије (акумулатори, пуњачи акумулатора, непрекидни извори напајања)
 - Каблови за пренос енергије, сила, сигнала
- Сензорски подсистем
 - Проприоцепцијски сензори MP-а (мере унутрашња померања механизма MP-а: енкодери, ризолвери, акцелерометри, компаси, жироскопи)
 - Перцепцијски сензори MP-а (мере појаве везане уз околни простор MP-а: контактни близински сензори, инфра црвени сензори, ултразвучни сензори удаљености, ласерски сензори удаљености, CCD камере)
 - Антене (служе за бежичну комуникацију са главним рачунарима и/или другим MP-има)
- Управљачки подсистем
 - Управљање померањем MP-а
 - Заокрет MP-а (управљање смером померања робота)
 - Брзина MP-а (управљање брзином померања MP-а у жељеном смеру)
 - Ова два управљачка задатка могу бити спрегнута или независна
 - Управљање стањем околине MP-а
 - Вентилација, климатизација, грејање
- ...

2.3. Мерни систем

Енкодер

Енкодери представљају једну широку класу дигиталних сензорских система који се користе за мерења линисјких и угаоних помераја. Већ из самог назива се види да се ради о чисто дигиталним системима, који кодирају (енг. *encoding*) угаони или линијску позицију, коришћењем одговарајућих електромеханичких склопова [4].

Генерално, енкодери се класификују у две велике групе:

1. Инкрементални енкодери (које смо ми користили), и
2. Апсолутни енкодери

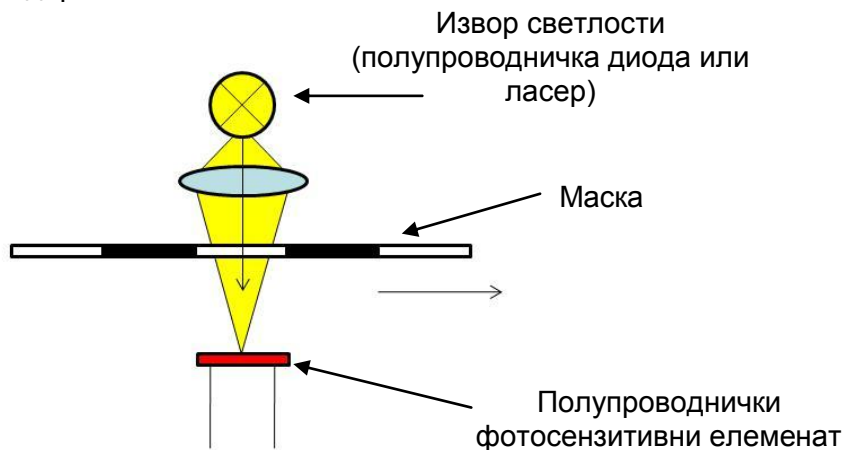
Инкрементални енкодери мере релативни положај у односу на неку унапред познату иницијалну координату (угаону или линијску). Апсолутни енкодери мере апсолутну позицију, која је дефинисана конструктивним решењем склопа у оквиру кога функционишу.

Мерење брзине се врши диференцирањем сигнала који генерише енкодер, односно дељењем угаоног или линијског инкремента са временом које је протекло за његово извршење.

Због свог специфичног конструкционог облика, енкодери немају особину нагомилавања грешке и такође, њихова тачност није условљена величином мерног опсега (максимални угаони или линијски померај), што је од посебног орактичног значаја. Алтернативни аналогни системи немају ово својство, тако да у савременим серворегулационим мехатронским склоповима, почев од машина алатки, робота, па до лансера ракета дигитални енкодери представљају незаменљиву компоненту. Различита конструктивна решења омогућавају велику употребну флексибилност, у смислу могућности уградње у врло различитим конструкцијама.

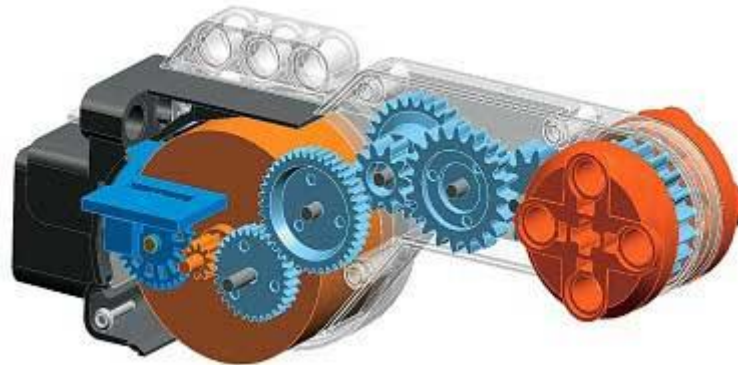
Енкодери се најчешће изводе коришћењем оптичких и магнетних електромеханичких претварача. Доминантна технологија у савременој индустријској пракси је базирана на оптоелектронским претварачима.

Оптички енкодери поседују оптичке елементе, фото диоде и фото транзисторе, од којих је један извор, а други пријемник. Поред оптичких елемената, енкодер садржи и маску на којој се налазе прозирна и непрозирна поља. Ова поља наизменично прекидају оптички пут између извора и пријемника. Величина ових поља је у директној сразмери са механичком резолуцијом енкодера.



Слика 2.3.1. Принцип рада енкодера

У нашем случају, на роботу *LEGO Mindstorms NXT* је имплементиран инкрементални енкодер, па ћемо га мало детаљније у наредних пар редова.



Слика 2.3.2.: Структура преносне гране мотора *LEGO* робота

Инкрементални енкодери раде на принципу генерисања поворке импулса чијим се бројањем израчунава угаони померај. Зато се инкрементални енкодери често називају бројачким енкодерима [7].

Генерално, постоје две врсте инкременталних енкодера:

1. Енкодери са дискретним генератором импулса,
2. Енкодери са аналогним генератором импулса и интерполатором.

Инкрементални енкодери се са подједнаком једноставношћу изводе као обртни и као линијски. Због бројачког карактера рада ове врсте енкодера, максимална дужина или угао који се мери ограничен је само капацитетом бројача.

Детекција смера се врши на следећи начин: у пракси се примењују енкодери који имају два генератора импулса, односно два канала која су фазно померена за $\pi/2$. Препознавањем који канал предњачи може се препознати и смер ротације. Уобичајено је да се ови канали означавају великим словима А и Б.

Референцирање: Поред наведених канала који генеришу поворку импулса, постоји и трећи канал који садржи само један импулс по обртају. То је референтни канал којим се означава локација иницијалне, односно нулте тачке од које се врши бројање. Уобичајено је да се овај канал означава словом *Z* (од енглеског *zero*). У процедури иницијализације мехатронског система, спроводи се и секвенца која омогућава да се идентификује референтни (нулти) импулс, чиме се ресетује припадајући бројач и тако дефинише координатни почетак конкретне осе.



Слика 2.3.3.: Позиција енкодера у *LEGO* роботу

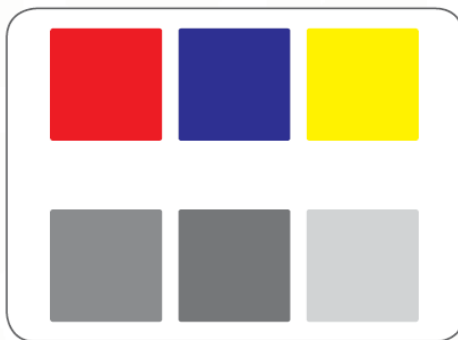
Светлосни сензор



Слика 2.3.4.: Изглед сензора за детекцију светлости

Основна функција светлосног сензора робота је да омогући да се направи разлика између светла и таме, да мери интензитет светлости у просторији и да се мери интензитет светла на некој обојеној површини [5].

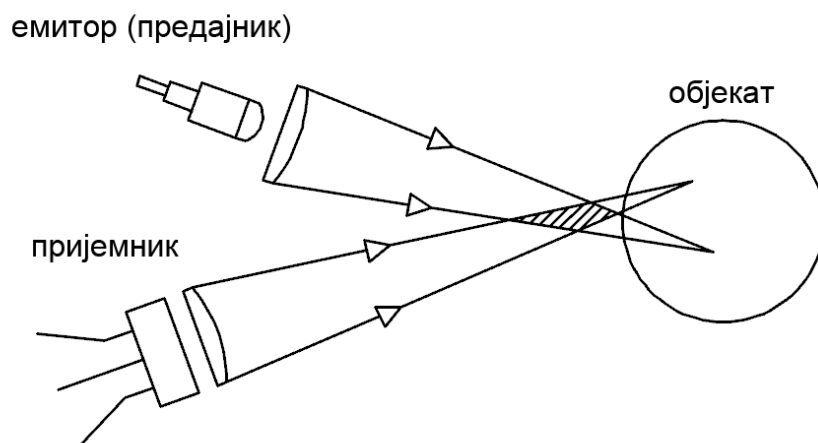
Сензор не препознаје саме боје, већ искључиво препознаје интензитет рефлексије светлости неке боје. То у преводу значи да су све боје у ствари само нијансе сиве, односно приближно црна или приближно бела боја. Ради лакшег разумевања, дат је приказ препознавања рефлексије боја на слици 2.3.5 [8].



Слика 2.3.5.: Препознавање боје у зависности од очитавања са оптичког сензора

Оптички сензор се састоји од предајника (емитора) који најчешће чини ласерска диода или у нашем случају LED диода као извор светлости и пријемника. Емитовани светлосни зрак пада на објекат, одбија се и враћа на пријемник, обично фототранзистор.

Зона осетљивости није јасно дефинисана. Недостатак сензора је чињеница да се сочива могу лако запрљати, па на то треба обратити пажњу с времена на време. Исто тако, подложни су утицају шума из околине, имају слабу осетљивост на објекте са ложом рефлексијом и слично.



Слика 2.3.6.: Принцип функционисања оптичког сензора

2.4. Вештачке неуронске мреже

Према дефиницији из [1]: Вештачка неуронска мрежа је парадигма вештачке интелигенције која се дефинише као конективни модел за резонување заснован на аналогiji са мозгом, уз наглашену когнитивну способност да учи и врши генерализацију стеченог знања.

Кључни елемент ове парадигме је неубичајена структура процеса обраде информација, састављена од великог броја међусобно повезаних елемената процесирања који заједнички раде на решењу проблема. Ове мреже, као и људи, уче на основу примера, тј историјског искуства, оне нису програмиране експлицитним инструкцијама, већ уче да изврше задатак користећи примере из стварног живота [9].

Вештачке неуронске мреже су у стању да препознају смисао у компликованим или непотпуним подацима, да препознају обрасце који су непрепознатљиви људима због своје сложености и који се не могу добити коришћењем других компјутерских техника. Вештачке неуронске мреже су софистициране технике моделирања, способне да моделирају веома комплексне функције.

Основна разлика према [9] између биолошких и вештачких неуронских мрежа је дата у табели 2.4.1.

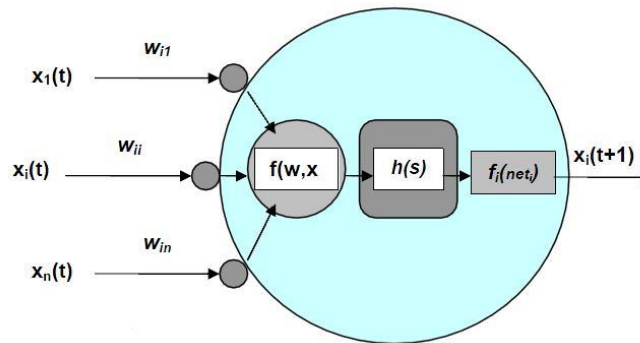
Табела 2.4.1.: Разлике биолошких и неуронских мрежа	
Неурони:	Рачунарски процесор:
Брзина – мала ($\sim 10^2$ Hz)	Брзина – велика ($\sim 10^9$ Hz)
Број – велики ($\sim 10^9$)	Број – мали (стотине)
Једноставан процесор	Комплексан процесор
Паралелна обрада	Секвенцијална обрада

Са инжењерског становишта биолошки неурони су слаби процесори и прилично су непоуздани. У мозгу ово је превазиђено великим бројем неурона и међусобних веза као и паралелном обрадом. Класични софтвер се може користити за унапред дефинисан скуп проблема, док неуронска мрежа има својство адаптивности па се може користити за најразличитије проблеме. Грубо речено, конвенционални рачунари су погодни за решавање добро дефинисаних задатака обраде информација у стабилном и савршено познатом

окружењу. Неуронске мреже су добре у обављању задатака када су подаци хетерогени, несређени, несигурни, па чак и неконзистентни и када не постоје савршена решења за практичне проблеме.

Основе неуронских мрежа

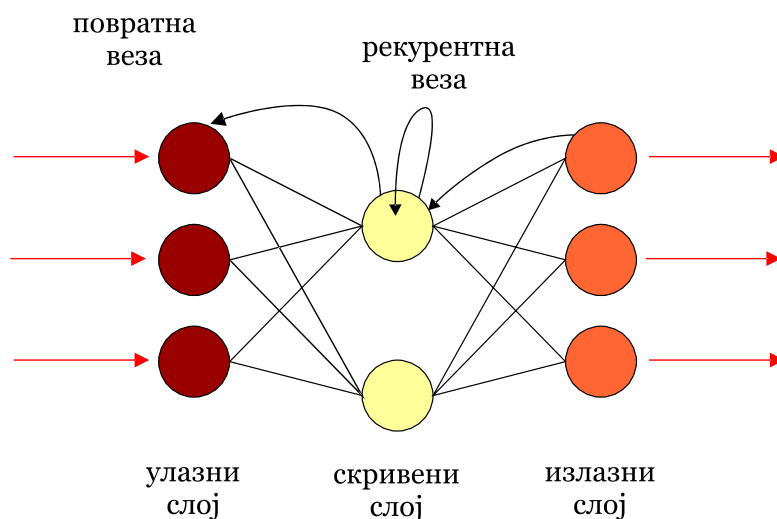
1. **Неурон** – прима улазе, процесира их и производи један излаз. Улаз може бити сирови податак или излаз из неког другог неурона. На наредној слици је приказано процесирање информације у вештачком неурону.



Слика 2.4.1.: Приказ типичног процесуирајућег елемента – неурона

Са слике видимо појмове које ћемо објаснити у наредних неколико редова:

- **Улазни оператор** $f(w,x)$, који обједињује улазе и тежинске односе међусобних веза w и формира јединствену вредност s (односно $s=f(w,x)=w^T \cdot x$), која је на тај начин припремљена за функцију преноса;
 - **Функција преноса** $h(s)$, која обрађује излаз из неуронског улазног оператора (врши интеграцију), формирајући потребну вредност за активациону функцију;
 - **Активациона функција** $f_i(net_i)$, која обрађујући излаз функције преноса управља излазном вредношћу неурона.
2. **Мрежа**. Свака мрежа се састоји од неурона груписаних у слојеве. Поред улазног и излазног слоја, мрежа може имати један или више скривених слојева. На слици је дат пример једне мреже са једним скривеним слојем.



Слика 2.4.2.: Вештачка неуронска мрежа

3. *Структура мреже.* Топологије мреже могу бити различите зависно од проблема који се решава, врсте улазних података, врсте излазних података и других фактора. То је разумљиво с обзиром да се неуронске мреже примењују у најразличитијим областима људске делатности.
4. *Процесирање информације у мрежи.* Основни концепти који се односе на процесирање су (још детаљније објашњени):
 - ❖ Улази – морају бити у нумеричком облику. Ако то није случај, морају се превести у нумеричке еквиваленције. Облик и врста улазних података могу бити различити и они у многоме одређују карактеристике мреже. Сваки улаз кореспондира једном атрибуту. На пример, код мреже која служи за изналагање могућности субјекта да врати кредит, улазни атрибут може бити величина месечног прихода.
 - ❖ Излази – представљају решење проблема. Мрежа може имати један или више излаза, зависно од конкретног проблема. Излаз може бити континуална вредност или квази логичка вредност, нпр +1 за "да", а 0 за "не".
 - ❖ Тежине – кључни елементи неуронске мреже. Мрежа учи подешавајући тежине према историјским подацима који се кроз њу пропуштају. Тежина изражава релативну значајност сваког улаза у неурон, она представља способност улаза да побуди неурон.
 - ❖ Функција сумирања- множи сваки улаз одговарајућом тежином, а затим сумира добијене умношке. Формула за n улаза у један неурон има следећи облик:

$$Y = \sum_{i=1}^n X_i W_i \quad (1)$$

Док за већи број неурона нпр j , формула гласи:

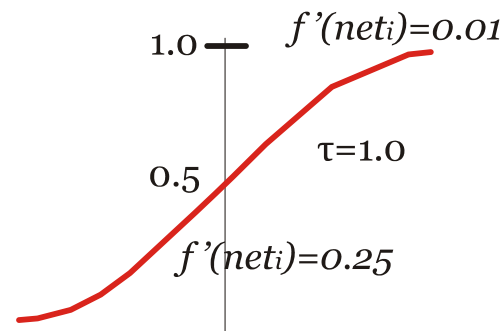
$$Y_j = \sum_{i=1}^n X_i W_{ji} \quad (2)$$

- ❖ Функција трансформације – одређује да ли резултат функције сумирања (интерна активизација неурона) може да произведе излаз. Веза између активизације и излаза може бити и линеарна и нелинеарна. Постоји више типова те функције, а сама чињеница да функција трансформације неурона скривених слојева може бити нелинеарна је од изузетне важности. У супротном, да је могућа само линеарна трансформација, мрежа би била у стању да моделује само линеарне функције, јер је познато да комбинација линеарних функција даје линеарну функцију. Зато је функција трансформације најчешће нелинеарна, а једна од најпознатијих је *сигмоидна* функција:

$$Y_t = \frac{1}{1 + e^{-Y}} \quad (3)$$

Где је Y_T нормализована вредност резултата функције сумирања Y . Нормализација значи да се овом трансформацијом излазни ниво доводи у границе између 0 и 1. Када не би било ове трансформације, излазна вредност би могла да буде исувише велика, нарочито у комплекснијим мрежама са више скривених слојева. Сигмоидна функција асимптотски тежи 0 и 1, па су све вредности преко 0.8 активни излази, а вредности испод 0.2 су неактивни излази. Чест је случај да се користи и функција *прага* (*threshold*) (нпр све изнад 0.5 постаје 1, а све испод 0.5 постаје 0).

На слици 2.4.3. је дат изглед сигмоидне функције.



Слика 2.4.3.: Сигмоидна активациона функција

- ❖ Учење – Неуронска мрежа учи из искуства. Процес учења се састоји из три задатка:
 - Рачунање излаза,
 - Упоређивање добијене вредности са жељеном,
 - Подешавање тежина и враћање на први корак.

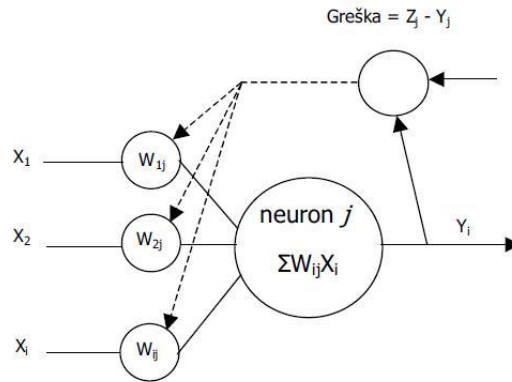
Пре почетка процеса учења постављају се иницијалне вредности тежинских односа случајно или по одређеном правилу. Са сваким новим скупом података, тежине ће се еподешавати тако да се смањује разлика између стварног и жељеног улаза. Ова разлика се назива *делта*. У најбољем случају делта треба да се сведе на нулу, међутим у пракси обично је случај да се унапред задаје ниво грешке (нпр 5%) који треба постићи да би се окончао процес учења или се унапред задаје број итерација при учењу. Кључ је да промене тежина иду у правом смеру (да конвергирају), јер се само тако делта смањује.

Најраспрострањенији модели вештачких неуронских мрежа су према [1]:

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

Backpropagation је скраћеница од *back error propagation* што би могло да се преведе као "ширење грешке уназад". Ово је највише коришћен алгоритам учења и прилично је једноставан за имплементацију. Да би се мрежа која учи на основу овог алгоритма користила, потребан је претходно обучавање, односно оптимизација параметара мреже. *Backpropagation* типа feed-forward и има један или више скривених слојева.

Почевши од излазног слоја, разлика (грешка) између жељених и стварних излаза се рачуна да би се на основу ње поправиле вредности тежина у претходном слоју. Принцип се јасно види на слици 2.4.4.



Слика 2.4.4.: Backpropagation - принцип функционисања

За сваки неурон излазног слоја, грешка (делта) се израчунава као:

$$(\text{делта}) = (Z_j - Y_j) \left(\frac{df}{dx} \right) \quad (4)$$

Где су Z и Y , жељени и стварни излаз, респективно. Сигмоидна функција $f = [1 + \exp(-x)]^{-1}$ је ефикасан начин да се израчуна излаз неурона (где је x пропорционално суми умножака и тежина). Извод сигмоидне функције (df/dx) и грешка су једноставна функција жељених и стварних излаза. Фактор $f(1-f)$ је *логистичка функција* која држи корекцију грешке у одређеним границама. Тежина сваког улаза у j -ти неурон се поправља пропорционално овој израчунатој грешци.

Процедура овог алгоритма се састоји из следећих корака:

1. Иницијализација тежина (случајна).
2. Учитавање улазног вектора и жељеног излаза.
3. Пропуштање улазног вектора кроз мрежу и добијање израчунатог излаза.
4. Израчунавање грешке.
5. Подешавање тежина идући уназад од излазног слоја ка скривеним слојевима.

Процедура се понавља за све улазне векторе све док се жељени и стварни улази не поклопе, узимајући у обзир неку толеранцију грешке. Великој мрежи може бити потребно доста времена за подешавање, тако да се понекад унапред задаје број итерација учења уместо толеранције. Са сваким новим скривеним слојем, време учења се знатно повећава.

Овај алгоритам учења је прилично једноставан и користи се често. Једна од замерки је да је прилично спор, па су развијени модалитети са хеуристичким побољшањима, какви су *Quickpropagation*, *Delta-Bar-Delta* и други. []

2.5. Калманов филтер

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

$$x_k = \mathbf{A}x_{k-1} + \mathbf{B}u_{k-1} + w_{k-1} \quad (5)$$

$$z_k = \mathbf{H}x_k + v_k \quad (6)$$

Ознака x је n -димензионална променљива стања система. Ради се о веома битној величини стања коју није могуће директно измерити. Одређивање те величине је задатак естиматора. Прва једначина говори о односу између променљивих стања у тренутном и претходном кораку. Може се видети да тренутно x зависи од претходне вредности променљиве x помножене са $n \times n$ димензионалном матрицом \mathbf{A} . Даље, тренутно стање зависи од улаза у систем из претходног корака представљеног l -димензионалним вектором u , помноженог са матрицом \mathbf{B} димензија $n \times l$. Осим ових детерминистичких чланова, тренутно стање зависи од случајног члана w . Ради се о случајном вектору којим се моделира шум датог процеса. Реч је о белом шуму (фреквенцијски независан, временски неограничен) с нормалном расподелом: $p(w) \sim N(0, \mathbf{Q})$. \mathbf{Q} је матрица коваријанси шума процеса. Може се одредити поступком идентификације процеса. [6]

Излаз система је представљен m -димензионалним вектором z који представља величину стања коју је могуће измерити. Излаз је детерминистички повезан са променљивом стања преко матрице \mathbf{H} димензије $m \times n$. Као и код претходне једначине стања и овде се појављује стохастичка величина. Ради се о белом шуму који је присутан при мерењу. Моделиран је вектором v , не зависи од шума w и расподељен је по нормалној расподели са коваријанцом \mathbf{R} ; $p(v) \sim N(0, \mathbf{R})$. Вредност \mathbf{R} се одређује експериментално.

Као што се види из горњег модела, систем и мерење су „загушени” шумом, што онемогућава доношење поузданог закључка о тренутној вредности променљиве стања x овог система. Задатак Калмановог филтера је управо да процени ту вредност. Уопштено говорећи, филтер се ради у два корака:

1. **Предикција** – на основу последњег познатог стања $k - 1$ предвиђа се следеће стање k . Ова процена садржи грешку због шума у систему који притом није узет у обзир.
2. **Корекција** – након што се стање стварно промени у следеће, мери се излаз система (вектор z у стању k) и на основу тог резултата са шумом, матрица која описује понашање процеса и података о процесу коригује процену стања x .

Након тога, цели циклус се понавља: предвиђа се следеће стање, мери се излаз, коригује процена следећег стања. Мерење, дакле, дели алгоритам реализације Калмановог филтера на два дела: део пре мерења (процена стања а priori) и након мерења (процена стања уз познато мерење, а постериори). У наставку, величине добијене пре мерења ознабене су знаком „-”.

Први корак алгоритма може се описати следећим једначинама:

$$\hat{x}_k^- = \mathbf{A}\hat{x}_{k-1}^- + \mathbf{B}u_{k-1} \quad (7)$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (8)$$

У првом кораку на основу детерминистичке једначине стања система процењује се вредност променљиве стања система. Ова процена доноси се само на основу вредности а постериори естимације променљиве стања из претходног корака (најпрецизнија естимација) и улаза из претходног корака. Квалитет ове процене зависи од квалитета модела којим је процес описан, као и утицају шума (моделирано вектором w) који је у овој процени изостављен у стварном процесу. Будући да се овим кораком на основу претходних величина може предвидети стање система у тренутном кораку, овај корак може се назвати предикцијом (предвиђа следеће стање система уз познато претходно стање). Разликом стварне и а priori процене стања се дефинише а priori грешка естимације:

$$e_k^- \equiv x_k - \hat{x}_k^- \quad (9)$$

На основу ове грешке може се дефинисати променљива грешке а priori естимације:

$$\mathbf{P}_k^- = E[e_k^-, e_k^{-T}] \quad (10)$$

Где E представља оператор математичког очекивања.

Једначина (8) се односи на мреже и показује како се на темељу а постериори променљиве грешке из претходног корака и променљиве шума система може израчунати променљива грешке а priori естимације за тренутни корак. Ово је потребно за други корак – корекцију.

Други корак алгоритма своди се на корекцију естимације на темељу изведеног мерења. Мерење, дакле, игра улогу повратне спреге. Овај корак описује једначина:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (11)$$

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k (z_k - \mathbf{H} x_k^-) \quad (12)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (13)$$

Средња једначина у овом кораку је једначина (12). Она описује корекцију естимације: а priori вредности естимације стања процеса додаје се корекцијски члан $\mathbf{K}_k (z_k - \mathbf{H} x_k^-)$ чиме се побољшава естимација променљиве x у стању k . Разлика $(z_k - \mathbf{H} x_k^-)$ назива се иновација мерења и њоме се моделира разлика (одступање) између стварног мерења z_k и теоријски предвиђеног резултата мерења који следи из једначина процеса: $\mathbf{H} x_k^-$. \mathbf{K}_k је фактор који се још назива Калманово појачање. Ради се о матрици димензија $n \times m$ чији је задатак да у а постериори фази минимизира променљиву грешке естимације \mathbf{P}_k , која је дефинисана као:

$$\mathbf{P}_k = E[e_k e_k^T] \quad (14)$$

Где је

$$e_k \equiv x_k - \hat{x}_k \quad (15)$$

Може се доказати да \mathbf{K}_k зависи од а priori вредности грешке \mathbf{P}_k^- , матрици \mathbf{H} и променљиве шума мерења на начин описан формулом (11). Даљом анализом види се да важе релације:

$$\lim_{R_k \rightarrow 0} \mathbf{K}_k = \mathbf{H}^{-1} \quad (16)$$

$$\lim_{P_k^- \rightarrow 0} \mathbf{K}_k = 0 \quad (17)$$

Другим речима, ако се смање грешке мерења, Калманово појачање поприма такав облик који у једначини корекције наглашава резултат стварног мерења z_k јер оно постаје ближе стварној вредности улаза него предвиђена вредност $\mathbf{H} \hat{x}_k^-$. Такође, уколико променљива грешке а priori естимације тежи нули, појачање такође тежи нули, што значи да а priori естимација стања постаје блиска коригованој естимацији. Односно, што је предикција била боље, то ће корекција а priori стања бити мања.

С обзиром да ће у следећем кораку предикције (за следеће стање) опет бити потребно познавање променљиве грешке из претходног корака (\mathbf{P}_k из овог корака ће се у следећем циклусу употребити као \mathbf{P}_{k-1}), на крају прорачуна се рачуна \mathbf{P}_k према формули (13).

Дакле, Калманов филтер је оптимални рекурзивни филтер који ради искључиво у временском (тј. временски дискретном) домену. Одликује га зависност само о једном претходном кораку (k зависи само од $k-1$). Може се користити за интерполацију претходних стања, естимацију тренутног, као и предвиђање следећег стања система. Постоји више облика Калмановог филтера. Тако се поједностављена верзија у којој се уместо матрица променљивих користе скаларне константе назива алфа-бета филтером. Постоји и проширена верзија у којој се управља нелинеарним системом. Овакав проширени Калманов филтер назива се Шмитовим филтером.

3. Концепцијско решење пројектног задатка

Задатак се састоји у успешном решавању кретања робота кроз погон фабрике, односно опслуживања машина материјалима, и сировинама, као што је већ раније поменуто. Робот на основу маркера који су на подлози залепљени одређује позицију, а затим, уколико је потребно врши преоријентацију, како би наставио пут ка жељеној дестинацији. Оно што робот мора да испуни јесте његово сопствено "знање" о положају у коме се тренутно налази и где се наредна позиција налази у односу на његову тренутну позицију.

За овакву врсту проблема постоји неколико решења која се огледају у коришћењу експертних система, правила базираних на индукцији, или техника *soft-computing-a: fuzzy* системи, генетички логаритми.

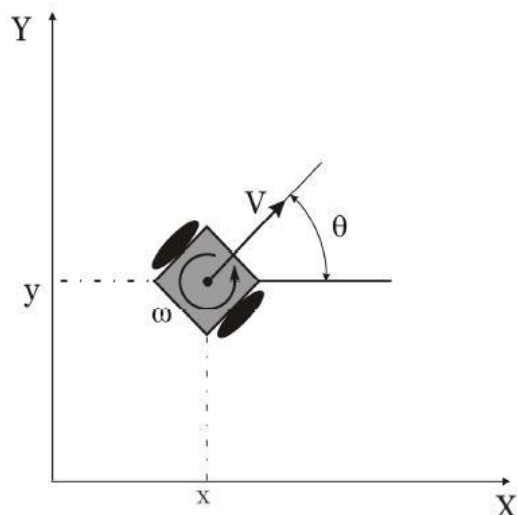
Ми нажалост нисмо имали времена да обрадимо све могуће приступе, већ смо прешли на конкретна решења која смо добили захваљујући коришћењу вештачких неуронских мрежа.

Други део задатка се односио на диспозицију технолошког окружења "XY". За дато предузеће било је потребно развити симулациони модел у *Anylogic* окружењу, а затим на основурезултата симулације предложити и нови диспозициони план технолошког окружења и формирати "троугаону" матирцу за нови модел, ако је то могуће. Овај део задатка је урађен према подацима о позицији машина у предузећу "XY" који се виде на слици 2.1.1. и подацима о самим машинама које су дате у поглављу 2.1.. Наиме, за затечено стање фабрике смо урадили симулацију производње, а потом смо за другачији распоред машина и транспортних токова урадили исту ствар. Резултати су дати у поглављу 4.2..

4. Пројектно решење

4.1. Мобилни интелигентни робот

Да би робот могао да се креће, онако како ми то желимо, прво што смо урадили јесте да смо успоставили комуникацију између робота и рачунара. Већ је раније поменуто да смо користили *Matlab* као главни софтвер за управљање роботом. За почетну контролу положаја робота, као и пређеног пута, користили смо модел кретања који се односи на пређени пут. Захваљујући том моделу, имамо приближну информацију о томе где се робот тренутно налази у односу на координатни почетак, као и како је оријентисан.



Слика 4.1.: Базни модел кретања робота

Табела 4.1.: Једначине које одређују модел кретања

$x' = \{x \ y \ \theta\}^T$	(18)
$\Delta x = \Delta s \cos\left(\theta + \frac{\Delta\theta}{2}\right)$	(19)
$\Delta y = \Delta s \sin\left(\theta + \frac{\Delta\theta}{2}\right)$	(20)
$\Delta\theta = \frac{\Delta s_d - \Delta s_l}{b}$	(21)
$\Delta s = \frac{\Delta s_d + \Delta s_l}{2}$	(22)
$x' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_d + \Delta s_l}{2} \cdot \cos\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \\ \frac{\Delta s_d + \Delta s_l}{2} \cdot \sin\left(\theta + \frac{\Delta s_d - \Delta s_l}{2b}\right) \\ \frac{\Delta s_d - \Delta s_l}{2b} \end{bmatrix}$	(23)
<p>Где су: $\Delta x, \Delta y, \Delta\theta$ – величине које одређују прираштај $\Delta s_d, \Delta s_l$ – пређени пут десног и левог погонског точка респективно b – растојање између точкова</p>	

Уносећи наше вредности у последњу матрицу, добили смо оперативне вредности за израчунавање тренутног положаја робота и његову оријентацију.

Следећи корак је имплементација Јакобијан матрице, на основу које ћемо у сваком тренутку знати где се налази енд-ефектор робота (у нашем случају оптички сензор) у односу на тежишну тачку робота или у односу на координатни почетак радног простора.

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{bmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \mu_{t-1,\theta}} \end{bmatrix} \quad (25)$$

Наредни корак је био усаглашавање контроле мотора за конфигурацију робота коју смо ми осмислили. У код који смо добили потребно је убацили полупречник точка, како би читавање са енкодера било валидно. У истом делу програма су та читавања претварана у пређени пут како би била лакша за разумевање нама као корисницима.

Део кода који је за то задужен је дат у наредних пар редова:

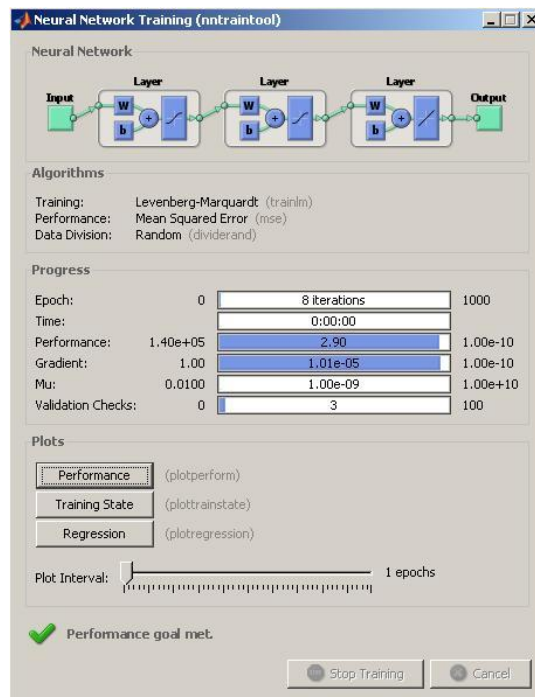
```
% ocitati vrednosti sa oba enkodera
MC = GetMotorSettings(MOTOR_C)
MB = GetMotorSettings(MOTOR_B)
% pretvoriti ocitavanja sa enkodera u predjeni put
dsl = MC.Angle/57.3*r
dsr = MB.Angle/57.3*r
```

Корак који је потом уследио је био обучавање вештачке неуронске мреже за одређивање закретања вратила мотора за жељене вредности углова. Мерења су рађена за вредности углова које су дате у табели 4.2. Мрежа коју смо обучавали за ову врсту експериментисања је се зове "Steering Control Net" или краће *Scnet*. *Scnet* је у ствари неуронска мрежа са простирањем сигнала унапред "feedforward", која користи *Levenberg-Marquardt*-ов алгоритам учења "trainlm", сигмоидну активациону функцију "transig" и поступак минималног градијента "learngdm". Приликом обучавања мреже, мењали смо њену конфигурацију више пута, како би добили оптималне резултате. Резултати обучавања су дати у табели 4.2.

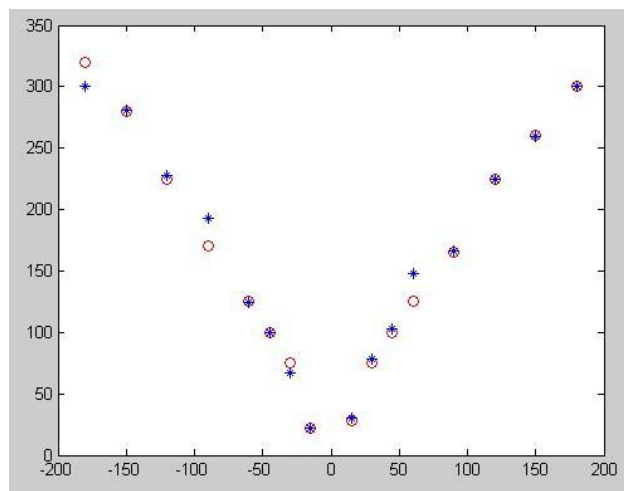
Табела 4.2. Вредности мерења AngleLimit у зависности од жељеног угла			
Жељена вредност угла	AngleLimit	Вредности обучене мреже	TurnRatio
+15°	28	30.6413	-100
-15°	22	22.3039	+100
+30°	75	78.6212	-100
-30°	75	66.5885	+100
+45°	100	102.3562	-100
-45°	100	100.1465	+100
+60°	125	148.2897	-100
-60°	125	124.1484	+100
+90°	165	166.3336	-100
-90°	170	193.0160	+100

+120°	225	224.1660	-100
-120°	225	228.1629	+100
+150°	260	259.5482	-100
-150°	280	281.2593	+100
+180°	300	299.9575	-100
-180°	320	299.9075	+100

Графички приказано, обучавање мреже је дато на слици 4.2., а резултати су дати на слици 4.3. Као што се може видети са слике 4.2. процес обучавања је завршен после 8 итерација, од максимално 1000 колико смо задали.



Слика 4.2.: Процес обучавања мреже



Слика 4.3.: Резултати обучавања мреже

Код у MATLAB-у је дат у наставку:

```
clc,clear
close all
% SetPower = const. = 20 => uvek!
% TurnRatio = 100 => pozitivan mat. smer = -100
%                               negativan mat. smer = 100

% Merenja su uradjena za sledece uglove:
% +/- [15 -15 30 -30 45 -45 60 -60 90 -90 120 -120 150 -150 180 -180]

AL = [28 22 75 75 100 100 125 125 165 170 225 225 260 280 300 320]
input = [15 -15 30 -30 45 -45 60 -60 90 -90 120 -120 150 -150 180 -180]

output = [AL]

SCnet_PR = newff(input,output,[10 6],{'tansig'},'trainlm','learngdm')

SCnet_PR.trainParam.show = 50;
SCnet_PR.trainParam.lr = 0.05;
SCnet_PR.trainParam.mc = 0.9;
SCnet_PR.trainParam.mu = 0.01
SCnet_PR.trainParam.epochs = 1000;
SCnet_PR.trainParam.goal = 1e-10;
SCnet_PR.trainParam.max_fail = 100;

SCnet_PR = train(SCnet_PR,input,output)
y = sim(SCnet_PR,input)

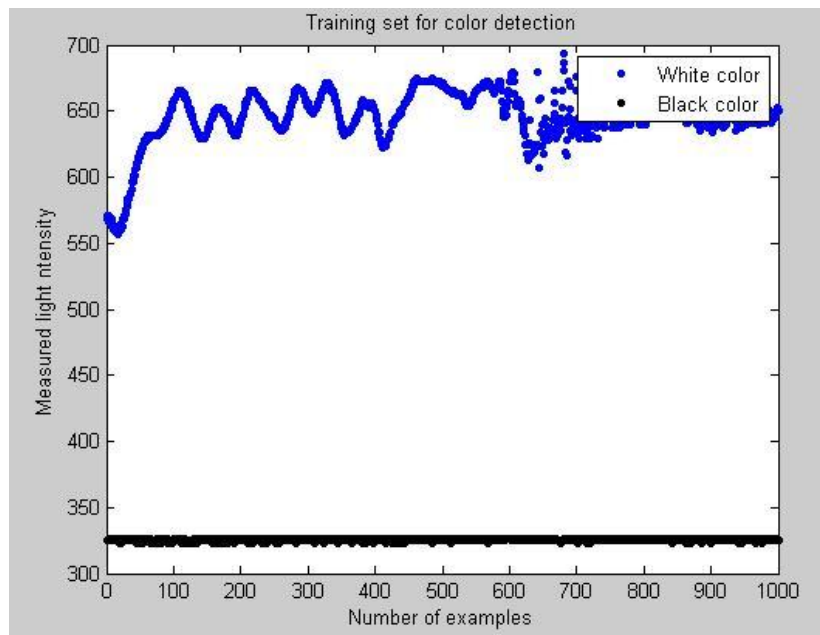
disp('y AL input ')
mreza_vs_AL=[y' AL' input' ]

figure(1),
plot(input,AL,'or'),hold on
plot(input,y,'b*')
% SCnet_PR2 = SCnet_PR
save SCnet_PR
```

Као што се може видети из кода, одлучили смо се за решење које има два скривена слоја и то 10 у првом и 6 у другом и са њим смо добили задовољавајуће резултате.

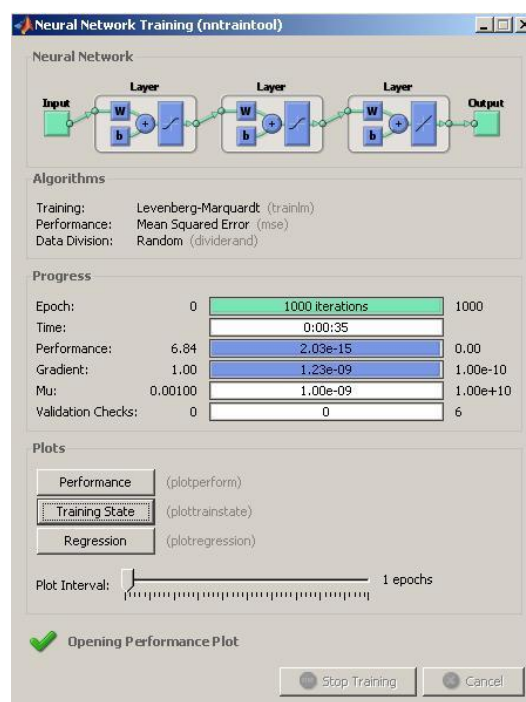
Следећа ставка је била мерење интензитета рефлексије о црну и белу површину. За обе боје смо радили обучавање вештачким неуронским мрежама, како би робот знао да се заустави или да се преоријентише кад наиђе на маркер који се налази на поду постројења. Идеја је да се робот креће кроз постројење тако да не удари ни у једну машину. Односно, када робот буде пролазио кроз те тачке (маркере), оптичким сензором ће покупити информацију да се налази у једној од референтних тачака, а потом ће коришћењем Калмановог филтера и вештачких неуронских мрежа рачунати тренутни положај, оријентацију и следећу тачку у коју треба да крене, а коју ми на почетку задајемо.

Резултати мерења интензитета црне и беле боје дати су на слици 4.4.



Слика 4.4.: Резултати мерења црне и беле боје оптичким сензором

За препознавање боја, користили смо мрежу са простирањем сигнала унапред "feedforward" са два скривена слоја. У наставку је дат део кода у коме се види да смо мрежу за препознавање боја такође тренирали са *Levenberg-Marquardt*-ов алгоритмом учења "trainlm". У првом скривеном слоју постоје два, а у другом пет неурона. Приликом обучавања мреже, као улаз су дате вредности које су очитане са сензора за белу и црну боју, а као излаз смо добили препознавање боје од стране робота током кретања у радном простору. За разлику од обучавања мрежа за окретање вратила приликом одређивања угла, у овом случају број итерација је дошао до максималне постављене цифре од 1000 итерација. То је са једне стране било и очекивано, јер је улазни вектор за боју био доста већи у односу на улазни вектор мреже за углове. На слици 4.5. је дат процес обучавања мреже за препознавање боја.



Слика 4.5.: Процес обучавања мреже за препознавање боје

4.2. Anylogic окружење

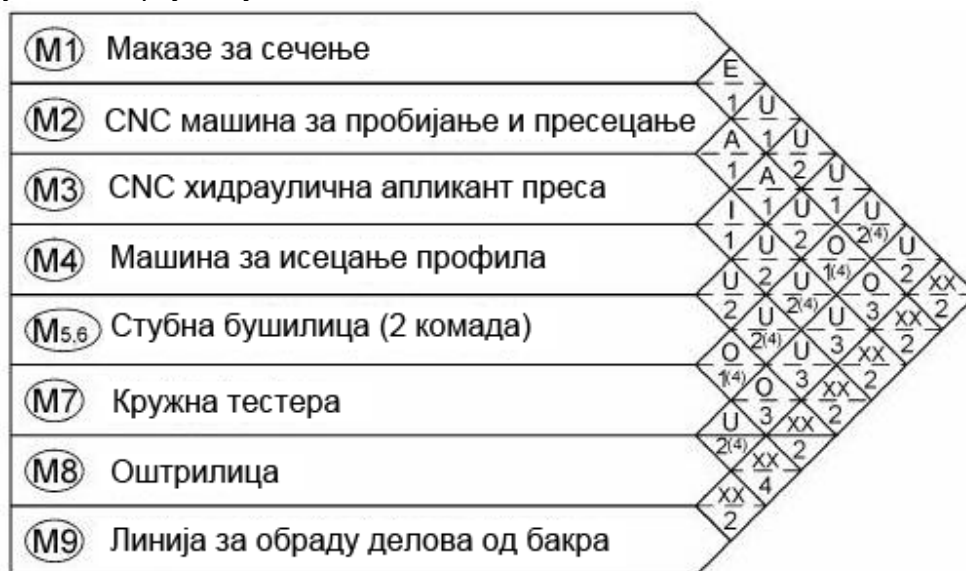
Anylogic је алат за моделирање динамичке симулације које обухвата динамику система, централно процесне (дискретни догађаји), и агентно базиране приступе у оквиру једног језика за моделовање и једног развојног окружења модела. Језик *Anylogic* нам омогућава да проучимо комплексност и хетерогеност пословања, економије и социјалних система у било ком жељеном нивоу детаља. *Anylogic-ов* скуп примитивних и "*library*" објеката нам омогућава моделирање производње и логистике, пословних процеса, људских ресурса, потрошача и понашања пацијената, као и животну средину у њиховој природној интеракцији.

Ми смо овај алат користили за симулацију производње у предузећу "XY" за параметре машина које смо добили. Табела 4.2.1. приказује времена обраде делова за сваки део понаособ (у секундама) и растојање између машина или складишта (у метрима). Потребно је напоменути да су сва мерења растојања овог пута рађена за машине које користимо у овом моделирању. Такође, програм нам је на једноставан начин показао кретање делова кроз погон, тј јасно се види на симулацији колико је неки конвејер оптерећен у датом тренутку.

Табела 4.2.1. Параметри машина и растојања између њих				
Машина и растојања	Transport fuse	Mainbusbar support	Support d800	Busbar 2L1
Растојање од складишта до машине М2	24.5	-	-	-
ЦНЦ машина за пробијање и просецање	6+2	-	-	-
Растојање од машине М2 до машине М3	11.4	-	-	-
Растојање од складишта до машине М3	-	-	31.2	-
ЦНЦ хидраулична апкант преса	2+2	-	4+4+4+2+2	-
Растојање од машине М3 до машине М4	15.7	-	11	-
Машина за исецање профила	2.5	-	32+37+6+3.5+18	-
Растојање од М4 до складишта	18.7	-	18.7	-
Растојање од складишта до машине М5	40.3	38.9	37.7	-
Стубна бушилица	-	10+6	-	-

Растојање од машине М5 до машине М7	-	3.9	-	-
Растојање од М7 до складишта	-	38.8	-	-
Растојање од складишта до машине М9	-	-	-	26.3
Линија за обраду делова од бакра	-	-	-	2+23+4+7+8+5

Сами односи између машина су дефинисани "троугаоном" матрицом, где се види зависност машина једне од друге и разлог зависности. Ову матрицу нисмо били у стању да мењамо, јер бисмо променом матрице само променили односе који владају у постројењу, што би у преводу значило да би се технолошки поступак обраде морао променити, а то не би било пожељно, јер је нама технологија обраде делова већ дата задатком. Троугаона матрица тренутног постројења је дата сликом 4.2.1.



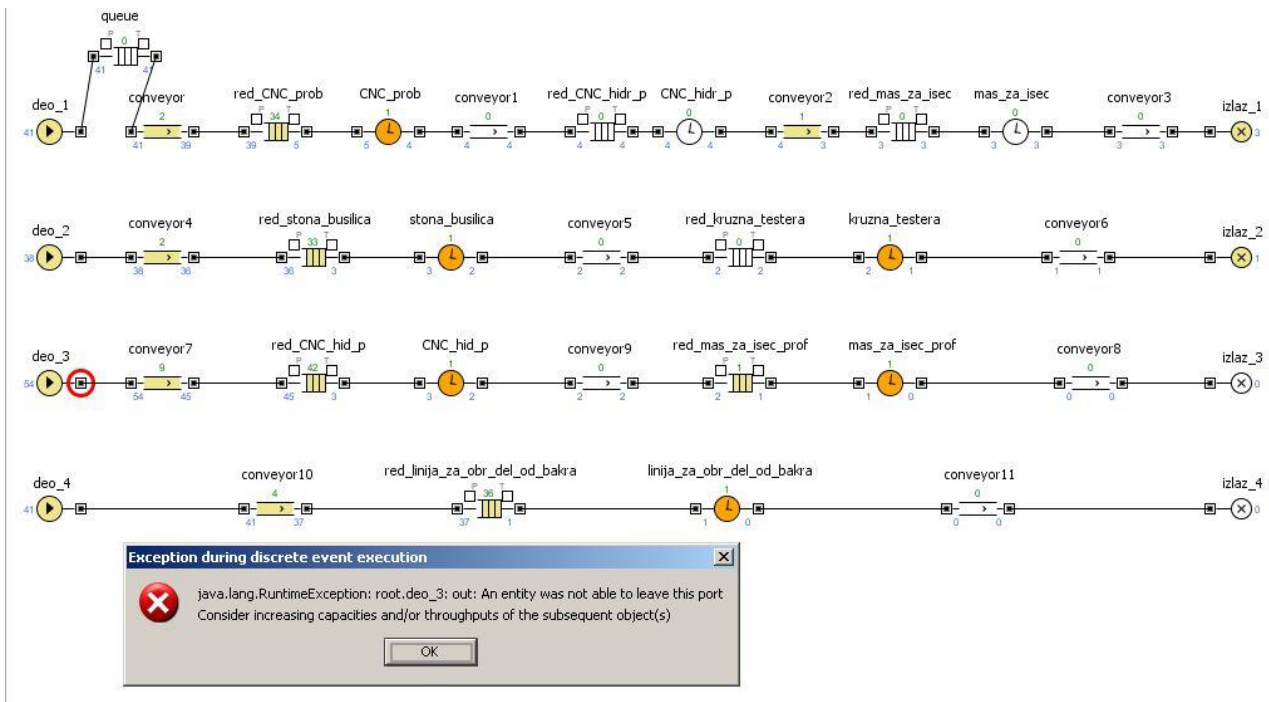
ознака	Степен зависности
A	Апсолутно неопходно
E	Веома важно
I	Важно
O	Потребно
U	Неважно
X	Непожељно
XX	Веома непожељно

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

Слика 4.2.1.: Приказ "троугаоне" матрице за предузеће "XY"

Мобилни интелигентни робот је графички приказан као конвејер, јер он и има функцију преноса материјала до машина.

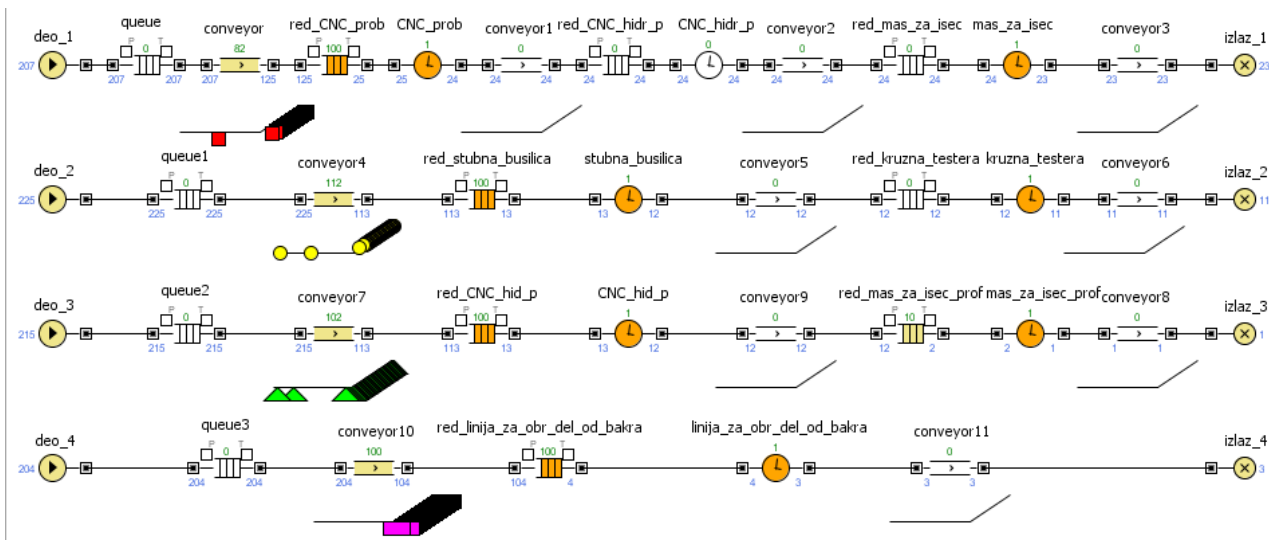
На слици 4.2.2. дат је изглед прве симулације.



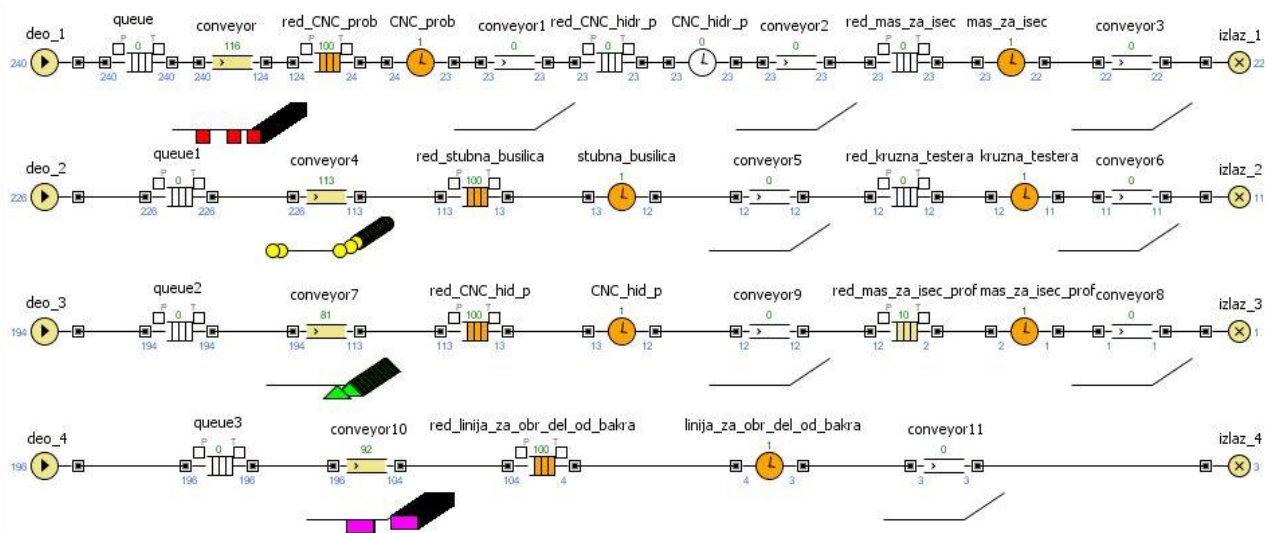
Слика 4.2.2.: привремени резултати прве симулације у Anylogic окружењу

Са слике се види да постоји грешка. Утврђено је да је грешка настала због недостатка додатног конвејера на улазу сваког од делова. Грешка се јавља зато што се делови нагомилavaju у реду испред машине толико да долази до загушења. У том првом случају није разматрана могућност да постоје конвејери који доводе материјал од складишта до машине, већ је сматрано да се делови одмах стављају у ред машине. Убацаивањем конвејера који су имали дужине које су наведене у табели 4.2.1., овај проблем је решен.

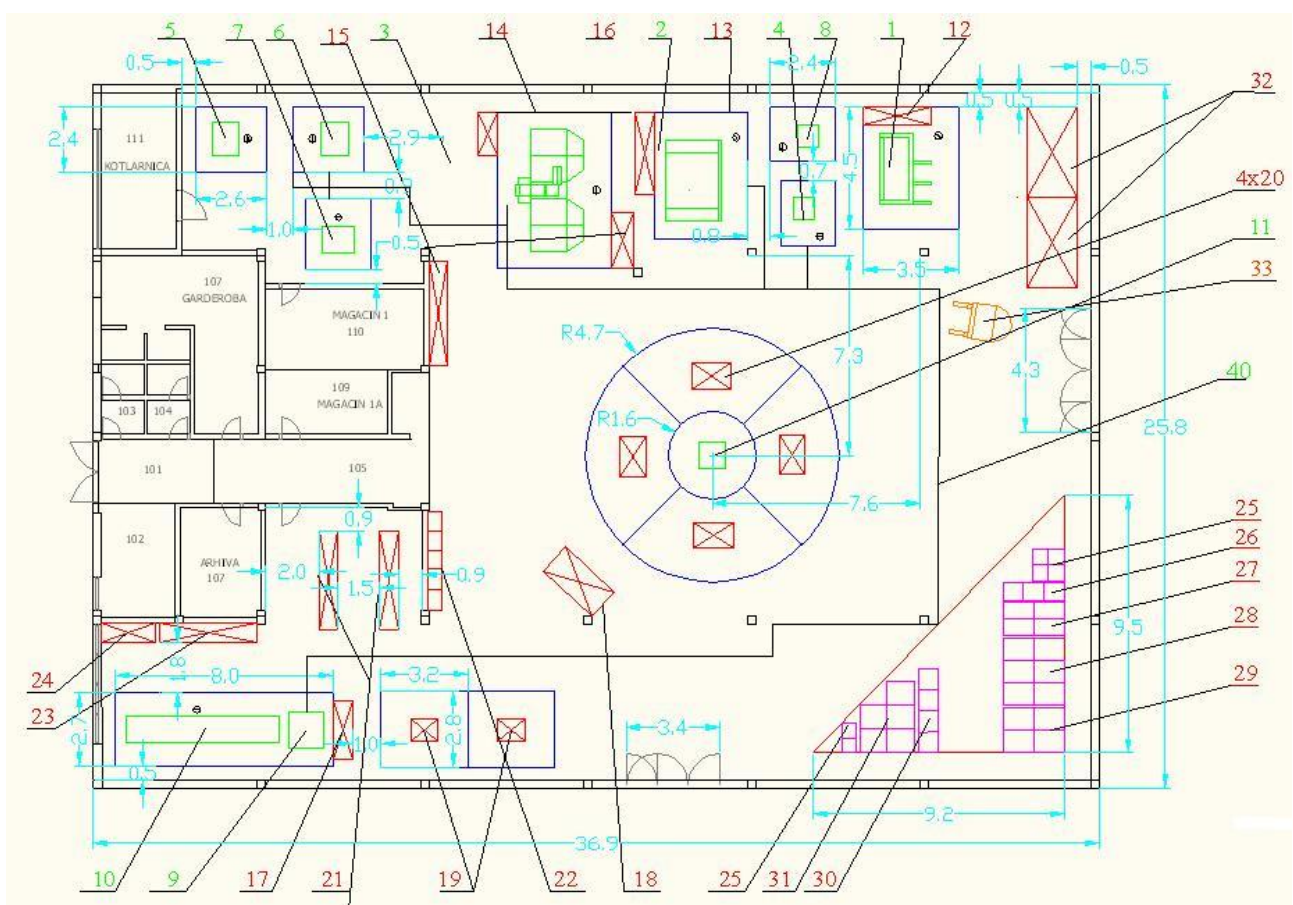
Коначни резултати прве и друге диспозиције су дати на сликама 4.2.3. и 4.2.4..



Слика 4.2.3.: коначни резултати прве симулације у Anylogic окружењу



Слика 4.2.4.: коначни резултати друге симулације у AnyLogic окружењу



Слика 4.2.5.: друга диспозиција фабричког окружења

На слици 4.2.5. су позицијом 40 показани транспортни путеви у фабрици. То су путеви који су у симулацији представљени конвејерима, а које у ствари треба да преже мобилни робот како би опслужио машине.

За период од 200 секунди колико траје симулација, добили смо и резултате у виду готових обрађених делова, делова који се тренутно обрађују у погону, делова који се налазе на конвејерима, колико делова чека у реду и слично. Резултати су дати у табели 4.2.2.

Табела 4.2.2.: резултати симулације у Anylogic софтверу

Елемент симулације	Број обрађених делова у првој симулацији	Број обрађених делова у другој симулацији	Процентуална разлика У односу на подразумеване вредности
Улаз 1	207	240	+15%
Улаз 2	225	226	+0.44%
Улаз 3	215	194	-10.77%
Улаз 4	204	196	-3.97%
Конвејер 01	125	124	-0.8%
Конвејер 02	24	23	-4.16%
Конвејер 03	24	23	-4.16%
Конвејер 04	23	22	-4.34%
Конвејер 05	113	113	0%
Конвејер 06	12	12	0%
Конвејер 07	11	11	0%
Конвејер 08	113	113	0%
Конвејер 09	12	12	0%
Конвејер 10	1	1	0%
Конвејер 11	104	104	0%
Конвејер 12	3	3	0%
Ред 01	0	0	0%
Ред 02	100	100	0%
Ред 03	0	0	0%
Ред 04	0	0	0%
Ред 05	0	0	0%
Ред 06	100	100	0%
Ред 07	0	0	0%
Ред 08	0	0	0%
Ред 09	100	100	0%
Ред 10	10	10	0%
Ред 11	0	0	0%
Ред 12	100	100	0%
ЦНЦ машина за пробијање и просецање	24	23	-4.16%
ЦНЦ хидраулична апкант преса	24	23	-4.16%
Машина за исецање профила	23	22	-4.34%
Стубна бушилица	12	12	0%
Кружна тестера	11	11	0%
ЦНЦ хидраулична апкант преса2	12	12	0%
Машина за исецање профила2	1	1	0%
Линија за обраду делова од бакра	3	3	0%
Излаз 1	23	22	-4.34%
Излаз 2	11	11	0%
Излаз 3	1	1	0%
Излаз 4	3	3	0%

Иако је за промену диспозиције број унетих делова у систем већи за највише 15%, то се није битно одразило на коначне резултате симулације који су нама најважнији. Из статистике се види да се променом распореда машина нису добиле велике промене на излазу из система. Наиме, једина промена за узрочно време од 200 секунди колико траје симулација, је смањење броја обрађених делова за 1 (процентуално 4.34%). То нам говори да је претходни распоред машина, односно онај је подразумеван ипак најбољи за употребу.

5. Експериментални резултати

После обаљених испитивања закључује се да концепцијско решење даје задовољавајуће резултате. Циљ се састојао од тога да робот препозна референтну тачку у којој се налази и да настави да се креће ка наредној задатој тачки. Грешка која је настала при обучавању мреже није износила више од 5°. Главни разлог који може да доведе до грешке и неправилног реаговања робота је несавршеност самих делова робота, првенствено мотора.

Највећи недостатак у целом експерименту било је конструкцијско решење робота. Робот није превиђен за професионално коришћење, зато је несавршеност појединих делова, као што су нпр енкодери, очигледна и прихватљива. Од робота се не очекује да извршава прецизне задатке и кретања, што је доказано и ценом самог робота и делова.

За позиционирање робота и његово кретање одговорни су погонски мотори. У нашем случају постоје два погонска мотора. На вратило мотора директно је постављен точак. Тако да имамо погон на два точка (*differential drive*). Трећи точак је мањи од прва два и налази се на задњем делу робота. Служи да одржи стабилност робота и омогући његово скретање. Скретање се врши тако што се један мотор окреће у једном смеру, а други у супротном. Ово омогућава роботу да ротира око своје осе. Тиме је угао заокретања сведен на минимум. Овако је изгледало концепцијско решење конструкције робота.

Проблеми су се прво јавили код мотора. У питању су обични мотори, а за прецизну контролу ротације вратила, а самим тим и прецизно кретање, потребни су степ мотори. Тако да и након команде да се мотори зауставе, они под силом инерције наставе да се крећу. Пошто су мале брзине кретања у питању ова грешка није велика, али робот у току обављања задатака и избегавања препрека константно покреће и зауставља моторе. Грешка се у овом случају стално акумулира, што на крају доводи до тога, да робот не дође до жељеног положаја. Овај проблем се решава дефинисањем карактеристичних објеката у окружењу које робот када препозна, може да одреди тачно свој положај и тиме поништи претходно акумулирану грешку. Из истих разлога имали смо проблема и при скретању.

Пошто је цела конструкција робота лако растављива и не постоје завртњи, током кретања осциловање и увијање конструкције може да утиче на кретање и да доведе до грешке. У овом случају грешке су мале, па смо их ми у нашем раду занемарили.

На прецизност при кретању у великој мери утиче и трећи точак, који може да проклиза и тиме онемогући заокретање под жељеним углом.

6. Анализа и дискусија

Задатак овог пројекта је успешно обављен, тј робот пролази кроз задате тачке са жељеном оријентацијом, а самим тим и опслужује машине које се налазе у постројењу предузећа. Успешност проласка робота је директно условљен обученим вештачким неуронским мрежама (што је боља мрежа, то је прецизност већа). У претходним тачкама је описано да су једине улазне величине у ствари подаци са оптичког сензора који се налази испред самог робота, прецизније на његовој предњој страни. За цену оптичког сензора кога имамо на коришћење добили смо сасвим солидна читавања у динамичком окружењу, мада се дешавало да због несавршености склопа, робот иако прође кроз маркер, не препозна црну боју. Та грешка се може снизити на минимални ниво инсталирањем скупље опреме, са рецимо ласерским препознавањем подлоге.

Примарни циљ пројекта је био да се у експерименталном моделу технолошког окружења изврши имплементација интелигентног мобилног робота. Секундарни циљ пројекта је утврђивање степена успешности примене вештачке неуронске мреже на интелигентне мобилне роботе. Као што је речено у поглављу 2.4., робот је почео да учи и резонује, што значи да је у стању и да решава проблеме секвенцијално.

Проблем настаје када се испред робота нађе чврста препрека. Робот није у стању да је препозна на основу само једног оптичког сензора, па се мора прибећи неком другом решењу.

Једно од решења које је могуће извести је да се на робота дода ултразвучни сензор (сонар) који би имао представу о препрекама у окружењу и затим се сналазио у простору. Такво решење би у овом случају било комплетно, јер би робот имао представу шта се испред њега налази, а оптичким сензором би јављао и да ли је на правом месту. У пакету, који смо добили на коришћење приликом вежбања, се налази један такав сонар. Време је било ограничавајући фактор и само због тога нисмо били у стању да имплементирамо сонар на робота, како би решење било потпуно.

7. Закључак

Циљ овог пројекта који треба постићи је да интелигентни мобилни робот, који има унапред дефинисану путању и редослед опслуживања машина, буде у могућности да прође кроз маркере и препозна их, како би знао где се налази и где тек треба да иде. Остваривање свести робота о његовом тренутном положају и наредној позицији смо постигли применом Калмановог филтера. Сврха неуронске мреже у овом случају је да препозна маркере и одреди углове закретања робота за наредну тачку, како би прошао кроз исту. Примена вештачких неуронских мрежа има широку заступљеност код интелигентних мобилних робота, највише због њихове способности да нађу корелацију између улазних и жељених излазних података. Метода која је коришћена базира се на машинском учењу, применом система вештачких неуронских мрежа, због њихових могућности да се прилагоде променама, које у систему изазивају поремећајни фактори и да на основу тога уче и минимализују грешку после сваке следеће итерације. Резултат рада представља робота који је у стању да зна где је и где иде само на основу ознака на подлози.

8. Литература

- [1] Миљковић З., Предавања на предмету Интелигентни технолошки системи (ПРО220-0131), Универзитет у Београду – Машински факултет, Београд 2009.
- [2] Бабић Б., Предавања на предмету Интелигентни технолошки системи (ПРО220-0131), Универзитет у Београду – Машински факултет, Београд 2009.
- [3] Петровић И., Предавања за предмету Мобилна роботика, Факултет електротехнике и рачунарства, Загреб 2009.
- [4] Петровић П., Предавања на предмету Мехатронски системи (ПРО), Универзитет у Београду – Машински факултет, Београд 2009.
- [5] Милутиновић Д., Предавања на предмету Индустијски роботи (ПРО), Универзитет у Београду – Машински факултет, Београд 2008.
- [6] http://en.wikipedia.org/wiki/Kalman_filter
- [7] <http://www.philohome.com/nxtmotor/nxtmotor.htm>
- [8] http://cache.lego.com/downloads/education/9797_LME_UserGuide_US_low.pdf
- [9] <http://loya.5u.com/strane/toc.html>