УНИВЕРЗИТЕТ У БЕОГРАДУ

МАШИНСКИ ФАКУЛТЕТ

АЛЕКСАНДАР В. ЈОКИЋ

ВИЗУЕЛНО УПРАВЉАЊЕ МОБИЛНОГ РОБОТА БАЗИРАНО НА БИОЛОШКИ ИНСПИРИСАНИМ ТЕХНИКАМА ВЕШТАЧКЕ ИНТЕЛИГЕНЦИЈЕ

Докторска дисертација

Београд, 2025. године

UNIVERSITY OF BELGRADE

FACULTY OF MECHANICAL ENGINEERING

ALEKSANDAR V. JOKIĆ

VISUAL CONTROL OF MOBILE ROBOT BASED ON BIOLOGICALLY INSPIRED ARTIFICIAL INTELLIGENCE TECHNIQUES

Doctoral Dissertation

Belgrade, 2025

Информације о ментору и члановима комисије за одбрану докторске дисертације

Ментор: др Милица Петровић, ванредни професор
Универзитет у Београду – Машински факултет

Чланови комисије: др Зоран Миљковић, редовни професор
Универзитет у Београду – Машински факултет

др Живана Јаковљевић, редовни професор
Универзитет у Београду – Машински факултет

др Радиша Јовановић, редовни професор
Универзитет у Београду – Машински факултет

др Никола Славковић, ванредни професор
Универзитет у Београду – Машински факултет

др Александар Родић, научни саветник
Институт Михајло Пупин, Београд

Датум одбране:

Изјаве захвалности

Желео бих да изразим најискренију захвалност својој менторки, проф. др Милици Петровић, на времену, труду и енергији уложеним у мој развој током година заједничког рада. Да није било њене иницијативе и несебичне подршке, ова докторска дисертација никада не би настала. Такође бих желео да се захвалим проф. др Зорану Миљковићу на многим животним саветима, стручним усмерењима и подршци у току мог усавршавања. Захвалност дугујем и проф. др Живани Јаковљевић која ми је омогућила стицање искуства и изван оквира Машинског факултета. Захваљујем се члановима комисије, проф. др Радиши Јовановићу, проф. др Николи Славковићу и научном саветнику др Александру Родићу на помоћи и сугестијама током израде докторске дисертације. Посебну захвалност дугујем својим блиским колегама из „240“ и са Катедре за производно машинство, који су допринели да овај мој дуги пут буде испуњен смехом, осећајем припадности и да резултира искреним пријатељствима.

Захваљујем се својим родитељима, Светлани и Владану, и сестри Ани за љубав и безусловну подршку које осећам током читавог живота.

На крају, желео бих да се захвалим и својој вереници Мартини за дане испуњене срећом, задовољством и љубављу.

ВИЗУЕЛНО УПРАВЉАЊЕ МОБИЛНОГ РОБОТА БАЗИРАНО НА БИОЛОШКИ ИНСПИРИСАНИМ ТЕХНИКАМА ВЕШТАЧКЕ ИНТЕЛИГЕНЦИЈЕ

Сажетак

Глобално тржиште XXI века постаје све захтевније у погледу рокова испоруке, као и варијантности производа, што изискује потребу за повишеним нивоом флексибилности и ефикасности технолошких система. Једна од технологија која може обезбедити потребну флексибилност је интеграција интелигентних мобилних робота као транспортних средстава јер омогућује ефикасне и оптимално терминиране токове материјала у технолошком систему. Како би то било могуће, мобилни роботи би требало да поседују сензорске системе за опажање динамичких промена у окружењу, управљачке системе базиране на сензорским подацима, као и алгоритме вештачке интелигенције који им омогућавају когнитивност, способност учења и доношења одлука о наредним акцијама у реалном времену. Како би искоришћење технолошког система било на високом нивоу, потребно је увести систем за интелигентно терминирање, како технолошких процеса, тако и роботизованих транспортних средстава.

Имајући претходно наведено у виду, у оквиру предметне докторске дисертације развијен је систем перцепције мобилног робота RAICO (енгл. *Robot with Artificial Intelligence based COgnition*) који на основу стерео визуелног система и семантичке сегментације генерисаних слика остварује високу тачност препознавања објеката у технолошком окружењу. Развијени су ефикасни модели конволуционих неуронских мрежа (енгл. *Convolutional Neural Networks*) који се могу имплементирати у реалном времену на рачунарској платформи *Nvidia Jetson Nano*. На генерисане семантичке мапе примењена су три оптимизациона алгоритма ради процеса регистрације слика. Након експерименталне верификације, еволуциони алгоритам (енгл. *OnePlusOne Evolutionary algorithm*) интегрисан са функцијом циља заједничких информација показује најбоље перформансе. Након извршеног процеса регистрације семантичких мапа, добијене матрице трансформација примењене су у стерео визуелном управљачком систему. Пет експерименталних евалуација изведених у 3D симулацији, као и са роботом RAICO, указују на то да се применом предложеног управљачког система остварује висок ниво тачности положаја мобилног робота, као и значајна робустност на поремећајне факторе попут заклоњености дела сцене, промене у осветљењу и грешке при семантичкој сегментацији. Такође, још једна важна предност предложеног система визуелног управљања огледа се у могућности коришћења циљних слика генерисаних у симулацији, чиме се значајно повећава флексибилност примене мобилних робота са визуелним управљањем.

На крају, развијена је методологија за терминирање технолошких процеса у оквиру које транспортне задатке изводи један мобилни робот. Предложен је математички модел и развијено је 13 функција циља. Комбинацијом различитих функција циља, генерисане су вишекритеријумске функције циља, а на основу две експерименталне верификације, метахеуристички алгоритам инспирисан интелигенцијом чопора вукова (енгл. *Grey wolf optimizer*) показао се као најбољи за решавање овог вишекритеријумског комбинаторно-оптимизационог проблема.

Кључне речи: интелигентни технолошки системи, визуелно управљање, мобилни роботски системи, дубоко машинско учење, конволуционе неуронске мреже, семантичка сегментација, биолошки инспирисани алгоритми оптимизације, терминирање мобилних роботских система, унутрашњи транспорт материјала, вештачка интелигенција.

Научна област: машинско инжењерство

Ужа научна област: производно машинство

УДК: 004.896:658.286.6:007.52(043.3)

VISUAL CONTROL OF MOBILE ROBOT BASED ON BIOLOGICALLY INSPIRED ARTIFICIAL INTELLIGENCE TECHNIQUES

Abstract

The global 21st century market is becoming increasingly demanding in terms of both delivery deadlines as well as product diversity, which necessitates a higher level of flexibility and efficiency of manufacturing systems. One of the technologies capable of providing the required flexibility is based on the integration of intelligent mobile robots within transport system, as they enable efficient and optimally scheduled material flows in a manufacturing system. To facilitate this, mobile robots should be equipped with sensor systems for detecting dynamic changes in the environment, control systems based on sensor data, and artificial intelligence algorithms that enable cognition, learning capability, and reasoning about future actions in real time scenarios. Finally, to maintain a high level of efficiency of the manufacturing system, it is necessary to introduce an intelligent system for simultaneous scheduling of process plans and material transport systems.

Bearing this in mind, a perception system for the mobile robot RAICO (Robot with Artificial Intelligence based COgnition) has been developed, based on a stereo vision system and semantic segmentation of the generated images, which achieves a high level of object recognition accuracy in manufacturing environment. Efficient Convolutional Neural Network models have been developed and implemented in real time on the Nvidia Jetson Nano computing platform. Three optimization algorithms were applied for the purpose of semantic maps registration. Following experimental evaluation, the OnePlusOne Evolutionary algorithm integrated with a mutual information-based objective function proved to have the highest performance. After the semantic map registration process was completed, the resulting transformation matrices were applied in the stereo vision-based control system. Five experimental evaluations conducted in 3D simulation, as well as with the RAICO robot, indicate that the proposed visual control system has a high level of position accuracy and significant robustness to disruptive factors such as occluded parts of the scene, lighting changes, and errors in semantic segmentation. Another essential advantage of the proposed vision-based control system lies in its ability to use target images generated in simulation, thereby significantly increasing the flexibility of mobile robots with vision-based control.

Finally, a methodology for process planning and scheduling was developed whereas transportation tasks are executed by a single mobile robot. A mathematical model was proposed, and 13 objective functions were developed. By combining different objective functions, two experimental verifications with multi-criteria optimization procedure were performed, and the Grey Wolf Optimizer algorithm proved to be the most effective for solving these combinatorial optimization problems.

Keywords: Intelligent manufacturing systems, visual servoing, mobile robots, deep machine learning, convolutional neural networks, semantic segmentation, biologically inspired optimization algorithms, mobile robot scheduling, material transport, artificial intelligence.

Scientific field: Mechanical engineering

Scientific subfield: Production engineering

UDC: 004.896:658.286.6:007.52(043.3)

Садржај

1. Увод 1

1.1. Интелигентни мобилни роботски системи као транспортна средства у технолошком систему – значај и примена 2

1.2. Циљ истраживања, полазне хипотезе и научни допринос 4

2. Интелигентни мобилни роботски систем RAICO 6

3. Систем перцепције мобилног робота 14

3.1. Дубоко машинско учење – основе 14

3.2. Ефикасни модели дубоког машинског учења 19

3.3. Семантичка сегментација 21

3.4. Развој система перцепције мобилног робота 24

3.4.1. Регистрација слика 29

3.4.2. Дефинисање функција циља 30

3.4.3. Оптимизациони алгоритми за регистрацију слика 32

3.4.3.1. Градијентни поступак 32

3.4.3.2. Алгоритам симулираног жарења 34

3.4.3.3. Еволуциони алгоритам 35

3.4.3.4. Упоредна анализа оптимизационих алгоритама за регистрацију слика 36

4. Визуелно управљање мобилних роботских система 38

4.1. Преглед стања у области визуелног управљања 39

4.1.1. Директно визуелно управљање 39

4.1.2. Визуелно управљање мобилних робота 44

4.1.3. Машинско учење у визуелном управљању робота 46

4.2. Визуелно управљање роботских система – математичко-алгоритамске основе 50

4.2.1. Поређење IBVS и PBVS алгоритама у симулацији 59

4.3. Стерео визуелно управљање нехолономних мобилних роботских система базирано на семантичкој сегментацији 66

4.3.1. Први експеримент – симулација 71

4.3.2. Други експеримент – симулација 74

4.3.3. Први експеримент у реалном окружењу 76

4.3.4. Други експеримент у реалном окружењу 80

4.3.5. Трећи експеримент у реалном окружењу 82

5. Терминирање мобилних робота у оквиру интелигентних технолошких система 85

5.1. Преглед стања у области терминирања роботских система 86

5.2. Кодирање јединки плана терминирања 88

5.3. Математички модел 92

5.4. Вишекритеријумска оптимизација планова терминирања 99

5.5. Алгоритам инспирисан интелигенцијом чопора вукова 101

5.6. Примена алгоритма инспирисаног интелигенцијом чопора вукова у вишекритеријумском терминирању једног мобилног робота 102
5.7. Експериментална верификација оптимизације планова терминирања 104
5.7.1. Први експеримент терминирања једног мобилног робота 105
5.7.2. Други експеримент терминирања једног мобилног робота 113
6. Закључак 120
7. Литература 123

Списак слика

Слика 1.1. Примери индустријских мобилних роботских система [4]. 2
Слика 2.1. Мобилни роботски систем RAICO. 6
Слика 2.2. Погонски систем мобилног робота RAICO. 7
Слика 2.3. Систем за манипулацију мобилног робота RAICO. 7
Слика 2.4. Стерео визуелни систем мобилног робота RAICO. 7
Слика 2.5. Хардверски део управљачког систем мобилног робота RAICO. 8
Слика 2.6. Лабораторијски модел технолошког окружења [2]. 9
Слика 2.7. Рачунарске платформе *Jetson Nano Developer Kit* (лево) и *Raspberry Pi 4* (десно)... 9
Слика 2.8. Мобилни роботски систем RAICO у лабораторијском моделу технолошког окружења и слике генерисане применом стерео визуелног система машинског гледања. 10
Слика 2.9. Слике генерисане са сочивима *Evetar M118B029528W* (лево) и *Evetar M118B0418W* (десно). 11
Слика 2.10. Мобилни роботски систем RAICO са новим стерео визуелним системом. 12
Слика 2.11. Слике генерисане са различитим параметрима камере. 13
Слика 3.1. Генерализована структура CNN мреже за задатак класификације слика. 16
Слика 3.2. Параметри конволуционог слоја. 17
Слика 3.3. ReLU активационе функције. 18
Слика 3.4. Разлика између конволуција. 20
Слика 3.5. Семантичка сегментација слике, приказана додавањем боје преко дела слике у коме су предмети који дефинишу различите класе објеката; на пример, жутом бојом је наглашен под, црвеном столови, зеленом столице, наранџастом зидови, светло црвеном завесе итд. 22
Слика 3.6. Генеричка архитектура CNN модела за семантичку сегментацију. 23
Слика 3.7. Четири градивна блока предложеног CNN модела. 25
Слика 3.8. Два градивна блока предложеног CNN модела. 26
Слика 3.9. Приказ архитектура RN_2222 и RN_1D_2330 предложеног CNN модела. 27
Слика 3.10. Приказ архитектура RN_sep_4400 и RN_sep_8000 предложеног CNN модела. 28
Слика 3.11. Пример слике (лево) и њене семантичке мапе (десно). 31
Слика 3.12. Вредности функција циља за различите параметре транслације слике. 32
Слика 3.13. Тачност поклапања пре и после регистрације. Слике а) и б) представљају поклапање семантичких мапа пре и после регистрације за ниво 2, респективно, док су в) и г) слике за ниво 1, а слике д) и ђ) су за ниво 0. 36
Слика 4.1. Подаци који се користе у оквиру различитих метода визуелног управљања. 38
Слика 4.2. Дефинисање а) координатних система мобилног робота и камера б) транслаторне брзине, угаоних брзина и параметара мобилног робота. 50
Слика 4.3. Модел инфинитезимално малог отвора бленде. 51
Слика 4.4. Стерео визуелни систем – основне карактеристике. 53
Слика 4.5. Разлике метода визуелног управљања. 53
Слика 4.6. Примена матрице хомографије. 57
Слика 4.7. Слика циља (**I***) у свим експериментима. 59
Слика 4.8. Илустрација управљачког алгоритма који се састоји из три фазе. 60

Слика 4.9. Симулационо окружење за поређење алгоритама IBVS и PBVS.....63
Слика 4.10. IBVS промена грешака по фазама.....63
Слика 4.11. PBVS промена грешака по фазама.....64
Слика 4.12. Промена положаја мобилног робота применом алгоритама IBVS (лево) и PBVS (десно).....64
Слика 4.13. Промена позиције карактеристичних објеката на слици применом алгоритама IBVS (лево) и PBVS (десно).....65
Слика 4.14. Трајекторије мобилног робота остварене применом алгоритама IBVS (лево) и PBVS (десно).....65
Слика 4.15. Фазе алгоритма стерео визуелног управљања на бази семантичке сегментације. 67
Слика 4.16. Померање координатног система камере у хоризонталној равни услед ротације робота.....68
Слика 4.17. Померање координатног система камере у вертикалној равни услед ротације робота.....69
Слика 4.18. Предложени алгоритам стерео визуелног управљања.....70
Слика 4.19. Мобилни роботски систем RAICO у симулационом окружењу.....71
Слика 4.20. Слике генерисане стерео паром у жељеном положају (прва врста) и почетном положају (друга врста).....72
Слика 4.21. Резултати првог експеримента у симулацији.....72
Слика 4.22. Путања коју је остварио мобилни робот за најбоље понављање првог експеримента у симулацији.....73
Слика 4.23. Промена оријентације мобилног робота током визуелног управљања.....73
Слика 4.24. Криве конвергенције грешака и угаоне брзине, током различитих фаза реализације алгоритма стерео визуелног управљања.....74
Слика 4.25. Грешка компоненти вектора стања мобилног робота током визуелног управљања. 75
Слика 4.26. Слике генерисане стерео визуелним системом током визуелног управљања. 75
Слика 4.27. Слике генерисане у крајњем (горе) и жељеном (доле) положају. 76
Слика 4.28. Слике стерео пара генерисане у жељеном положају (прва врста) и почетном положају (друга врста) за први експеримент у реалном окружењу.....77
Слика 4.29. Путања мобилног робота у најбољем понављању првог експеримента. 77
Слика 4.30. Слике апсолутне разлике интензитета пиксела између жељених и крајњих слика. 78
Слика 4.31. Криве конвергенције грешке и угаоне брзине за различите фазе алгоритма визуелног управљања.....78
Слика 4.32. Семантичке мапе слика генерисаних током првог експеримента у реалном окружењу.....79
Слика 4.33. Разлика тренутних и жељених семантичких мапа пре и после процеса регистрације.....80
Слика 4.34. Слике у жељеном и почетном положају за други експеримент у реалном окружењу.....81
Слика 4.35. Криве конвергенције грешака и угаоних брзина свих фаза стерео визуелног алгоритма генерисане током другог експеримента у реалном окружењу.....81
Слика 4.36. Семантичке мапе почетних слика у другом експерименту. 82

Слика 4.37. Слике апсолутне разлике интензитета пиксела између жељених и крајњих слика у другом експерименту. 82
Слика 4.38. Почетни положај мобилног роботског система RAICO у трећем експерименту. 82
Слика 4.39. Жељене слике у симулацији (прва врста) и слике генерисане у почетном положају (друга врста). 83
Слика 4.40. Криве конвергенције грешака и угаоних брзина свих фаза стерео визуелног алгоритма генерисане током трећег експеримента у реалном окружењу. 83
Слика 4.41. Жељене слике генерисане у симулацији и њихове семантичке мапе (прва врста) и слике у финалном положају и њихове семантичке мапе (друга врста). 84
Слика 4.42. Слике апсолутне разлике интензитета пиксела између жељених и крајњих слика у трећем експерименту. 84
Слика 5.1. Мрежа алтернативних технолошких процеса обраде репрезентативног дела 1 (ti представљају времена обраде) [2], [128]. 89
Слика 5.2. Мрежа алтернативних технолошких процеса обраде репрезентативног дела 2 (ti представљају времена обраде) [2], [128]. 90
Слика 5.3. Мрежа алтернативних технолошких процеса обраде репрезентативног дела 3 (ti представљају времена обраде) [2], [128]. 91
Слика 5.4. Кодирање плана терминирања. 91
Слика 5.5. Гантов дијаграм. 94
Слика 5.6. Гантов дијаграм са означеним параметрима везаним за мобилни робот, ВЧР – време чекања робота, ТД – транспорт дела, а ПР – позиционирање робота. 94
Слика 5.7. Пример Парето фронта са доминантним решењима [147]. 100
Слика 5.8. Стратегије за одабир лидера код GWO алгоритма. 102
Слика 5.9. Процес модификације решења током једне генерације GWO алгоритма. 103
Слика 5.10. Парето фронтови за све алгоритме CF1-6. 109
Слика 5.11. Парето фронтови за све алгоритме CF7-9. 110
Слика 5.12. Прва два Гантова дијаграма за одабрана оптимална решења. 111
Слика 5.13. Друга два Гантова дијаграма за одабрана оптимална решења. 112
Слика 5.14. Хистограм приказаних решења за други експеримент. 119

Списак табела

Табела 2.1. Карактеристике компоненти стерео визуелног система. 8
Табела 2.2. Техничке спецификације анализираних рачунарских платформи. 9
Табела 2.3. Техничка спецификација камере *Basler Dart daA1600-60uc*. 10
Табела 2.4. Техничка спецификација анализираних сочива. 11
Табела 3.1. Резултати поређења CNN модела. 29
Табела 3.2. Псеудокод градијентног поступка. 33
Табела 3.3. Псеудокод алгоритма симулираног жарења. 34
Табела 3.4. Псеудокод еволуционог алгоритма. 35
Табела 3.5. Положаји мобилног робота. 36
Табела 3.6. Резултати поређења оптимизационих алгоритама и функција циља; GDe – градијентни поступак, SA – симулирано жарење, EV – еволуциони алгоритам. .. 37
Табела 4.1. Методе директног визуелног управљања. 39
Табела 4.2. Алгоритам IBVS за управљање нехолономног мобилног робота. 61
Табела 4.3. Алгоритам PBVS за управљање нехолономног мобилног робота. 62
Табела 4.4. Резултати другог експеримента у симулацији. 74
Табела 4.5. Грешке положаја при првом експерименту. 77
Табела 4.6. Поређење предложеног алгоритма са методама из литературе. 80
Табела 5.1. Псеудокод GWO алгоритма за вишекритеријумско терминирање. 103
Табела 5.2. Комбинације функција циља за вишекритеријумску оптимизацију. 105
Табела 5.3. Резултати првог експеримента за GD метрику. 106
Табела 5.4. Резултати првог експеримента за IGD метрику. 106
Табела 5.5. Резултати првог експеримента за SP метрику. 107
Табела 5.6. Резултати првог експеримента за MS метрику. 108
Табела 5.7. Поставка другог експеримента 113
Табела 5.8. Резултати другог експеримента за CF #6 – IGD метрика. 114
Табела 5.9. Резултати другог експеримента за CF #6 – MS метрика. 114
Табела 5.10. Резултати другог експеримента за CF #7 – GD метрика. 115
Табела 5.11. Резултати другог експеримента за CF #7 – SP метрика. 116
Табела 5.12. Резултати другог експеримента за CF #8 – IGD метрика 116
Табела 5.13. Резултати другог експеримента за CF #8 – MS метрика. 117
Табела 5.14. Резултати другог експеримента за CF #9 – GD метрика. 118
Табела 5.15. Резултати другог експеримента за CF #9 – SP метрика. 118

Списак скраћеница

CPS	Кибернетско физички системи (енгл. *Cyber Physical Systems*)
2D	Дводимензионални
3D	Тродимензионални
DL	Дубоко машинско учење (енгл. *Deep Learning*)
AI	Вештачка интелигенција (енгл. *Artificial Intelligence*)
AGV	Аутоматски вођена колица (енгл. *Automatic Guided Vehicles*)
ML	Машинско учење (енгл. *Machine Learning*)
RAICO	енгл. *Robot with Artificial Intelligence based COgnition*
CNN	Конволуционе неуронске мреже (енгл. *Convolutional Neural Networks*)
ReLU	Позитивна линеарна активациона функција (енгл. *Rectified Linear Unit*)
FLOPs	Број компјутерских операција у покретном зарезу (енгл. *FLoating Point Operations)*
FPS	Фрејмови у секунди (енгл. *Frames Per Second*)
mIoU	Средњи количник пресека и уније (енгл. *mean Intersection over Union*)
BB	Основни блок
BRB	Умањујући блок
1DB	Једнодимензионални (1D) блок
1DRB	Једнодимензионални (1D) умањујући блок
SPB	Блок MobileNet 1D конволуције
SRB	Умањујући блок MobileNet 1D конволуције
AC	Тачност (енгл. *Accuracy*)
MI	Заједничке информације (енгл. *Mutual Information*)
GDe	Градијентни поступак (енгл. *Gradient Descent*)
SA	Симулирано жарење (енгл. *Simulated Annealing*)
EV	Еволуциони алгоритам (енгл. *OnePlusOne EVolutionary algorithm*)
IBVS	Управљање на бази грешака у параметрима слике (енгл. *Image-Based Visual Servoing*)
PBVS	Управљање на бази издвојених карактеристика са слике неопходних за естимацију положаја изабраног објекта (енгл. *Position-Based Visual Servoing*)
HVS	Хибридни системи управљања (енгл. *Hybrid Visual Servoing)*
DVS	Системи директног визуелног управљања (енгл. *Direct Visual Servoing*)
GMM	Гаусови мешовити модели (енгл. *Gaussian Mixture Models*)
HOG	Хистограм оријентисаних градијената (енгл. *Histogram of Oriented Gradients*)
IMU	Инерциона мерна јединица (енгл. *Inertial Measurement Unit*)
GWO	Алгоритам инспирисан интелигенцијом чопора вукова (енгл. *Grey Wolf Optimizer*)
GA	Генетички алгоритам (енгл. *Genetic Algorithm)*
PSO	Алгоритам инспирисан интелигенцијом роја честица (енгл. *Particle Swarm Optimization*)
GD	Дистанца (енгл. *Generational Distance*)
IGD	Инверзна дистанца (енгл. *Inverted Generational Distance*)
SP	Одстојање (енгл. *Spacing*)
MS	Максимално растојање (енгл. *Maximum Spread*)
AOA	Аритметички оптимизациони алгоритам (енгл. *Arithmetic Optimization Algorithm*)

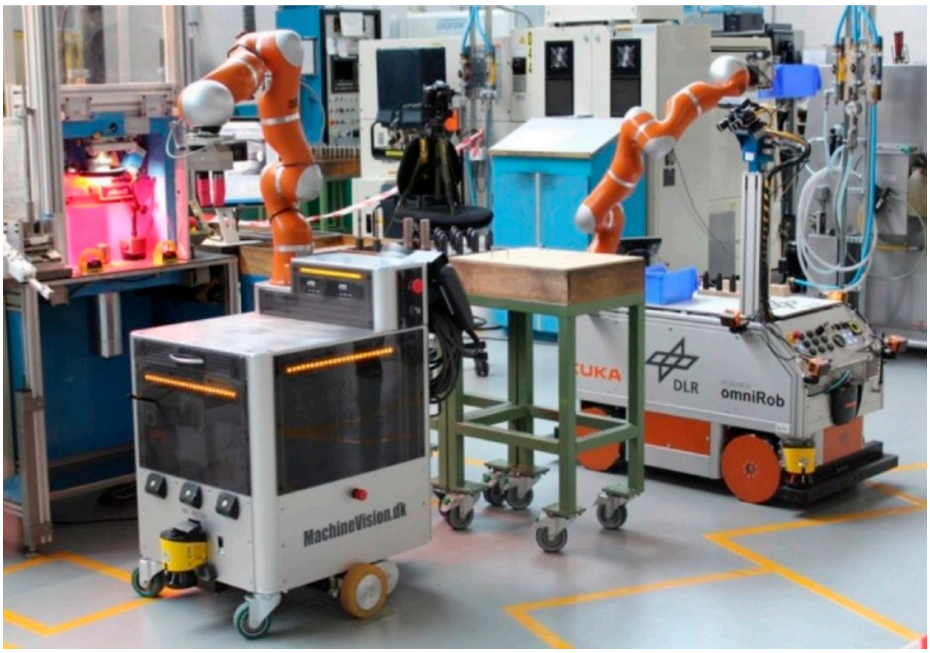
1. Увод

Ради остваривања конкурентности на изузетно компетитивном светском тржишту, почетком треће деценије 21. века све већи број производно оријентисаних компанија инкорпорира парадигму Индустрије 4.0 као основ за развој иновативних производа, подизање нивоа флексибилности и ефикасности технолошких система. У основи Индустрије 4.0 су кибернетско физички системи (енгл. *Cyber Physical Systems* – CPS) који представљају системе у оквиру којих су интегрисани физички и рачунарски процеси са израженом могућношћу међусобне интеракције у реалном времену. Примери CPS-а су паметни сензори и актуатори, или мобилни роботи. Основне карактеристике кибернетско физичких производних система које је потребно остварити у оквиру Индустрије 4.0 су: компатибилност CPS-а, креирање дигиталних близанаца, децентрализација CPS-а, могућност аквизиције и анализе података из технолошког окружења у реалном времену и модуларност CPS-а [1]. Кибернетско физички системи анализирани у оквиру предметне докторске дисертације, који се могу применити ради остваривања постулата парадигме Индустрије 4.0 су мобилни роботски системи, примењени у оквиру задатака унутрашњег транспорта материјала у интелигентом технолошком окружењу. Њиховом применом ниво флексибилности, а посебно физичке реконфигурабилности и ефикасности транспортног система значајно се повећава. Један од најутицајнијих праваца развоја мобилних роботских система односи се на интеграцију са техникама вештачке интелигенције (енгл. *Artificial Intelligence* – AI). Овом интеграцијом може се подићи ниво аутономности мобилних роботских система, тј. могу се остварити различите когнитивне способности. Напредне технике вештачке интелигенције, а поготово модели дубоког машинског учења (енгл. *Deep Learning* – DL), омогућавају роботским системима да тачно врше перцепцију свог окружења, на основу чега могу да доносе одлуке и врше управљање у реалном времену, а све то како би реализовали постављени технолошки задатак. У том смислу, визуелни системи базирани на камерама као „сензорима будућности“ обезбеђују значајан скуп репрезентативних података о окружењу на основу којих се, применом техника дубоког учења, може поуздано интерпретирати непосредно роботско окружење, а на основу тога вршити процеси одлучивања и визуелног управљања. Како би мобилни роботски систем могао да изврши транспортни задатак, потребни су му јасно дефинисани параметри везани за сам ток технолошког процеса. Оптимални редослед технолошких операција, алокација технолошких ентитета (ресурса) за те операције, као и оптимални распоред транспортних задатака, остварује се интегрисаном вишекритеријумском оптимизацијом технолошких процеса и планова терминирања са додатним ограничењима везаним за транспортни систем. У оквиру поменутог оптимизационог проблема, примењени су биолошки инспирисани оптимизациони алгоритми, који такође представљају једну од техника вештачке интелигенције у оквиру предметне докторске дисертације. Вишекритеријумска оптимизација омогућава истовремену минимизацију више супротстављених параметара перформанси технолошких система, као што су време производње и минимални транспортни токови материјала. Остваривањем оптималних планова терминирања, на основу више критеријума оптимизације, у значајној мери повећава се продуктивност самог интелигентног технолошког система, а истовремено се смањују временски губици и повећава искоришћење ресурса.

Интеграцијом техника вештачке интелигенције, као што су биолошки инспирисани алгоритми оптимизације и вештачке неуронске мреже, унапређују се традиционални технолошки системи, на основу чега се може остварити највиша класа флексибилних технолошких система односно интелигенти технолошки системи. Циљ примене интелигентних технолошких система је перманентно повећање успешности понашања целокупног технолошког система, чак и при значајној присутности стохастичких неодређености у самом систему [2].

1.1. Интелигентни мобилни роботски системи као транспортна средства у технолошком систему – значај и примена

Један од фундаменталних захтева за остваривање технолошког система који је у стању да оствари високе захтеве парадигме Индустрије 4.0 је робустан, реконфигурабилан и ефикасан унутрашњи транспорт. Унутрашњи транспорт материјала у оквиру технолошких система подразумева транспорт, складиштење и заштиту полуфабриката, готових делова, подсклопова и склопова у току процеса производње, чија се реализација остварује применом транспортних средстава [3]. Традиционални видови транспортних средстава попут ручних виљушкара, конвејера или аутоматски вођених колица (енгл. *Automatic guided vehicles* – AGV), немају потребан ниво флексибилност и/или ефикасности за остваривање постављеног задатка. Њихови недостаци представљени су ниским нивоом реконфигурабилности самих транспортних токова материјала или директно ниском могућношћу аутоматизације, чиме ефикасност знатно опада. Међутим, интелигенти роботски системи обезбеђују довољну флексибилност и реконфигурабилност, јер тада транспортни токови не зависе од инфраструктуре самог технолошког система, а изменом хватача могуће је извршити манипулацију и транспорт различитих машинских делова. Са друге стране, ефикасност мобилних робота у транспорту полуфабриката, подсклопова, склопова и готових делова је на високом нивоу, јер се може обезбедити оптимална путања за целокупни ток материјала, као и за позиционирање и постављање делова на машине алатке. Додатни показатељ потенцијала мобилних индустријских робота огледа се и кроз тренд светских произвођача индустријских робота (нпр. *Kuka*, *Fanuc*, *ABB*, *Yaskawa*) да све више улажу у развој индустријских мобилних робота, а неки већ имају значајан портфолио комерцијалних решења[1,2]. Примери изгледа индустријских мобилних роботских платформи представљени су на слици 1.1. Поред основних функција транспорта материјала и опслуживања машина алатки у производном окружењу, индустријски мобилни роботи имају могућност повећања своје продуктивности извршавањем додатних операција, као што су спајање, допуна система за аутоматску монтажу, инспекција делова и палетизација или паковање делова.



Слика 1.1. Примери индустријских мобилних роботских система [4].

[1] https://www.yaskawa.eu.com/application/type/application/mobile-robots_a11132, датум последњег приступа 10.03.2025.
[2] https://www.kuka.com/en-de/products/amr-autonomous-mobile-robotics, датум последњег приступа 10.03.2025.

Основна предност индустријских мобилних роботских система, у односу на традиционалне видове транспорта материјала, остварена је на основу интерпретације података прикупљених са сензора ради аутономног понашања и одлучивања о акцијама потребним за оптимално извршавања постављеног технолошког задатка. С обзиром на то, многи аутори су анализирали различите сензорске системе [5] који би омогућили мобилним роботима да остваре овакав вид понашања. Сензор који има највећи потенцијал и који је добио атрибут „сензор будућности“ је камера. У односу на остале сензоре за опажање попут LIDAR-а (енгл. *Laser Imaging, Detection And Ranging*), радара или ултразвучних сензора, камере омогућавају роботима да прикупљају велике количине података, а уједно су приступачне и веома распрострањене. Напредни LIDAR сензори су за један до два реда величине скупљи, а при томе обезбеђују искључиво геометријске информације о роботском окружењу. Супротно, камера обезбеђује и податке везане за боју, осветљеност и текстуру објеката у тренутној сцени, што омогућава остваривање вишег нивоа тачности алгоритама за перцепцију окружења. Уколико је на основу система перцепције препознато окружење мобилног робота, могу се применити управљачки алгоритми ради извођења одређених акција и извршавања постављеног задатка. Примарни недостатак камере је у перспективној трансформацији 3D података у 2D простор слике, чиме се губе информације о удаљености релевантних објеката у односу на координатни систем камере. Међутим, по угледу на људски визуелни систем, могуће је остварити стерео визуелни систем додавањем још једне паралелно постављене камере на одговарајућем осном растојању. Применом стерео визуелног система, могуће је извршити процес триангулације ради одређивања комплетне 3D позиције (укључујући и удаљеност) карактеристичних објеката у координатном систему стерео камере, чиме се поменути недостатак негира на рачун потребе за додатним процесорским ресурсима. На основу истакнутих предности стерео визуелног система, произилази и његова популарност при пројектовању напредних мобилних роботских система[3]. Како би се информације добијене од стерео визуелног система примениле за кретање мобилног роботског система, односно остваривање жељеног положаја (позиције и оријентације), могу се применити алгоритми визуелног управљања (енгл. *Visual servoing*). Визуелно управљање представља скуп алгоритама код којих се управљање динамичких система (односно роботских система разматраних у оквиру предметне докторске дисертације) врши искључиво на основу информација добијених од камера. Такође, човек преко 80% података из свог непосредног окружења добија путем свог визуелног система, а те визуелне информација посредују у одговарајућој мери у минимално 80% људских активности везаних за перцепцију, когнитивност и учење [6]. С обзиром на то, може се закључити да човеково чуло вида директно и најинтензивније (у односу на друга чула) утиче на процесе учења, што додатно поспешује интеграцију роботског визуелног система са алгоритмима машинског учења, тј. вештачке интелигенције.

Области вештачке интелигенције и роботике, а поготово мобилне роботике, у модерном инжењерству доведене су у симбиозу. Развој напредних мобилних роботских система обезбеђује истраживачима у области вештачке интелигенције платформу за трансфер развијених алгоритама из симулације у реална окружења, док развој алгоритама вештачке интелигенције унапређује могућности роботских система кроз остваривање интелигенције и аутономности. Различити проблеми у области роботике, као што су перцепција која укључује разумевање окружења (енгл. *Scene understanding*), аутоматско планирање редоследа акција (енгл. *Automated planning*) и планирање трајекторије (енгл. *Trajectory planning*) могу се у великој мери решити применом техника вештачке интелигенције.

Како постоје различите технике вештачке интелигенције, у наставку ће бити представљене све релевантне за предметну докторску дисертацију. Машинско учење (енгл. *Machine Learning* – ML) представља скуп математичко-алгоритамских метода за обучавање параметарског модела система, применом одређеног алгоритма учења, а сходно прикупљеним

[3] https://pal-robotics.com/robot/

репрезентативним подацима и критеријуму перформанси [7]. За разлику од ML код кога су улазни вектори реда величине до неколико стотина елемената, напредним алгоритмима дубоког машинског учења (енгл. *Deep learning*) на улазу могу се представити тензори од неколико милиона елемената (нпр. слике резолуције 1920×1080×3). Предност дубоког учења у односу на технике машинског учења, огледа се у ефикасности процеса учења при обучавању на изузетно великим скуповима података. Баш ова предност омогућава високу тачност коју алгоритми дубоког учења остварују, а притом је могуће оптимизовати структуру модела ради примене у реалном времену.

Поред дубоког учења, у фокусу предметне докторске дисертације су и метахеуристички биолошки инспирисани алгоритми оптимизације (енгл. *Biologically inspired optimization algorithms*), примењени у оквиру различитих задатака попут регистрације слика и терминирања мобилних робота. Предност метахеуристичких метода оптимизације огледа се у могућности претраге изузетно великог простора могућих решења, за шта би традиционалним алгоритмима оптимизације било потребно знатно више времена. Такође, велики број локалних минимума у оквиру функције циља би онемогућио примену традиционалних алгоритама, што за метахеуристичке алгоритме не представља проблем. На крају, иако метахеуристички алгоритми не гарантују остваривање оптималног решења *NP-hard* недетерминистичких полиномно оптимизационих проблема (какав је терминирање једног мобилног робота), у инжењерству где је претрага комплетног простора решења немогућа, овакви алгоритми представљају изузетно корисну алтернативу.

1.2. Циљ истраживања, полазне хипотезе и научни допринос

Основни циљ истраживања спроведених у оквиру предметне докторске дисертације подразумева развој методологије за примену техника машинског учења и метахеуристичке оптимизације у визуелном управљању мобилног робота у интелигентном технолошком систему. Поред основног циља, додатна научна истраживања спроведена су у домену интегрисаног терминирања једног мобилног робота и технолошких ентитета интелигентног технолошког система. Полазне хипотезе, које уједно представљају и декупловане циљеве истраживања, су:

- Систем перцепције, који се заснива на дубоком машинском учењу и стерео визуелном машинском гледању, обезбеђује могућност мобилном роботском систему да са високом тачношћу детектује и препознаје производно-технолошке ентитете.
- Стерео визуелно управљање базирано на семантичкој сегментацији могуће је применити за остваривање жељеног положаја (позиције и оријентације) нехолономног мобилног робота са аспекта постизања захтеване тачности задатака транспорта и манипулације.

Теоријска и експериментална верификација постављених циљева истраживања примарно је приказана кроз истраживања публикована у оквиру научних рада у часописима изузетних вредности, као и у два поглавља објављена у монографијама светски признатих реномираних издавача, чиме су постављене хипотезе у потпуности доказане.

Докторска дисертација организована је у шест поглавља у оквиру којих су представљени (i) увод у основе предметних области, (ii) преглед стања у области истраживања, као и (iii) доприноси остварени током истраживања на самој докторској дисертацији. Уводно поглавље посвећено је приказу области Индустрије 4.0, са фокусом на примену интелигентних мобилних робота, као транспортних средстава у оквиру интелигентних технолошких система. Такође су приказане полазне хипотезе, као и циљеви истраживања извршених у оквиру предметне докторске дисертације.

У оквиру другог поглавља дат је приказ процеса развоја мобилног роботског система RAICO (*Robot with Artificial Intelligence based COgnition*) са освртом на развој хардверско-софтверске структуре, као и стерео визуелног система. Све хипотезе и развијени алгоритми доказани су експерименталном верификацијом изведеном помоћу мобилног роботског система RAICO.

Треће поглавље посвећено је развоју система перцепције мобилног робота RAICO. Приказане су основе везане за методе машинског учења са фокусом на дубоко машинско учење, тј. на конволуционе неуронске мреже (енгл. *Convolutional Neural Networks* – CNN). Анализирани су основни слојеви који сачињавају архитектуру CNN модела. Затим, представљени су посебно развијени слојеви који чине процес конволуције значајно ефикаснијим, ради могућности примене у реалном времену, што је од изузетног значаја за њихову имплементацију у оквиру мобилних роботских система. Након тога, приказан је проблем семантичке сегментације сцене, као и приступ у решавању овог проблема применом CNN модела. На крају, представљен је развој система перцепције мобилног робота на бази ефикасних CNN модела, тренираних да изврше семантичку сегментацију лабораторијског модела технолошког окружења. Такође, приказана је процедура регистрације семантичких мапа применом три алгоритма оптимизације, као и три различите функције циља развијене у сврху одређивања тачности регистрације. На основу експерименталне верификације, приказан је оптимални CNN модел.

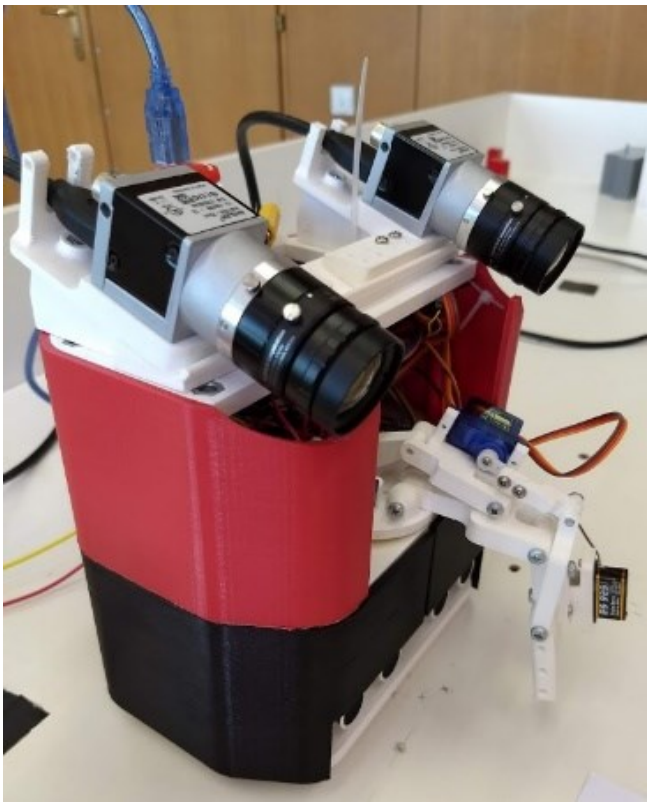
Четврто поглавље садржи развој стерео визуелног система управљања на бази семантичке сегментације. Прво је представљен преглед стања у области визуелног управљања роботских система, класификован у три целине. Прва је подразумевала методе директног визуелног управљања. У оквиру друге, приказана је примена алгоритама визуелног управљања код мобилних роботских система, док је трећа целина посвећена анализи примене метода машинског учења у оквиру система визуелног управљања. Затим, дате су математичко-алгоритамске основе везане за визуелно управљање роботских система, класификација метода визуелног управљања, а посебан фокус је на примени визуелног управљања код нехолономних мобилних робота чије се кретање извршава помоћу диференцијалног погона. На крају, приказан је развој система за стерео визуелно управљање мобилног робота RAICO. Основу развијеног система представља систем за перцепцију на основу кога се генерише семантичка мапа, а затим изврши њена регистрација са семантичким мапама слика циља. Извршено је пет експерименталних верификација, од којих су прве две извршене у 3D симулацији, а наредне три у реалном окружењу. Предност предложеног система, поред остварене тачности, приказује се кроз робустност на промену осветљења, нетачност семантичке сегментације, заклоњеност дела сцене, као и могућност примене на сликама које су генерисане у симулацији чиме се флексибилност алгоритма знатно повећава.

Пето поглавље односи се на развој алгоритма за терминирање технолошког система где се транспортни задаци извршавају применом једног мобилног робота. Приказан је преглед стања у оквиру терминирања роботских система, а затим је извршено дефинисање математичког модела, стратегије кодирања и 13 функција циља. Након тога, приказан је нови метахеуристички алгоритам инспирисан интелигенцијом чопора вукова (енгл. *Grey Wolf Optimizer*) прилагођен за примену у оквиру вишекритеријумске оптимизације планова терминирања. На основу две експерименталне верификације, доказано је да предложени алгоритам има боље перформансе процеса оптимизације у односу на поређене алгоритме.

Шесто поглавље посвећено је дискусији о реализованим истраживањима и развијеним алгоритмима, као и рекапитулацији најважнијих појединости везаних за допринос предметне докторске дисертације. На крају, дати су правци даљих истраживања.

2. Интелигентни мобилни роботски систем RAICO

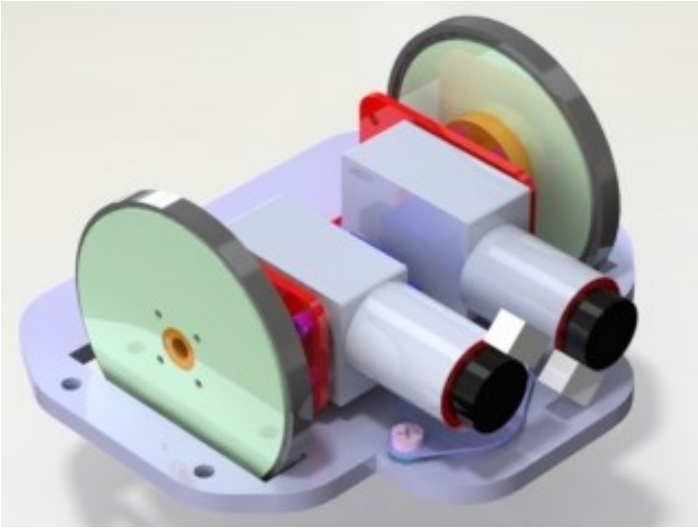
Мобилни робот RAICO (енгл. *Robot with Artificial Intelligence based COgnition*), од милоште назван Рајко, представља основни систем коришћен ради евалуације развијених алгоритама предметне докторске дисертације. Прва конфигурација мобилног роботског система RAICO (слика 2.1) настала је 2019. године као резултат истраживања у оквиру Лабораторије за индустријску роботику и вештачку интелигенцију (*Robotics & AI*).



Слика 2.1. Мобилни роботски систем RAICO.

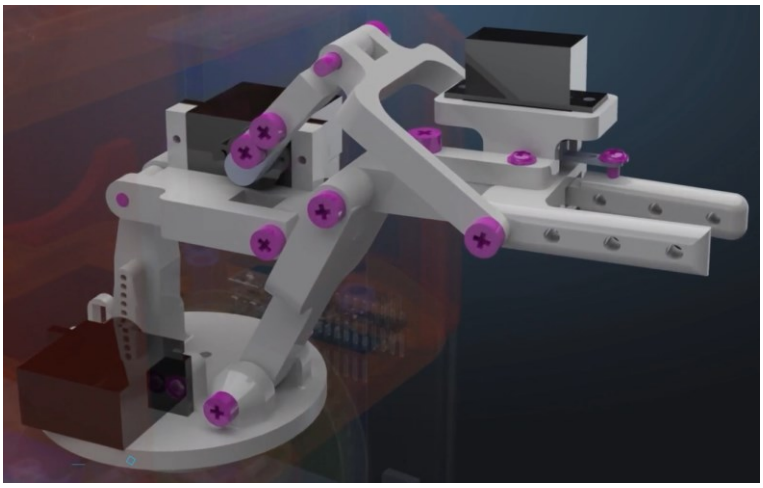
RAICO представља лабораторијски прототип мобилног роботског система. Примарна мотивација за развој једног оваквог роботског система је спровођење истраживања везаних за интелигентно стерео визуелно управљање. Поред интелигентног визуелног управљања, током година спроведени су различити експерименти везани за роботску колаборацију [8], интелигентно препознавање објеката у лабораторијском моделу технолошког окружења [9], интелигенту оцену успешности извршавања постављеног задатка манипулације [10] и терминирање мобилних роботских система [11]. Мобилни роботски систем састоји се из три паралелне платформе. На првој платформи постављен је диференцијални погонски систем, на другој платформи је манипулатор са три степена слободе, док се на трећој платформи налази стерео визуелни сензорски систем.

Диференцијални погон подразумева да мобилни робот поседује два независна погонска точка, као и трећи пасивни точак који служи за остваривање стабилности при кретању. Погонски систем мобилног робота (слика 2.2) остварен је на основу два серво мотора једносмерне струје са пужним редуктором који има преносни однос 40. Серво управљање обезбеђено је на основу инкрементaлних магнетних енкодера са резолуцијом од 2048 инкремената по обрту. Већина механичких делова на погонском систему израђена је адитивним технологијама, осим точкова који су добијени на обрадном систему струга, док је споља додата гумена превлака.



Слика 2.2. Погонски систем мобилног робота RAICO.

Систем за манипулацију (слика 2.3) изведен је у вертикалној зглобној конфигурацији са три ротационе осе. Свака оса ојачана је једним серво мотором, док је због стабилности између прве и друге осе постављен механизам зглобног четвороугла. На другој платформи, поред манипулатора, налазе се и различите електронске компоненте (које ће бити објашњене у наставку овог поглавља), батерија, као и управљачка јединица на бази *Arduino Mega* микроконтролера.



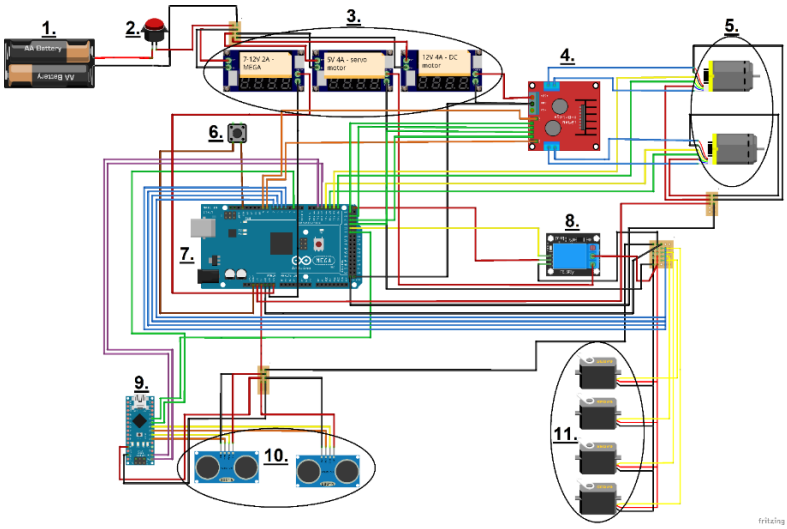
Слика 2.3. Систем за манипулацију мобилног робота RAICO.

Стерео визуелни систем (слика 2.4) састоји се од две паралелно постављене индустријске *Basler acA1920-25uc* камере и сочива *Fujinon DF6HA-1B* и налази се на највишој платформи робота, оријентисан под углом инклинације од 20°. Камере су постављене на осном растојању од 130 mm, а техничке спецификације камера и сочива дате су у табели 2.1.

Слика 2.4. Стерео визуелни систем мобилног робота RAICO.

Табела 2.1. Карактеристике компоненти стерео визуелног система.			
Камера *Basler acA1920-25uc*		Сочиво *Fujinon DF6HA–1B*	
Произвођач сензора	*ON Semiconductor*	Примена	Камере у боји
Тип сензора	MT9P031 – CMOS	Фокална дужина	6 mm
Димензије сензора	4,2 × 2,4 mm	Величина отвора	F1,2 - F16
Максимална резолуција	1920×1080 px	Хоризонтални угао видног поља	57,3°
Димензије пиксела	2,2 × 2,2 μm	Вертикални угао видног поља	43,8°
Максимална учестаност узроковања слика	25 FPS	Димензије	29,5 × 36,7 mm
Врста комуникације	USB 3.0	Прирубница сочива	C – mount
Ширина, дужина и висина	29,3 × 29 × 29 mm	Маса сочива	55 g
Прирубница за сочиво	C – mount		
Маса камере	80 g		

Првобитна конфигурација хардверског дела управљачког система мобилног робота RAICO дата је на слици 2.5, у оквиру које се може видети електрична шема и начин повезивања свих компоненти. Као што се може уочити, *Arduino Mega* управља свим моторима, док је *Arduino Nano* намењен за управљање ултразвучним сензорима који су постављени на прву платформу. Како серво мотори за манипулатор, серво мотори једносмерне струје за погонски систем и *Arduino Mega* раде са различитим напонима, било је потребно увести три регулатора напона који улазни напон подешавају на потребан. Додатно је постављен електрични прекидач чијом се активацијом омогућава укључивање манипулатора. Комуникација између *Arduino* микроконтролера остварена је коришћењем UART протокола.

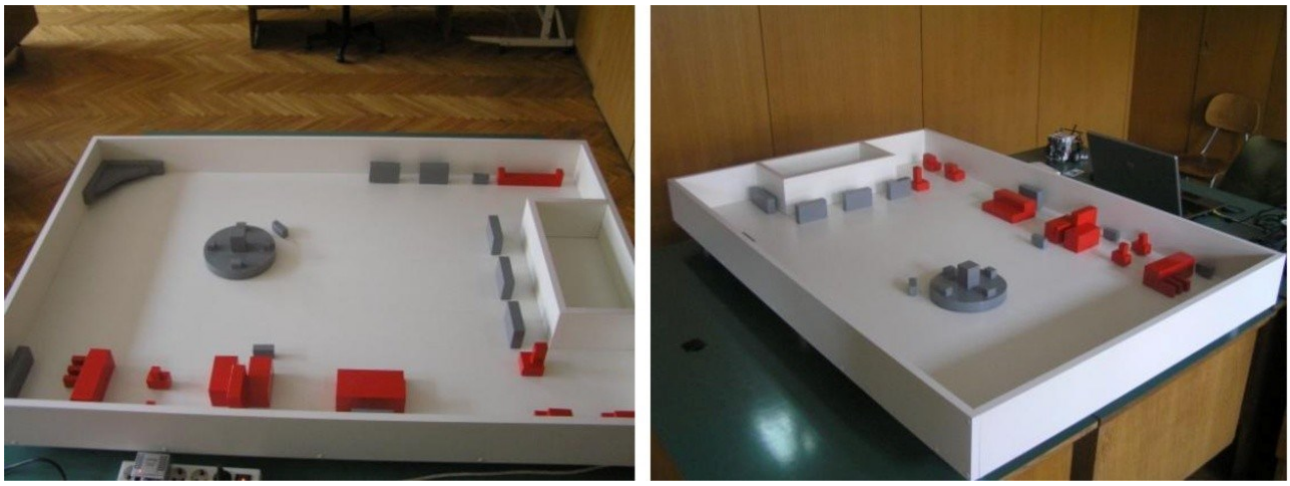


1. Напајање
2. Све стоп дугме
3. Регулатори напона
4. *H* мост
5. Серво мотори једносмерне струје
6. Ресет тастер
7. Микроконтролер *Arduino Mega*
8. Релеј
9. Микроконтролер *Arduino Nano*
10. Ултразвучни сензори
11. Серво мотори за манипулатор

Слика 2.5. Хардверски део управљачког систем мобилног робота RAICO.

Како нису постојале компјутерске платформе одговарајућих димензија и процесорских могућности, а микроконтролер *Arduino Mega* по својој конфигурацији није предвиђен за извршавање изузетно процесорски захтевних задатака, какви су обрада слике и дубоко машинско учење, камере су директно повезане на десктоп рачунар у оквиру кога је вршено процесирање. Из тог разлога, *Arduino Mega* је задужен за управљање хардверским компонентама, а процесорски захтевни алгоритми су имплементирани на рачунару у MATLAB софтверском окружењу. Након генерисања излаза алгоритма управљања, преко USB кабла и UART комуникације, информације су биле прослеђене на *Arduino Mega*. Камере су са рачунаром биле повезане коришћењем USB3.0 протокола. Лабораторијски модел технолошког

окружења у оквиру кога су тестирани сви алгоритми имплементирани на мобилни роботски систем RAICO приказан је на слици 2.6.

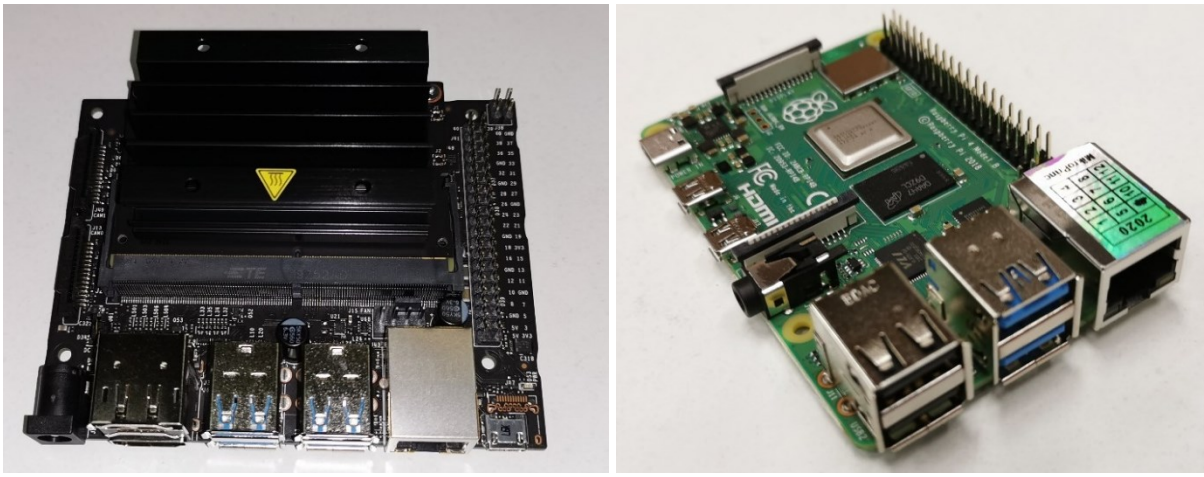


Слика 2.6. Лабораторијски модел технолошког окружења [2].

Након интензивног тестирања сваког подсистема мобилног робота RAICO, изведени су следећи закључци. Први недостатак иницијалне конфигурације робота односио се на крутост каблова којима су спојене камере и десктоп рачунар. Наиме, услед крутости каблова који су спојени на врху роботског система долазило је неправилног кретања мобилног робота, иако је било могуће позиционирањем каблова минимизирати њихов утицај. Поменути недостатак је касније отклоњен додавањем рачунарске платформе на мобилни робот. Разматране су рачунарске платформе *Jetson Nano Developer Kit* (слика 2.7) и *Raspberry Pi 4* (слика 2.7) [12], чије су техничке спецификације дате у табели 2.2.

Табела 2.2. Техничке спецификације анализираних рачунарских платформи.

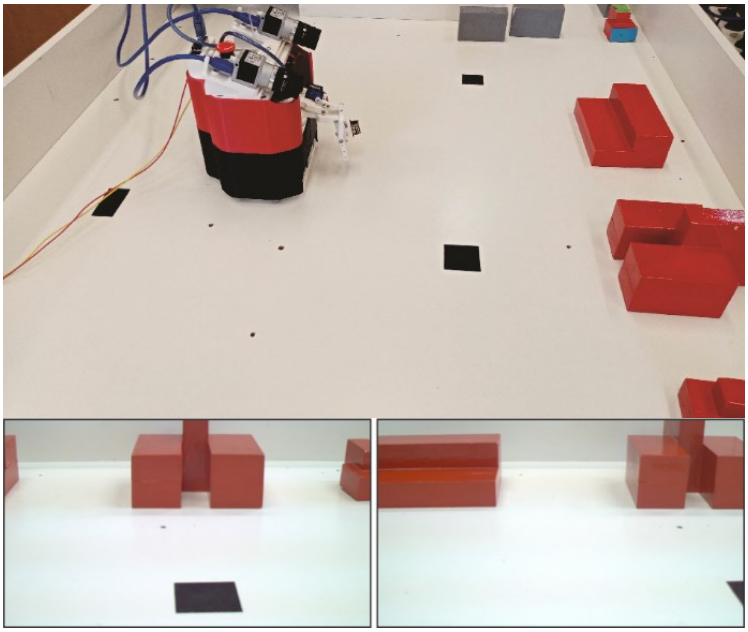
	Jetson Nano Developer Kit	*Raspberry Pi 4*
Графичка картица	NVIDIA 128-core Maxwell	Broadcom VideoCore VI
Процесор	Quad-core ARM A57 @ 1.43 GHz	Broadcom BCM2711B0 quad-core A72 (ARMv8-A) 64-bit @ 1.5 GHz
RAM меморија	4GB 64-bit LPDDR4 25.6 GB/s	8GB LPDDR4-3200 SDRAM
Интерфејс за камеру	2 × MIPI CSI-2 DPHY lanes	MIPI CSI-2 lanes
USB	4 × USB 3.0	2 × USB 3.0 ports; 2 × USB 2.0 ports
Димензије	≈ 80 × 100 mm	56 × 85 mm
Хлађење	Интегрисан хладњак	Нема
WiFi модул	Нема	2.4 GHz, 5.0 GHz IEEE 802.11ac wireless



Слика 2.7. Рачунарске платформе *Jetson Nano Developer Kit* (лево) и *Raspberry Pi 4* (десно).

Прва евидентна разлика између анализираних рачунарских платформи, која се може увидети на основу информација датих у табели 2.2, је да *Raspberry Pi 4* има процесор бољих перформанси, док са друге стране, *Jetson Nano* има бољу графичку картицу. Друга чињеница, која је знатно утицала на одабир рачунарске платформе, је да *Jetson Nano* по својој структури има могућност интеграције 2 камере (поседује 2 интерфејса искључиво за камере), што је неопходно за стерео визуелни систем. Међутим, како обе платформе поседују два USB3.0 порта, могуће је повезати две *Basler daA1600-60uc* камере на обе рачунарске платформе. На основу детаљне анализе, остварене након свеобухватног експерименталног рада, коначно селектовање *Jetson Nano* рачунарске платформе извршено је због могућности оптимизације *Nvidia* графичке картице за примену дубоког машинског учења у реалном времену, с обзиром да поседује *CUDA* језгра.

Други недостатак стерео визуелног система био је везан за остварени опсег видног поља визуелног система. За одабране камере и сочива, остварен је изузетно мали опсег видног поља стерео система од 38,8×22,4°, због чега су апликације за које је стерео визуелни систем могао бити коришћен биле ограничене. Пример положаја мобилног робота и слика генерисаних у том положају дати су на слици 2.8.



Слика 2.8. Мобилни роботски систем RAICO у лабораторијском моделу технолошког окружења и слике генерисане применом стерео визуелног система машинског гледања.

Затим је настављен развој стерео визуелног система у оквиру кога су интегрисане нове индустријске камере *Basler Dart daA1600-60uc* (табела 2.3) које имају знатно веће димензије сензора и на које се могу поставити сочива са мањом фокалном дужином, чиме је било могуће остварити знатно већи опсег видног поља стерео система.

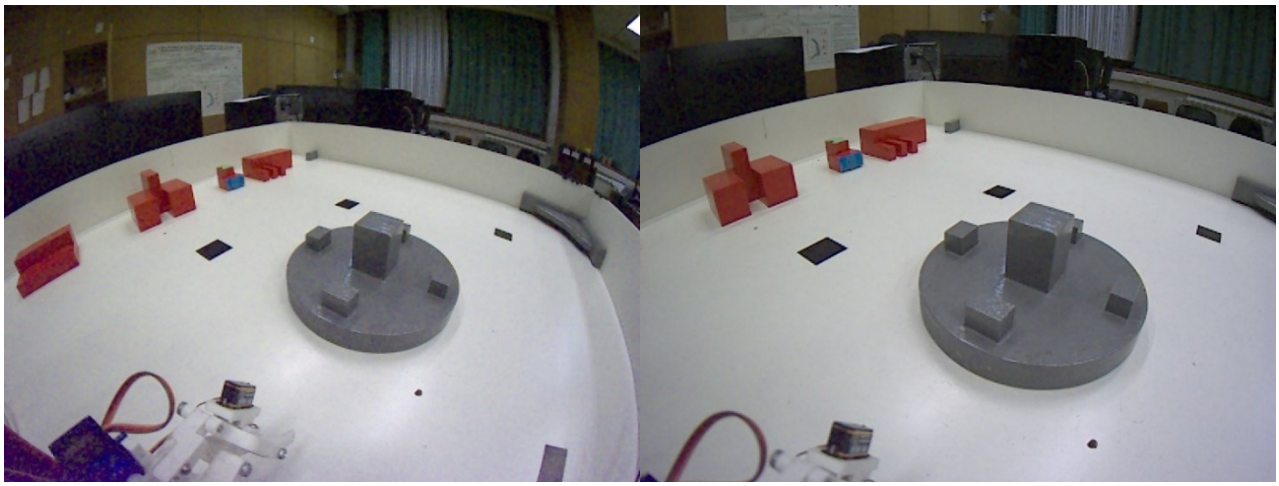
Табела 2.3. Техничка спецификација камере *Basler Dart daA1600-60uc*.

Произвођач сензора	*Teledyne e2v*
Тип сензора	CMOS
Димензије сензора	7,2 × 5,4 mm
Максимална резолуција	1600 × 1200 px
Димензије пиксела	4,5 × 4,5 μm
Максимална учестаност узроковања слика	60 FPS
Комуникација	USB 3.0
Прирубница за сочиво	S – Mount
Димензије кућишта	19,9 × 29,3 × 29 mm
Маса	15g

Како камере *Basler daA1600-60uc* немају интегрисано сочиво, потребно је одабрати и поставити компатибилно сочиво. При избору сочива, водило се рачуна о основним параметрима као што су фокална дужина (која утиче на видно поље камере) и врста прирубнице, као и о чињеници да је потребно сочиво за камере које генеришу слике у боји. Анализирана су и тестирана два сочива, чије се техничке спецификације могу пронаћи у табели 2.4.

Табела 2.4. Техничка спецификација анализираних сочива.		
	Evetar M118B029528W	*Evetar M118B0418W*
Фокална дужина	2,95 mm	4 mm
Максимални пречник слике	1/1,8“	1/1,8“
Максимални отвор бленде	F2,8	F1,8
Прирубница за сочиво	S - Mount	S - Mount
Максимални углови видног поље (X/B)*	143/106°	101/76°
Углови видног поља (X/B)* са камером *daA1600-60uc*	101,3/84,9°	84/68°
*X – хоризонтала, B – вертикала.		

Примарна разлика анализираних сочива је у њиховој фокалној дужини, што проузрокује различите вредности видног поља камере. Међутим, недостатак примене камере са већим вредностима углова видног поља огледа се у мањој густини пиксела по јединици површине, као и у већој дисторзији слике. Оба сочива припадају групи сочива код којих је наглашена дисторзија слике (енгл. *Fisheye lens*). Две слике, генерисане са различитим сочивима у истом положају мобилног робота, као и са истим параметрима аквизиције и претпроцесирања слике, приказане су на слици 2.9.

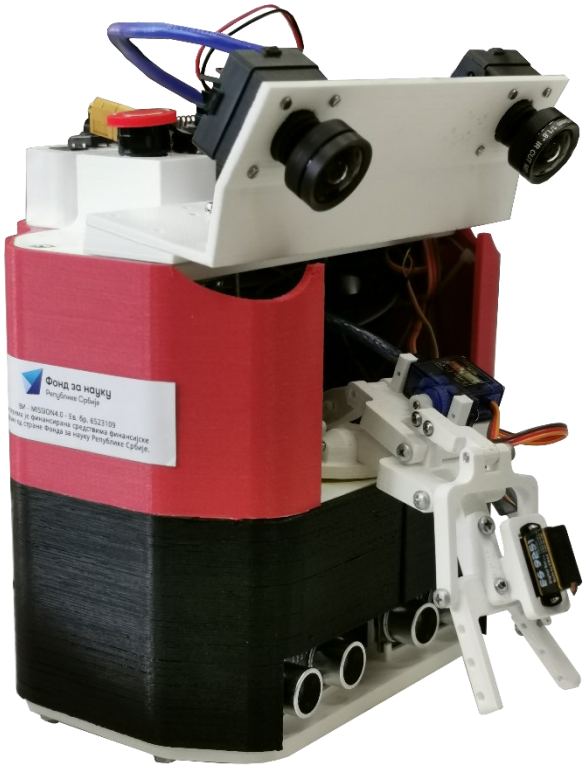


Слика 2.9. Слике генерисане коришћењем сочива *Evetar M118B029528W* (лево) и *Evetar M118B0418W* (десно).

Као што се може уочити са слике 2.9 (лево), са сочивом *Evetar M118B029528W* се остварује веће видно поље стерео визуелног система, али такође постоји и већи шум, као и знатно присутнија дисторзија слике. Са друге стране, слике генерисане коришћењем сочива *Evetar M118B0418W* покривају знатно мањи опсег (слика 2.9 десно) у лабораторијском моделу технолошког окружења, али је зато генерисана слика „оштрија“, са мање шума и са мањом дисторзијом. Како је видно поље камере са сочивом *Evetar M118B0418W* сасвим задовољавајуће за потребе управљања на основу стерео визуелног система мобилног робота RAICO, то сочиво се усваја као коначно решење и користи у финалној конфигурацији.

Како су одабране све кључне компоненте неопходне за пројектовање стерео визуелног система, извршена је њихова интеграција у одговарајуће кућиште и повезивање са интелигентним мобилним роботским системом RAICO. Носач камера пројектован је тако да

се састоји из једног дела, како би паралелност камера била на високом нивоу тачности. Такође, како би било могуће брзо извршити измену сочива, а тиме повећати флексибилност стерео система, сочива су постављена са предње стране роботског система (слика 2.10).



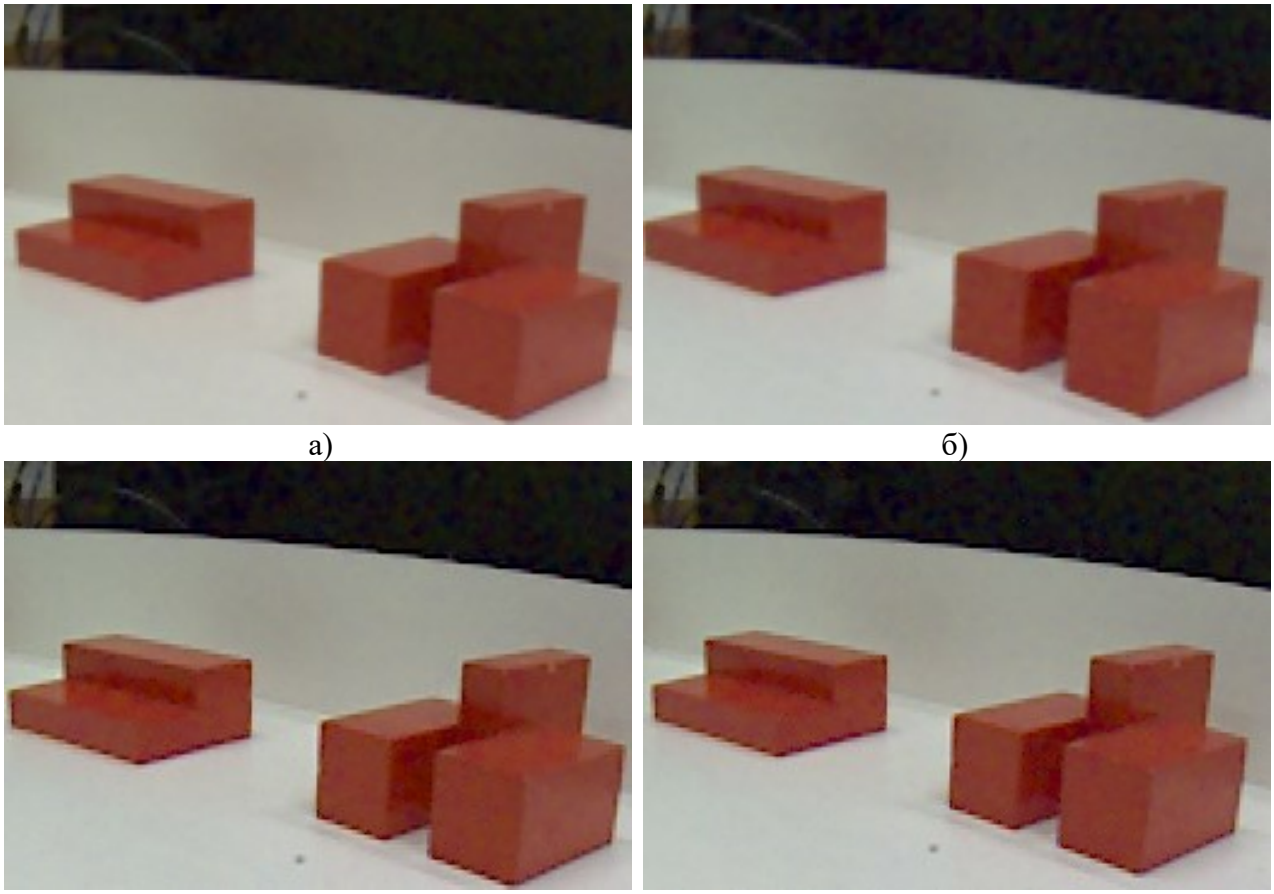
Слика 2.10. Мобилни роботски систем RAICO са новим стерео визуелним системом.

Након развоја хардверског дела стерео визуелног система, следећи корак процеса пројектовања била је анализа софтверских могућности камера. Како анализиране индустријске камере имају различите параметре везане за аквизицију слика, филтрирање шума и побољшање квалитета боја, неки од најутицајнијих параметара су дати и детаљно анализирани у наставку. Параметри који су подешени да се континуално аутоматски модификују током процеса аквизиције слика су:

1. Параметар који се користи за прилагођавање вредности интензитета осветљености пиксела – *Balancing White Auto*,
2. Параметар који прилагођава појачање камере, чиме се подешава ниво осветљености свих пиксела на слици до жељене средње вредности – *Gain Auto*,
3. Параметар који одређује временски период осветљавања сензора, чиме се такође подешава ниво осветљености свих пиксела на слици до жељене средње вредности – *Exposure Auto* и
4. Параметар којим се отклања негативни ефекат настао услед фреквенције наизменичне струје која утиче на вештачко осветљење у окружењу мобилног робота – *Auto Function Profile (AntiFlicker50Hz)*.

Предност континуалне модификације ових параметара је у томе што се превазилазе негативни утицаји промене осветљења на процес формирања слике. Уколико је због природног дневног светла ниво осветљености у просторији повећан, континуалном модификацијом параметара ће се појачање и период осветљавања сензора смањити, чиме ће просечан ниво осветљености пиксела генерисане слике остати у жељеном опсегу. Приказ слика формираних са другачије подешеним параметарима аквизиције дат је на слици 2.11. На слици 2.11а), где су прва три наведена параметра активна, може да се примети да бела боја технолошког окружења има примесе розе боје. Када се активира параметар *Auto Function Profile*, уочава се континуитет

боја у оквиру слика 2.11б), в) и г). Како се може уочити са слике 2.11а) и 2.11б), ивице објеката нису изоштрене па је додатно подешен параметар *Sharpness Enhancement* на вредност од 0,3, што видно утиче на „оштрину“ слика приказаних на 2.11в) и 2.11г). Недостатак примене овог параметра одражава се на повећање шума на слици, што се јасно може уочити на црној позадини. Остали параметри које је потребно подесити подразумевају: (i) ротацију слике, како би камере могле бити постављене у одговарајућем положају унутар кућишта, (ii) учестаност аквизиције слика, (iii) резолуцију слика, као и (iv) формат слике.



Слика 2.11. Слике генерисане са различитим параметрима камере.

Код погонског система, иако је било могуће поуздано управљати роботским системом, после одређеног времена експлоатације дошло је до знатног повећања зазора у редукторима мотора. Из тог разлога, појављују се грешке приликом управљања, чиме је проузрокована потреба за побољшањем погонског система, тј. увођењем квалитетнијих мотора са мањим зазором. У оквиру пројекта MISSION4.0 (*Deep Machine Learning and Swarm Intelligence-based Optimization Algorithms for Control and Scheduling of Cyber-Physical Systems in Industry 4.0*) обезбеђена су средства за набавку серво мотора *XL-430* компаније *Dynamixel*, који се одликују малим габаритима, поузданошћу и високом тачношћу. Погодност коју пружају *Dynamixel* мотори, поред већег обртног момента и мање масе, је у томе што су енкодери, као и целокупна управљачка јединица мотора, интегрисани у једну компактну целину па при коришћењу ових мотора није било потребе за додатним електронским компонентама. Такође, интегрисани управљачки систем подразумева три начина управљања мотором: управљање задавањем брзине, позиције или ширинском модулацијом снопа (енгл. *Pulse Width Modulation*), што је значајно убрзало интеграцију мотора у управљачки систем робота.

3. Систем перцепције мобилног робота

Систем перцепције подразумева интеграцију сензорског подсистема ради опажања промена и аквизиције података из окружења, са подсистемом за интерпретацију прикупљених података ради одређивања потребних семантичких и геометријских информација о стању окружења мобилног робота. Сензорски подсистем базира се на стерео визуелном систему за машинско гледање (који је детаљно приказан у поглављу 2). Такође, потребно је нагласити и основну разлику између визуелних система имплементираних у оквиру роботских система и стандардних компјутерских визуелних система. Роботски системи имају приступ знатно мањим процесорским ресурсима (у односу на радне станице за компјутерско гледање) које је неопходно определити свим подсистемима па се при интеграцији визуелног система мора водити рачуна и о његовој ефикасности и комплексности. Међутим, предност роботских визуелних система огледа се у томе да су роботи активни учесници у интеракцији са својим окружењем те се може одређеним акцијама повећати сигурност у сензорске информације (нпр. приближавањем жељеном предмету ради тачније детекције његових карактеристика). Ради обраде прикупљених података добијених од стерео визуелног система и екстракције информација које су од круцијалног значаја за извршавање задатака мобилног робота, алгоритми машинског учења показали су изузетан потенцијал и остварују висок ниво тачности у различитим задацима роботског гледања. Приликом развоја система перцепције мобилног робота, фокус је на развоју конволуционих неуронских мрежа за задатак семантичке сегментације, што ће бити детаљно приказано у потпоглављима 3.1-3.3.

3.1. Дубоко машинско учење – основе

Машинско учење (енгл. *Machine learning*) представља једну од области истраживања у оквиру домена вештачке интелигенције чија дефиниција гласи: „*Машинско учење представља процес обучавања одговарајућег параметарског модела система или процеса сходно усвојеном алгоритму обучавања и критеријуму перформанси*“ [7]. Дакле, на основу прикупљеног скупа података, врши се обучавање модела. Постоје три основна приступа машинског учења: супервизорско учење, несупервизорско учење и учење ојачавањем [13]. За све приступе машинског учења, превасходно је неопходно дефинисати критеријум перформанси, односно функцију циља (енгл. *cost function* или *fitness function*), на основу које се оцењује прогрес учења. Код супервизорског учења, моделу који се обучава поред улазног скупа података дат је и скуп података који представља тачан одзив система за сваки улазни вектор понаособ. Овај приступ генерише највиши ниво тачности (у поређењу са осталим приступима), међутим, потребна је и највећа припрема података јер се за сваки улаз мора експлицитно дефинисати очекивани излаз модела. Код несупервизорског приступа обезбеђује се само улазни скуп података, а од модела се очекује да самостално (на основу података) препозна карактеристичне одлике улазних вектора. Машинско учење ојачавањем представља приступ код кога се дефинише скуп могућих акција и на основу њиховог утицаја на окружење интелигентни агент учи из сопственог искуства. У оквиру предметне докторске дисертације фокус ће бити постављен на супервизорске методе машинског учења. Најзаступљенији модели машинског учења су вештачке неуронске мреже (енгл. *Artificial Neural Networks*) чија дефиниција гласи: „*Вештачка неуронска мрежа је парадигма вештачке интелигенције која се дефинише као конективни модел за резоновање заснован на аналогији са мозгом, уз наглашену когнитивну способност да учи и врши генерализацију стеченог знања*“ [2].

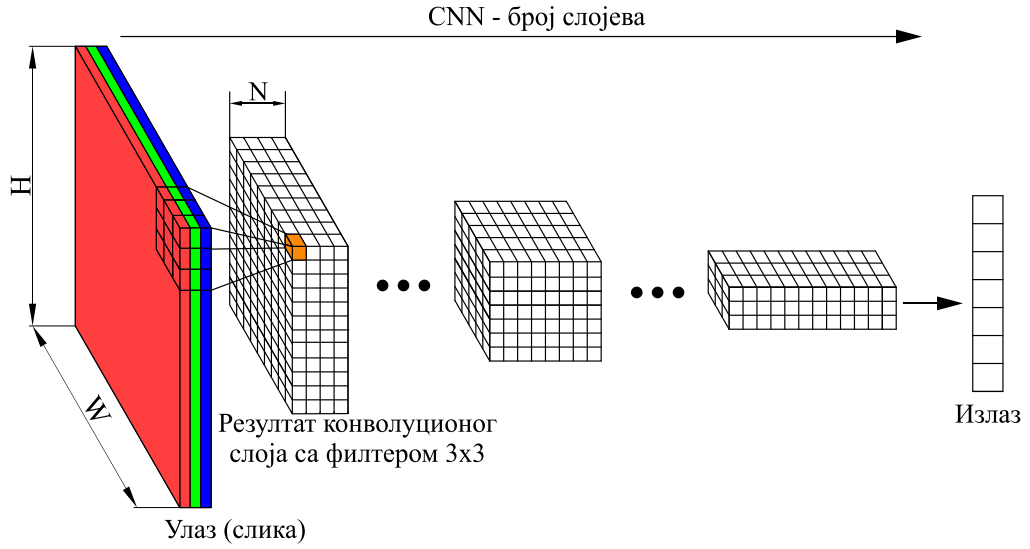
Области машинског учења и вештачке интелигенције, од својих почетака средином XX века, пролазиле су кроз деценијске периоде великих очекивања, као и периоде недовољних остварења исхода истраживања [14]. Објављивање нове архитектуре вештачких неуронских мрежа (назване *AlexNet*) у оквиру рада [15] 2012. године сматра се почетком развоја дубоког машинског учења. Ентузијазам научне заједнице за примену техника вештачке интелигенције

у периоду после 2012. године најбоље је приказан кроз чињеницу да је рад [15] цитиран преко 140.000 пута (извор *Google Scholar*) током 13 година од његовог објављивања. Главна предност архитектуре *AlexNet* у односу на све друге алгоритме је у изузетно високој тачности на проблему класификације слика на највећем скупу података (*ImageNet*) [16] који је до тада прикупљен у сврхе машинског гледања. Од тада многи аутори наставили су са развојем вештачких неуронских мрежа по узору на *AlexNet* (*VGG* [17], *ResNet* [18] и *Xception* [19]), а та група модела названа је конволуционе неуронске мреже (енгл. *Convolutional Neural Networks*). Како се ове архитектуре одликују великим бројем слојева, целокупна област везана за њихов развој, обучавање и имплементацију названа је дубоко машинско учење (енгл. *Deep Learning* – DL). Међутим, прве CNN архитектуре (односно CNN модели) предложене су још крајем прошлог века [20] па је потребно анализирати како се тек сада дошло до њихове примене у пуном капацитету. Првенствено, за обучавање CNN модела потребни су изузетно велики скупови података. Многи други алгоритми остварују већу тачност када су коришћени скупови података мали (ред величина стотине или хиљаде улазних вектора), међутим, за *ImageNet* скуп података, који има око 3,2 милиона слика разврстаних у 1.000 различитих категорија (класа), CNN модели су се показали као значајно тачнији. Друга чињеница која указује на то зашто су CNN модели тек у другој деценији XXI века доживели своју широку примену је у потреби за великим процесорским ресурсима за њихово обучавање, а опште је прихваћено да се ови ресурси дуплирају сваке године на основу Муровог закона [21]. Још једна предност DL модела је у томе што није потребно из слика издвајати карактеристике па на основу њих обучавати модел (као што је то случај код других техника машинског учења), већ се целокупне слике директно доводе на улаз модела, што знатно олакшава сам процес обучавања.

Модели дубоког машинског учења разликују се по комбинацији, редоследу и броју слојева који чине целокупну архитектуру. Слојеви у оквиру модела дубоког машинског учења могу вршити различите функције у оквиру мреже па се на основу тога могу поделити у следеће категорије:

- Слојеви конволуције (енгл. *Convolution layers*),
- Слојеви сажимања (енгл. *Pooling layers*),
- Слојеви активације,
- Потпуно повезани слојеви (енгл. *Fully Connected layers*) и
- Слојеви регуларизације.

Свака од наведених категорија може укључити различите слојеве који имају сличну намену, а такође неки слојеви могу извршавати и функцију других слојева (нпр. конволуциони слој може извршити функцију сажимања уколико су одређени параметри подешени на такав начин). Генерализована структура CNN модела за задатак класификације може се уочити на слици 3.1.



Слика 3.1. Генерализована структура CNN мреже за задатак класификације слика.

Као што се може уочити на основу слике 3.1, резолуција улазног тензора је највећа (W, H) и са слојевима се смањује, док се трећа димензија тензора повећава. Излаз на крају представља само један вектор који дефинише класу. Како се конволуционе неуронске мреже дефинишу као мреже које садрже бар један конволуциони слој, овај слој ће бити први представљен. Иако се DL модели могу применити за различите типове улазних података, примена CNN-а у оквиру ове докторске дисертације је у системима вештачког гледања па ће као пример улаза бити коришћена слика. Слика у боји представља тензор $W \times H \times 3$, где је W број пиксела по ширини слике, H број пиксела по висини (параметри W и H заједно дефинишу резолуцију слике), док се у оквиру треће димензије дефинишу нијансе црвене, зелене и плаве боје сваког пиксела.

Задатак конволуционог слоја је да из улазних података, након завршеног процеса дубоког машинског учења, издвоји одређене карактеристике (нпр. линије, ивице или контуре) на основу којих је могуће извршити постављени задатак (нпр. класификацију слика или детекцију објеката на сликама). Као што се може уочити са слике 3.1, слика је велике резолуције и тада се могу детектовати само димензионо мали карактеристични објекти. Међутим, како се процес наставља, улазном тензору се смањује резолуција и филтер у конволуционом слоју може да издвоји (детектује) и веће карактеристике са слике.

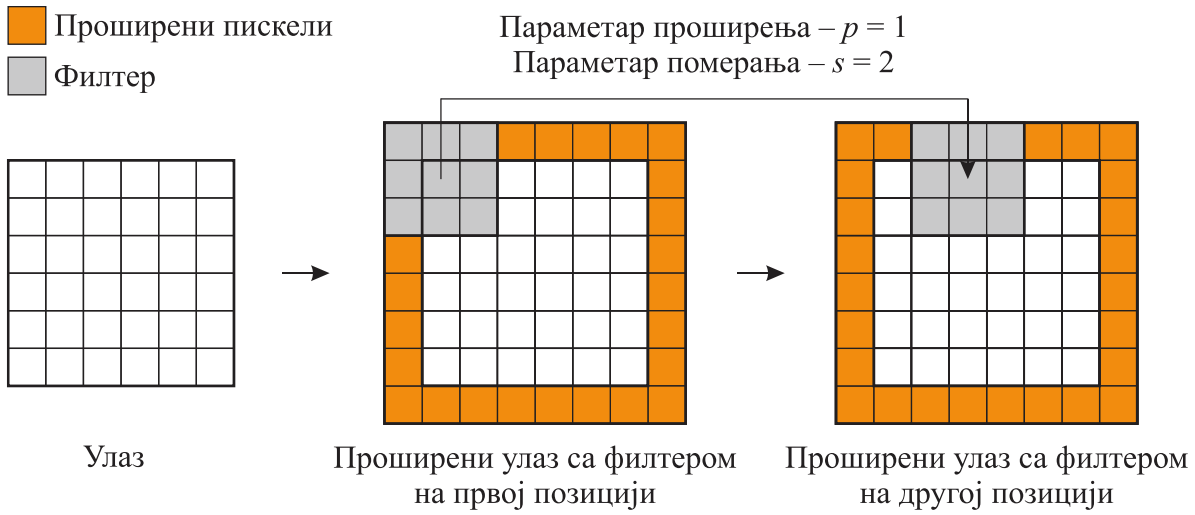
Над целокупним улазним тензором извршава се процес обраде података применом конволуционог филтера. На слици 3.1 може се уочити како је постављен филтер 3×3 у првој позицији. С обзиром на то да улазни тензор има трећу димензију једнаку три, целокупан филтер ће имати 27 тежинских вредности, односно он ће представљати тензор 3×3×3. Свака од вредности у оквиру филтера множи се са одговарајућом вредношћу улазног тензора. Уколико улазни и излазни тензор конволуционог слоја имају исту резолуцију и постоји N филтера, једначина за једну позицију процеса конволуције дата је (3.1):

$$net_{i,k} = \sum_{j=1}^{27} w_j x_j + \theta_{\mathrm{k}}, i = 1,...,WH; k = 1,...,N\,, \tag{3.1}$$

где је $net_{i,k}$ вредност i-тог елемента излазног тензора за k-ти филтер, w_j су тежински односи филтера, x_j су вредности улазног тензора, а $\theta_{\mathrm{к}}$ је вредност бијас неурона.

Затим се филтер помера по једној оси слике и процес се наставља. Број пиксела за који се филтер помера дефинисан је параметром померања (енгл. *Stride*). Овај параметар мора имати целобројну вредност и може се дефинисати са сваку осу понаособ. Што је параметар померања већи, резолуција излазног тензора биће мања. Такође, како би могла да се одржи константна резолуција излазног тензора, уведен је додатни параметар проширења (енгл. *Padding*), којим се повећа резолуција улазног тензора најчешће постављањем вредности нула

у те додатне елементе тензора. Приказ утицаја параметара померања и проширења на филтер у 2D може се видети на слици 3.2.



Слика 3.2. Параметри конволуционог слоја.

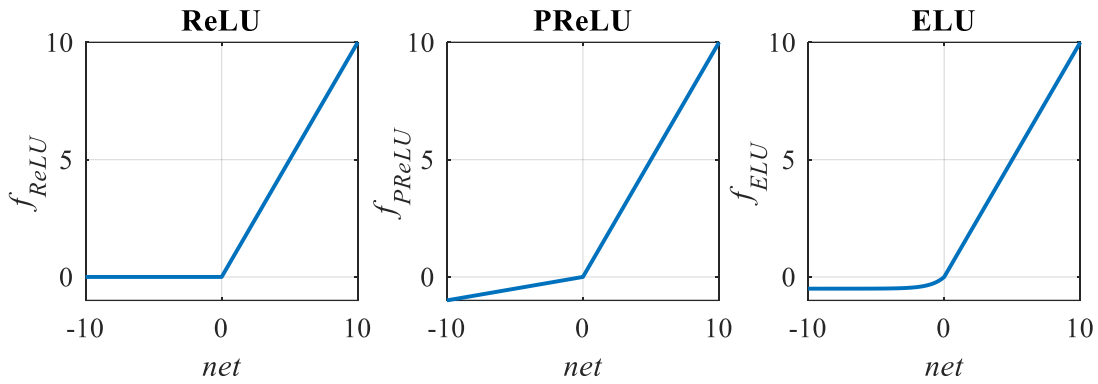
Резолуција излазног тензора зависи од поменутих параметара, а трећа димензија од броја различитих филтера (N) који се користе у том конволуционим слоју. За пример са слике 3.1, први слој има четири филтера, тј. укупно постоји 27·4 тежинских коефицијента који треба да се подесе (на основу процеса дубоког машинског учења) у оквиру тог слоја. Наредни конволуциони слој са слике 3.1 би имао филтер димензија 3×3×4 и 8 различитих филтера, чиме се број параметара који је потребно научити значајно повећава. Постоје и специфичне врсте конволуционих слојева који имају додатне параметре или се неки део процеса другачије изводи, а неки од њих су: Атрос конволуција [22], транспонована конволуција [23], конволуција по резолуцији [24] и 1D конволуција [24].

Слојеви сажимања служе како би се смањила резолуција тренутног тензора у оквиру CNN модела. Смањење резолуције је некад неопходно извршити како би се смањили процесорски ресурси потребни за генерисање излаза из мреже, што је потребно како при обучавању мреже, због смањења времена обучавања, тако и при имплементацији, како би мрежа могла бити примењена у реалном времену. Филтер се креће кроз улазни тензор (исто као и код конволуционог слоја) и бира се највећа (енгл. *Max pooling*) или просечна вредност (енгл. *Average pooling*) свих параметара у улазном тензору како би се генерисала једна вредност у излазном тензору. Параметри проширења и померања дефинишу се идентично као и код конволуционог слоја. Једначине за највеће и просечно сажимање једне позиције филтера димензије 3×3×3 означене су са (3.2) и (3.3), респективно:

$$mpool = \max_{j=1,\dots,27}\left(x_j\right), \tag{3.2}$$

$$apool = \underset{j=1,\dots,27}{avg}\left(x_j\right). \tag{3.3}$$

Слојеви активације обично су постављени одмах након конволуционог слоја и служе како би се генерисали излази неурона. Иако су у оквиру потпуно повезаних неуронских мрежа углавном коришћене активационе функције попут сигмоидне или хиперболичког тангенса, код CNN модела су се као најбоље показале позитивне линеарне активационе функције (енгл. *Rectified Linear Unit layer* – ReLU, (3.4)) [25], [26]. Постоје различите форме ReLU активационе функције (слика 3.3), а неке од најчешће коришћених су параметризована ReLU (PreLU, (3.5)) [27] и експоненцијална ReLU функција (ELU, (3.6)) [28]:



Слика 3.3. ReLU активационе функције.

$$f_{ReLU} = \max\left(net_{i,k}, 0\right), \tag{3.4}$$

$$f_{PReLU} = \max\left(net_{i,k}, net_{i,k} \cdot 0{,}1\right), \tag{3.5}$$

$$f_{ELU} = \begin{cases} net_{i,k} & , net_{i,k} \geq 0 \\ \alpha\left(e^{net_{i,k}} - 1\right), & net_{i,k} < 0 \end{cases}, \alpha = 0{,}5\,. \tag{3.6}$$

где је са *net* означен улаз у активациону функцију.

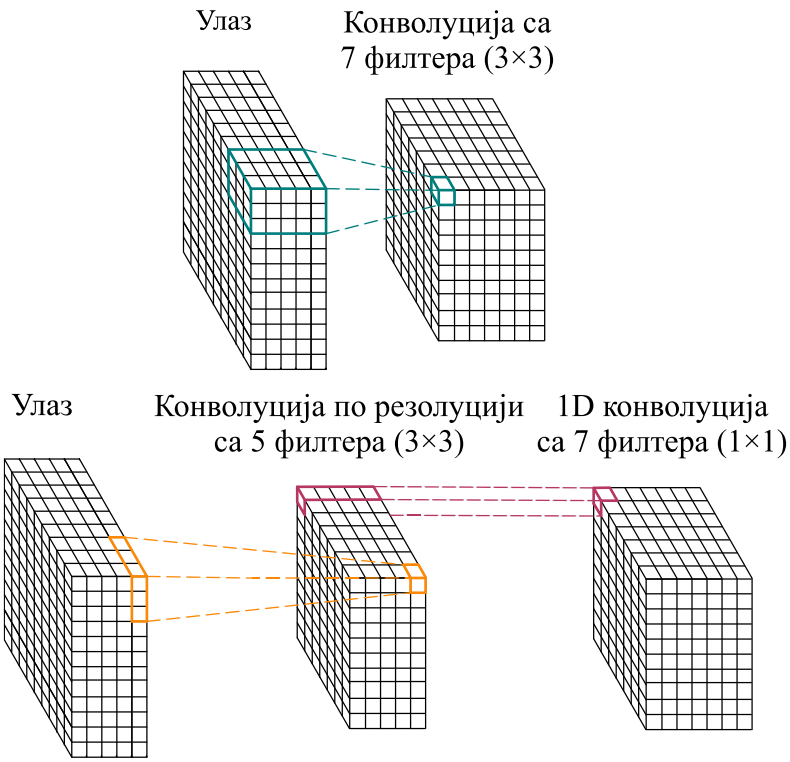
Наредна група подразумева потпуно повезане слојеве, код којих су сви елементи тензора у претходном слоју директно повезани са свим елементима тензора наредног слоја. Како се овим слојем генерише изузетно велики број тежинских коефицијената, ови слојеви нису често у употреби при дубоком машинском учењу. Међутим, код неких архитектура попут *VGG* и *AlexNet*, последњи слој је потпуно повезан како би све издвојене карактеристике утицале на све елементе излазног слоја. Такође, потпуно повезани слојеви показали су се значајним при примени механизма пажње (енгл. *Attention mechanism* [29]) у трансформационим (енгл. *Transformers* [30]) неуронским мрежама, које у последње време постају све популарније [31].

Последња категорија слојева који се користе у оквиру CNN модела су слојеви регуларизације. Један од проблема при обучавању, како модела машинског учења, тако и CNN мрежа, је преобучавање (енгл. *Overfitting*). Преобучавање настаје када је модел машинског учења у стању да адекватно генерише излазе за све обучавајуће парове у скупу за обучавање, али нема могућност генерализације и показује знатно нижу тачност на скупу података за валидацију (или тестирање). Како би се спречио проблем преобучавања, развијени су слојеви за регуларизацију. Изостављајући слој (енгл. *Dropout* layer) [32] је слој који при обучавању CNN модела, у различитим епохама учења, изоставља повезаност случајно одабраних неурона. Процесом изостављања онемогућује се превелик фокус на одређене тежинске коефицијенте CNN модела и поспешује се учење користећи различите везе унутар мреже. Други слој који се може користити за регуларизацију је нормализација тренутне групе улаза (енгл. *Batch normalization*) [33]. Како су скупови података за обучавање CNN модела изузетно велики, не може се одједном целокупни скуп користити за обучавање, већ се скуп дели на мање групе, зависно од доступне меморије на графичкој картици на којој се врши обучавање. У оквиру слоја нормализације, тренутни улазни тензори се нормализују у зависности од целокупне групе и тиме се остварује стабилнији и бржи процес обучавања.

3.2. Ефикасни модели дубоког машинског учења

Успешност примене DL модела у реалном технолошком окружењу зависи од различитих фактора. Првенствено, она зависи од репрезентативности скупа података који је прикупљен и коришћен, што утиче на тачност модела при имплементацији на реалном систему. Секундарни фактор који утиче је могућност имплементације обученог модела у систем са ограниченим рачунарским ресурсима, какви су роботски и остали кибернетско физички системи у технолошком окружењу. Најмодернији модели дубоког учења обучавају се данима, а некад и месецима на супер компјутерима изузетних рачунарских могућности. Једна анализа [34] обухватила је оцену потрошње енергије потребне за процес обучавања модела за разумевање људског говора, долазећи до закључка да је потрошено чак 1,287 MWh. Имајући то у виду, истраживачи су почели да развијају моделе који су доста ефикаснији, како при обучавању, тако и при имплементацији модела дубоког машинског учења. Ефикасност CNN модела одређује се на основу броја компјутерских операција у покретном зарезу (енгл. *FLoating Point Operations* – FLOPs) потребним за генерисање излаза на основу одговарајућег улаза. Друга метрика коју је могуће користити, искључиво уколико су CNN модели поређени на истом хардверском систему, је време потребно за генерисање излаза мерено у ms. У области роботике, време потребно за генерисање излаза је релевантнија метрика, јер се директно може закључити да ли је неки модел могуће применити у реалном времену. Када је реч о апликацијама у роботском гледању, уместо времена потребног за обраду једне слике, могуће је моделе поредити и применом броја слика које се обраде у секунди (енгл. *Frames Per Second* – FPS). За све моделе који могу да обраде више од 30 слика у секунди, сматра се да раде у реалном времену.

Један од првих ефикасних CNN модела предложен је у раду [35] и назван *SqueezeNet*. Пре овог рада, у оквиру CNN модела често су били имплементирани конволуциони филтери великих димензија 7×7 или 5×5. У оквиру новог модела, аутори су предложили смањење броја параметара потребних за обучавање коришћењем филтера резолуције 1×1 и 3×3. Тачност је одржана смањењем резолуције у каснијим слојевима, а основни градивни елемент ове мреже је тзв. ватрени модул који се састоји од два конволуциона слоја 1×1, након чега се извршава конволуција са филтером 3×3 и ReLU активација. Предложена мрежа остварује тачност сличног нивоа као и *AlexNet* са приближно 50 пута мање параметара. У оквиру рада [24], предложена је ефикаснија имплементација конволуционог слоја, на основу кога је могуће значајно смањити број FLOPs потребних за његово извршавање применом конволуција по резолуцији и 1D конволуције. Уместо истовремене конволуције читавог улазног тензора (што је приказано на горњем делу слике 3.4), извршавају се два одвојена процеса конволуције (доњи део слике 3.4). Прво се извршава конволуција по резолуцији, у оквиру које се смањи резолуција улазног тензора, док трећа димензија остаје иста. Затим, извршава се 1D конволуција у оквиру које се тензор доводи на задате димензије (слика 3.4).



Слика 3.4. Разлика између конволуција.

CNN модел дефинисан на основу поменутих конволуционих слојева назван је *MobileNet*. За пример са слике 3.4, дате су једначине (3.7) и (3.8) које приказују ефикасност новог конволуционог слоја са аспекта коришћених параметара (броја тежинских коефицијената):

$$P_c = F_{wc}F_{hc}M_cN_c = 3\cdot 3\cdot 5\cdot 7 = 315\,, \tag{3.7}$$

$$P_m = P_d + P_p = F_{wd}F_{hd}M_dN_d + F_{wp}F_{hp}M_pN_p = 3\cdot 3\cdot 1\cdot 5 + 1\cdot 1\cdot 5\cdot 7 = 80\,, \tag{3.8}$$

где је са P означен број параметара, F_w и F_h су димензије филтера, M је трећа димензија улазног тензора, N представља број филтера (трећа димензија излазног тензора), док индекси свих параметара означавају: c – стандардна конволуција, d – конволуција по резолуцији, p – 1D конволуција, m – *MobileNet* CNN модел. За исти пример, дате су једначине које указују на то колико је FLOPs потребно за извршавање конволуционих слојева (бијас није узет у разматрање, параметар проширења је 0, а параметар померања износи 1):

$$F_c = F_{wc}F_{hc}M_cD_wD_hN_c = 3\cdot 3\cdot 5\cdot 10\cdot 7\cdot 7 = 22050\,, \tag{3.9}$$

$$\begin{aligned} F_m = P_d + P_p = F_{wd}F_{hd}M_dN_dD_wD_h + F_{wp}F_{hp}M_pN_pD_wD_h = \\ 3\cdot 3\cdot 1\cdot 5\cdot 10\cdot 7 + 1\cdot 1\cdot 5\cdot 7\cdot 10\cdot 7 = 3150 + 2450 = 5600 \end{aligned}\,, \tag{3.10}$$

где је са F означен број FLOPs целокупног слоја, D_w и D_h су димензије излазног тензора. Као што се може уочити, број параметара који утиче на меморијске перформансе и време потребно за извршавање слоја дефинисано у FLOPs је приближно 4 пута мањи за конволуциони *MobileNet* слој. Такође, како зависност није линеарна, за слојеве већих димензија разлика може бити и значајнија, чак и до 8 пута [24]. Још један анализирани ефикасни CNN модел је *ShuffleNet* предложен у раду [36]. Код овог модела користи се конволуција по резолуцији и 1D конволуција, али са примењеном групном конволуцијом која такође додатно повећава ефикасност слоја. Групна конволуција подразумева да се целокупна група улаза подели на одређени број сегмената и да се конволуција врши на сваком сегменту посебно. Затим је додат слој којим се замене вредности у различитим сегментима, чиме се поспешује дељење информација унутар сегмената улаза. Применом *ShuffleNet*-а повећава се тачност модела (у

односу на *MobileNet*), док ефикасност остаје иста. Оба поменута модела имају новије унапређене архитектуре представљене у радовима [37] и [38]. Унапређење ефикасности CNN модела остварено је применом додатних операција у конволуционом слоју у оквиру *GhostNet* архитектуре [39]. Како је уочена сличност у вредностима улазног тензора конволуционих слојева, аутори предлажу умањење броја FLOPs-а применом линеарних оператора. Први део излазног тензора добија се стандардним процесом конволуција, док се њиховим линеарном трансформацијом добија други део. Интеграцијом оваквог типа конволуција, добија се већа тачност и краће време процесирања у односу на *MobileNetV2* и *ShuffleNetV2*. Аутори рада [40] предлажу нови ефикасни CNN модел *FasterNet* који се базира на парцијалној конволуцији. Парцијална конволуција подразумева да се процес конволуције извршава само на делу улазног тензора, што је оправдано због велике редундантности у оквиру улазног тензора. Након парцијалне конволуције, извршава се 1D парцијална конволуција како би сви подаци из улазног тензора утицали на наредне слојеве. Предност парцијалне конволуције огледа се у смањењу броја приступа меморијским ресурсима, што аутори закључују да постаје уско грло модерних ефикасних CNN модела. Експерименталном верификацијом на GPU доказано је да предложени модел остварује приближно 1,5 пута брже процесирање од *ShuffleNetV2* и *MobileNetV2*.

3.3. Семантичка сегментација

Семантичка сегментација представља поступак додељивања класе сваком пикселу на слици. Резултат семантичке сегментације је семантичка мапа која је представљена матрицом димензија улазне слике, а у оквиру које је дефинисана класа сваког пиксела. Тачније, семантичка мапа обезбеђује информацију о томе ком објекту припада сваки пиксел на слици. За разлику од стандардне сегментације слика, где се слика групише у различите категорије, семантичка сегментација обезбеђује класе које су смислене за људе нпр. столови, столице, зидови итд. Ова процедура често се користи у оквиру роботских апликација јер пружа информације о сцени које могу бити искоришћене за извршавање различитих задатака [41], [42]. Информације о постајању објеката у сцени и њихове узајамне позиционе зависности могу бити коришћене за одређивање семантичке локализације [43]. На основу препознавања пиксела који припадају подлози, могуће је извршити планирање кретања или избегавање препрека [44]. Одређивање позиције и оријентације објекта на слици може значајно повећати успешност процеса аутономног роботског хватања. Такође, користећи поменуте информације, робот може да доноси одлуке о даљим акцијама које треба предузети како би се постигао постављени циљ. Пример семантичке сегментације на слици генерисаној у оквиру Лабораторије за роботику и вештачку интелигенцију Универзитета у Београду – Машинског факултета приказан је на слици 3.5. Како је семантичка мапа у оквиру слике 3.5 генерисана применом CNN модела, она није апсолутно тачна.

Слика 3.5. Семантичка сегментација слике, приказана додавањем боје преко дела слике у коме су предмети који дефинишу различите класе објеката; на пример, жутом бојом је означен под, црвеном столови, зеленом столице, наранџастом зидови, светло црвеном завесе итд.

Тачност семантичке сегментације одређује се на основу различитих метрика за чије дефинисање је потребно увести неколико појмова. Поменути појмови зависе од семантичке мапе коју је генерисао модел дубоког машинског учења и потпуно тачне семантичке мапе, а одређују се за сваку класу, тј. сваки објекат понаособ:

- *tp* – тачно позитиван (енгл. *true positive*), када је пиксел класификован као тренутни објекат на обе семантичке мапе,
- *tn* – тачно негативан (енгл. *true negative*), када је пиксел класификован као одређени други објекат (друга класа) на обе семантичке мапе,
- *fp* – лажно позитиван (енгл. *false positives*) када је пиксел класификован као тренутни објекат на излазу модела дубоког машинског учења, а пиксел припада другом објекту на тачној семантичкој мапи,
- *fn* – лажно негативан (енгл. *false negatives*) када је пиксел класификован да припада другом објекту на излазу модела дубоког машинског учења, а пиксел припада тренутном објекту на тачној семантичкој мапи.

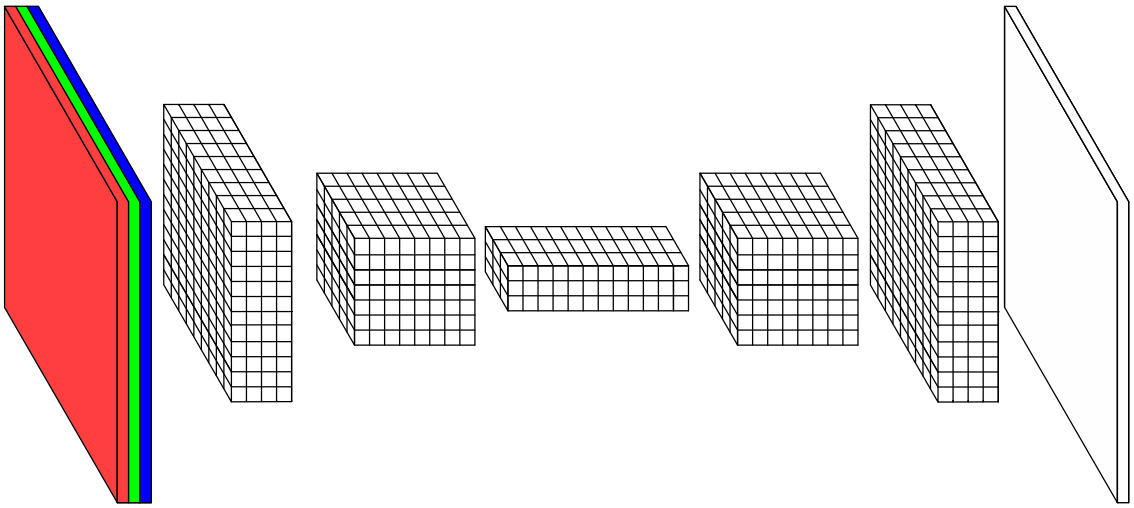
Најзаступљеније метрике су тачност (енгл. *Global Accuracy* – GA, једначина 3.11) и средњи количник пресека и уније (енгл. *mean Intersection over Union* – mIoU, једначина 3.12):

$$GA = \sum_{d=1}^{k} \sum_{i=1}^{N} tp_{k,i} \,, \tag{3.11}$$

$$mIoU = \frac{1}{k}\sum_{d=1}^{k}\frac{\sum_{i=1}^{N} tp_{k,i}}{\sum_{i=1}^{N} tp_{k,i} + \sum_{i=1}^{N} fp_{k,i} + \sum_{i=1}^{N} fn_{k,i}}, \tag{3.12}$$

где је са N обележен укупан број пиксела у семантичкој мапи, а са k укупан број класа (различитих објеката). Иако се обе метрике користе при одређивању тачности модела за семантичку сегментацију, mIoU је заступљенија због реалистичније оцене тачности сегментације за скупове података код којих постоји велика разлика у количини података за различите класе.

У последњих десетак година, технике дубоког машинског учења, а поготово CNN мреже, истакле су се као алгоритми највише тачности који се користе за задатке семантичке сегментације [45]. Бројне CNN архитектуре могу извршити овај задатак са задовољавајућом тачношћу. Главна модификација, у односу на архитектуре за класификацију, која се мора увести када се врши семантичка сегментација је та да димензије излаза морају бити исте као димензије улаза. Из тога разлога, CNN модели за семантичку сегментацију углавном имају два дела архитектуре: први у коме се димензије тензора смањују (енгл. *encoder*), а други у коме се димензије повећавају (енгл. *decoder* – слика 3.6). Постоји два разлога због којих је неопходно смањити димензије слике: први разлог везан је за знатно смањивање процесорских ресурса потребних ради генерисања излаза, а други се односи на могућност препознавања објеката који заузимају велики део тренутне сцене.



Слика 3.6. Генеричка архитектура CNN модела за семантичку сегментацију.

Једна од често коришћених CNN архитектура за семантичку сегментацију је вештачка неуронска мрежа са искључиво конволуционим слојевима (енгл. *Fully Convolutional Network* – FCN) [46] јер њена флексибилност омогућава брзу интеграцију у различите системе. Главна предност FCN модела је могућност промене величине улаза без потребе за адаптацијом архитектуре, што се остварује коришћењем само конволуционих слојева.

Поред FCN модела за семантичку сегментацију, у литератури је развијен значајан број ефикаснијих архитектура са циљем обезбеђивања примене у реалном времену. Прва анализирана мрежа је *Enet* [47], у оквиру које су аутори унапредили *ResNet* модел повећавајући редукцију резолуције тензора излаза са слојем сажимања или слојем конволуције са параметром померања 2. Нормализација групе улаза додата је ради регуларизације после сваког слоја конволуције. Резултати су показали да предложени CNN модел остварује тачност сличну као најбоље *state-of-the-art* архитектуре, са далеко већом ефикасношћу, што такође

указује чињеница да је остварен 21 фрејм у секунди (енгл. *Frames Per Second* – FPS) на хардверској платформи *Jetson TX1*. Тренд развоја ефикасних мрежа се наставио и следећи је представљен *ERFNet* модел [48]. Унапређење је остварено раздвајањем конволуције са филтером n×n, на два слоја са филтерима n×1 и 1×n, што представља сличну логику развоја која је предложена у радовима [24] и [36] анализираним у потпоглављу 3.2. Остварена је виша тачност него код *Enet*-а, међутим са мањом ефикасношћу. Аутори рада [49] представљају CNN модел са механизмом брзе пажње (енгл. *fast attention*). Предложени модел остварио је висок ниво тачности са чак 75,0 mIoU, док је при имплементацији на хардверску платформу *Jetson Nano* остварио 31 FPS. Резултати овог рада су побољшани у односу на остале остварене у том периоду. Још један значајни модел предложен је у раду [50] у оквиру кога је интегрисана *U-net* [51] архитектура са резидуалним везама предложеним у оквиру мреже *ResNet*. Однос тачности и ефикасности остварен је применом итеративног алгоритма оптимизације, који дефинише параметар компресије модела. Предложена мрежа остварила је изузетно високу ефикасност од 25 FPSа на *Jetson Nano* платформи са тачношћу од 52,72 mIOU.

3.4. Развој система перцепције мобилног робота

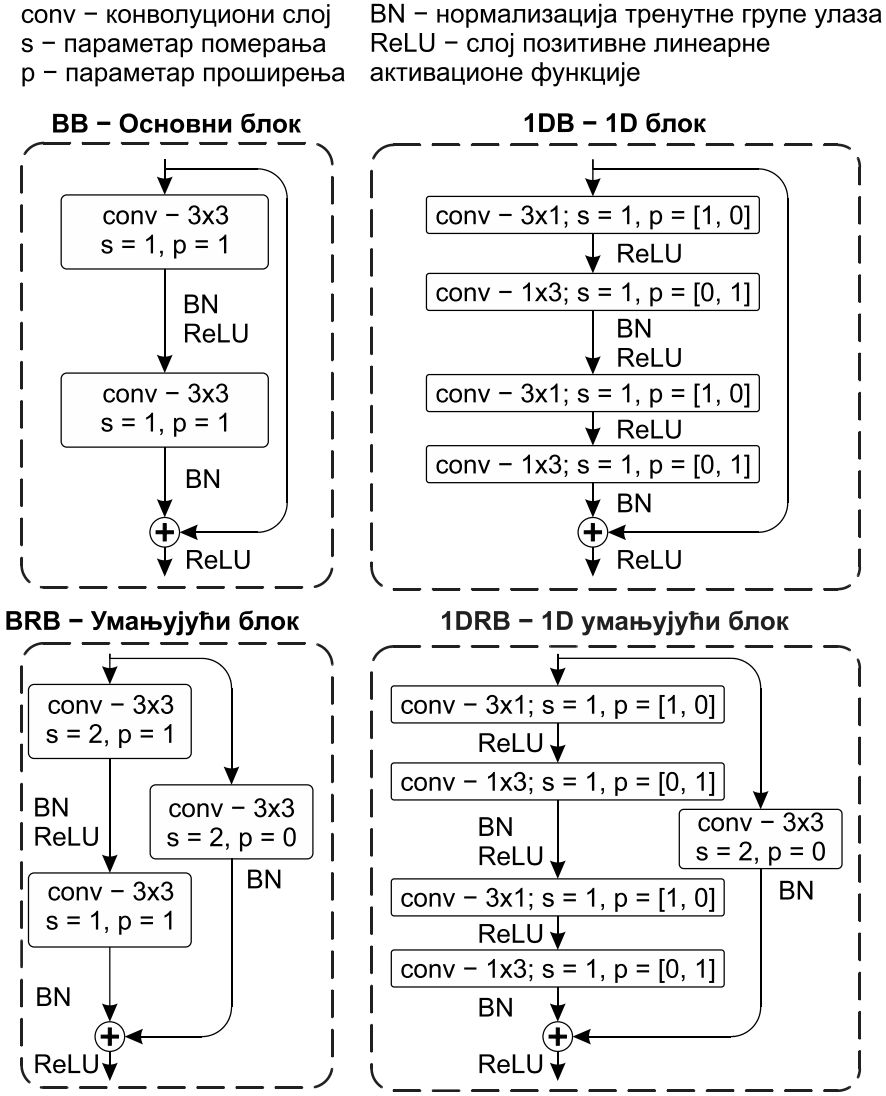
Остваривање аутономног понашања мобилних роботских система у недовољно познатом технолошком окружењу у великој мери зависи од података добијених од сензорског подсистема, као и од начина њиховог процесирања. Резултат аквизиције и обраде података требало би мобилном роботу да омогући опажање његовог тренутног окружења, препознавање технолошко-производних ентитета, као и њихову локализацију у сцени. На тај начин, обезбеђена је могућност планирања акција за наредне технолошко-транспортне операције које мобилни робот треба да изврши. Такође, како су процесорски ресурси чак и најмодернијих мобилних роботских система ограничени, систем перцепције такође треба да буде ефикасан у довољној мери, тј. требало би обезбедити могућност рада у реалном времену, имајући у виду да систем перцепције мора бити интегрисан паралелно са осталим алгоритмима који обезбеђују аутономност роботског система.

У овом раду, систем перцепције развијен је тако да се омогући имплементација у реалном времену на мобилни роботски систем RAICO, тј. на рачунарску платформу *NVIDIA Jetson Nano* [44]. Како би резултати били поновљиви, обучавање је извршено на *cityscapes* скупу података [52], који представља један од најчешће коришћених скупова података за семантичку сегментацију. Улазне слике у боји (енгл. *Red Green Blue* – RGB) имају резолуцију од 512×256 пиксела. У наставку овог потпоглавља, биће анализирани различити слојеви који представљају градивне блокове целокупног CNN модела.

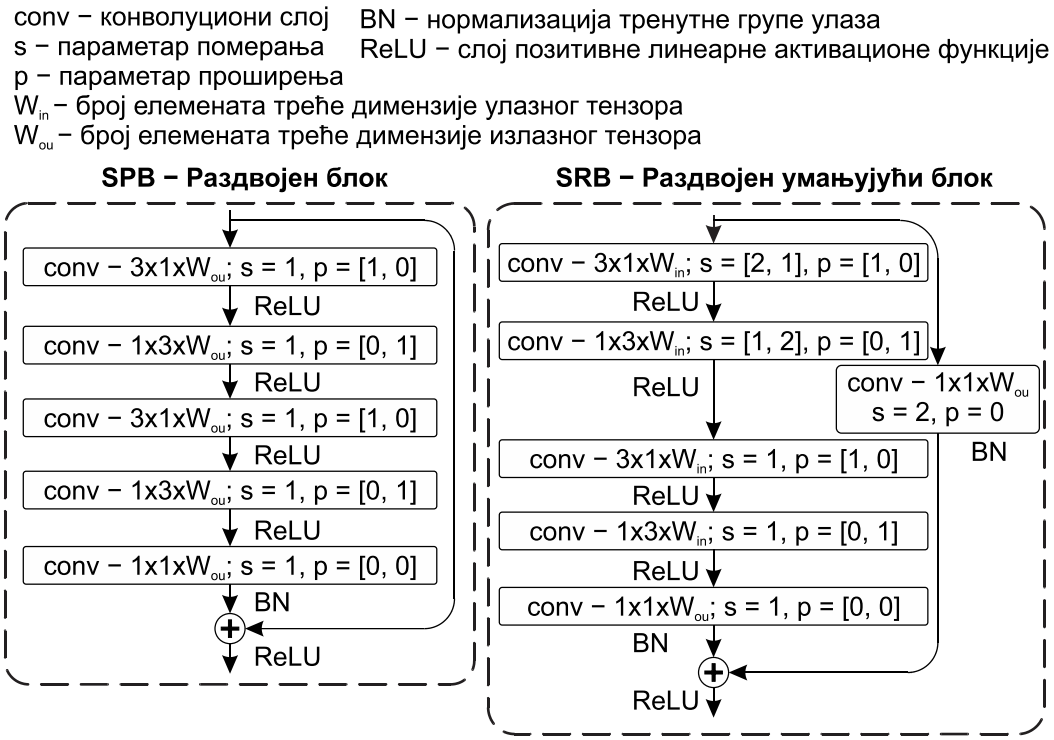
Структура на основу које је започет развој новог CNN модела за семантичку сегментацију базира се искључиво на конволуционој *ResNet18* [18] архитектури предложеној у [53]. Због ограничених ресурса, семантичка мапа неће бити исте резолуције као улазна слика, што проузрокује да ће један пиксел семантичке мапе означавати неколико пиксела улазне слике. На слици 3.7 приказана су четири градивна блока целокупне архитектуре. Основни блок (BB) састоји се од следећих слојева [18]: конволуција, нормализација, позитивна линеарна активација и још једна конволуција, након чега се излаз сабира са улазом у блок, како би се омогућило обучавање модела који имају значајно већи број слојева. Уколико је потребно да излаз из блока буде мање резолуције него улаз, уместо основог блока користиће се умањујући блок (BRB). Код овог блока се на основу вредности параметара померања, проширења и додатног конволуционог слоја умањује излазна резолуција. Као што је већ поменуто у потпоглављу 3.2, могуће је повећати ефикасност мреже раздвајањем слоја конволуције n×n на два слоја n×1 и 1×n, а основи блок са овако имплементираном конволуцијом назван је 1D блок (1DB). По истом принципу као и за основи блок уведен је 1D умањујући слој (1DRB).

Још једно унапређење конволуционог слоја односи се на увођење верзије конволуције која је предложена у оквиру *MobileNet* мреже [24], где се основна конволуција раздваја на

конволуцију по резолуцији и 1D конволуцију. Имплементација *MobileNet* конволуције на 1D основни (SPB) и умањујући (SRB) блок приказана је на слици 3.8.

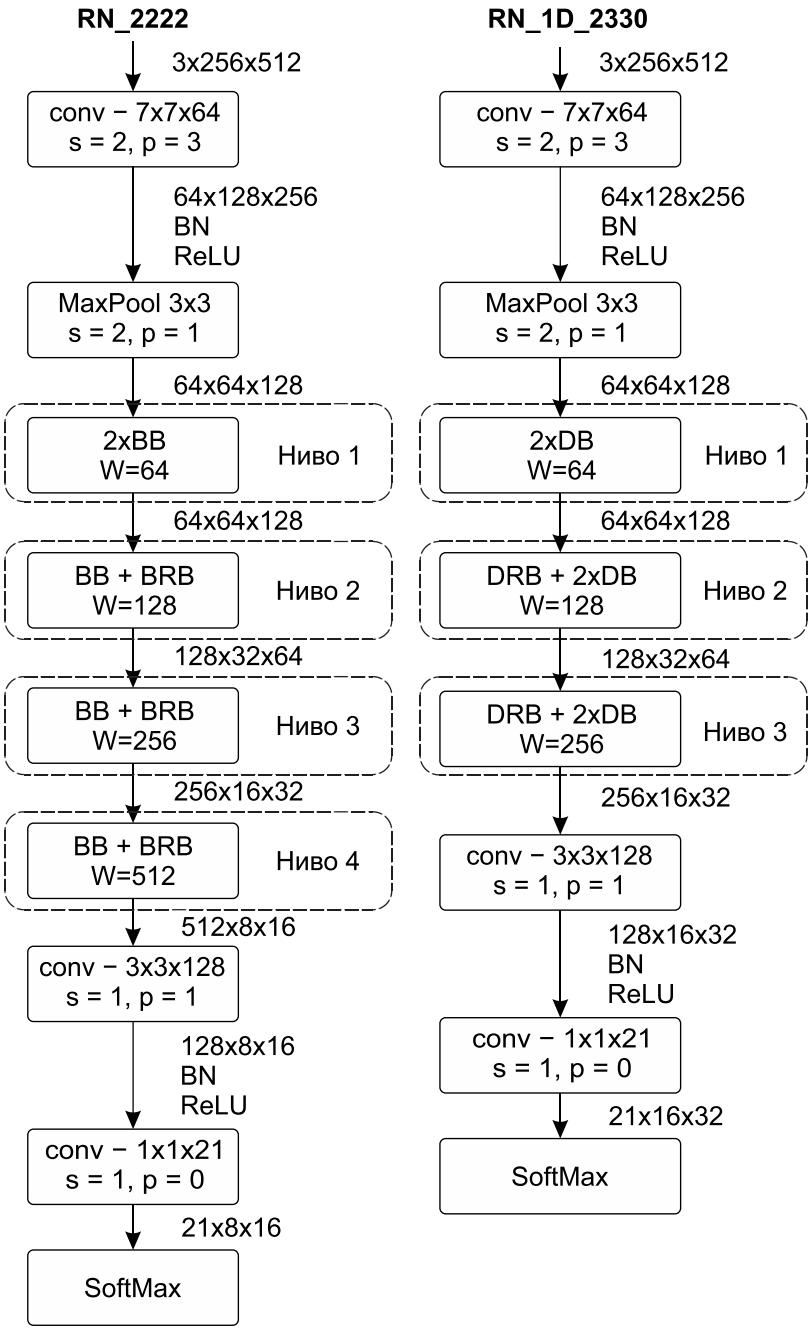


Слика 3.7. Четири градивна блока предложеног CNN модела.



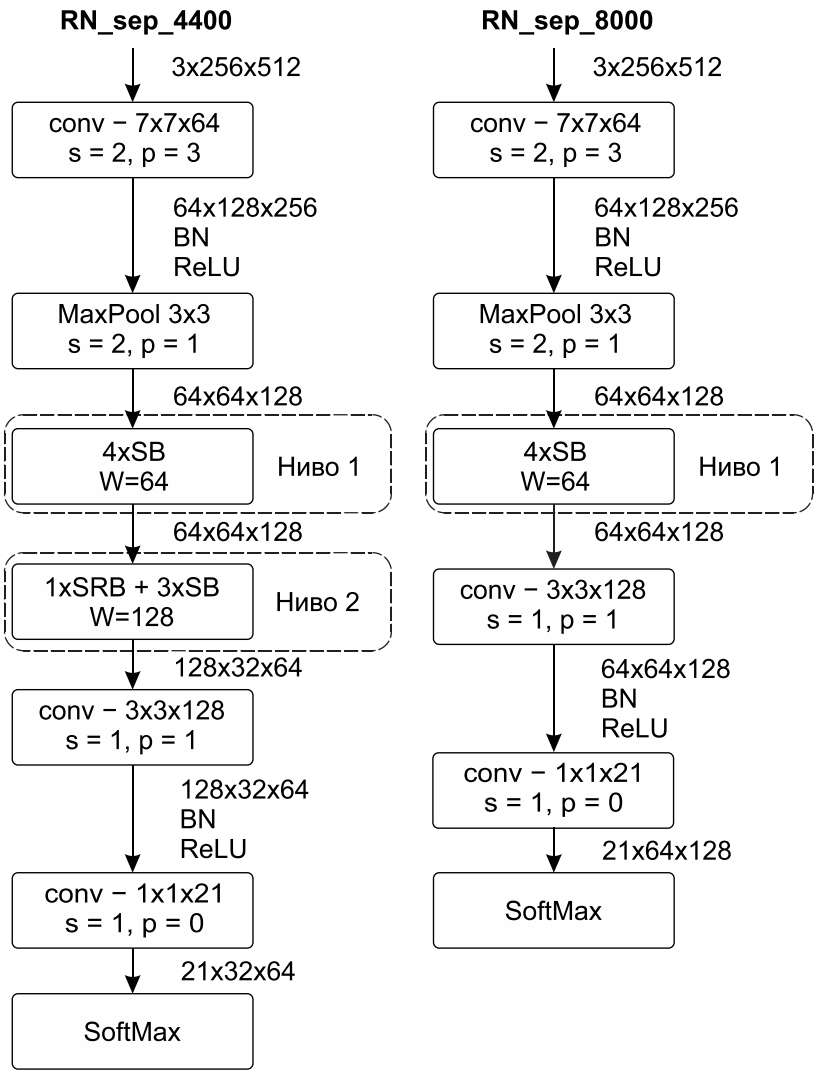
Слика 3.8. Два градивна блока предложеног CNN модела.

Архитектура CNN модела коришћена у оквиру система перцепције састоји се од комбинације поменутих шест блокова и то у оквиру четири нивоа. Она може имати различит број блокова и нивоа, док се резолуција излазне семантичке мапе смањује са сваким додатним нивоом [44]. На слици 3.9 приказане су две архитектуре, RN_2222 и RN_1D_2300, где је са RN представљена ознака *ResNet* мреже, а са „2222“ означено је да сваки од четири нивоа има по два основна или умањујућа блока. RN_1D_2330 представља модел који се састоји од три нивоа, у оквиру којих су имплементирани 1D или умањујући 1D блокови у три нивоа. За сваки ниво приказане су величине улазног и излазног тензора, као и индивидуални слојеви који нису део блокова, односно нивоа.



Слика 3.9. Приказ архитектура RN_2222 и RN_1D_2330 предложеног CNN модела.

Још две архитектуре са *MobileNet* конволуцијом, са два, односно једним нивоом, приказане су на слици 3.10. Као што се може уочити, резолуција излазне семантичке мапе свих архитектура варира од 8×16 до 64×128 пиксела.



Слика 3.10. Приказ архитектура RN_sep_4400 и RN_sep_8000 предложеног CNN модела.

Поређење тачности и ефикасности за 24 предложене архитектуре (табела 3.1) извршено је у оквиру експерименталне верификације, а за сваки тип конволуције имплементирано је осам модела [44]. Број блокова за сваку архитектуру је исти, али је њихов распоред по нивоима (као и број нивоа) различит. Обучавање је извршено на радној станици за дубоко машинско учење са три *NVidia Quadro RTX 6000* графичке картице и два *Xeon Silver 4208* процесора. Обучавање је извршено у оквиру библиотеке за дубоко машинско учење *PyTorch v1.6.0*. Сви модели су конвертовани у *TensorRT*, на основу *ONNX* формата у *FP16/INT8* прецизности, како би се убрзало време генерисања излаза на *Jetson Nano* рачунарској платформи. Како су мреже тестиране на реалном мобилном роботском систему, ефикасност је приказана у FPS.

CNN модел који је остварио највећу тачност, мерену у mIoU је RN_8000, док је модел RN_1D_8000 заузео најмању количину меморије, односно само 1,6 MB. Модел који је најефикаснији са аспекта броја FPS-а је RN_sep_1115. Међутим, као најбољи модел је узет RN_2600 где је остварен најбољи однос тачности (mIoU) и броја FPS-а. Након успешно извршеног процеса семантичке сегментације, а у циљу коришћења информација добијених са семантичке мапе у визуелном управљању, потребно је извршити процес регистрације семантичких мапа различитих слика генерисаних у истој сцени.

Табела 3.1. Резултати поређења CNN модела.							
Редни број	Назив архитектуре	Број блокова по нивоима	Излазна резолуција	Меморија модела [MB]	FPS	mIoU [%]	Тачност [%]
1.	RN_2222	[2 2 2 2]	8×16	46,0	44,7	28,674	77,816
2.	RN_1133	[1 1 3 3]	8×16	67,6	46,0	27,672	77,255
3.	RN_1115	[1 1 1 5]	8×16	95,3	46,8	27,060	76,541
4.	RN_2330	[2 3 3 0]	16×32	17,3	43,4	36,091	81,543
5.	RN_1160	[1 1 6 0]	16×32	28,5	44,7	34,508	81,089
6.	RN_4400	[4 4 0 0]	32×64	5,7	41,9	40,350	83,484
7.	RN_2600	[2 6 0 0]	32×64	7,5	42,6	40,668	83,651
8.	RN_8000	[8 0 0 0]	64×128	2,4	38,7	**40,804**	**84,589**
9.	RN_1D_2222	[2 2 2 2]	8×16	33,7	41,4	28,313	77,236
10.	RN_1D_1133	[1 1 3 3]	8×16	48,2	41,7	27,304	77,115
11.	RN_1D_1115	[1 1 1 5]	8×16	66,6	42,2	26,854	76,420
12.	RN_1D_2330	[2 3 3 0]	16×32	12,3	40,9	34,819	80,928
13.	RN_1D_1160	[1 1 6 0]	16×32	19,8	41,3	33,937	80,693
14.	RN_1D_4400	[4 4 0 0]	32×64	4,0	40,2	39,065	83,304
15.	RN_1D_2600	[2 6 0 0]	32×64	5,2	40,8	39,212	83,061
16.	RN_1D_8000	[8 0 0 0]	64×128	**1,6**	37,7	37,947	84,058
17.	RN_sep_2222	[2 2 2 2]	8×16	25,6	46,4	28,901	78,000
18.	RN_sep_1133	[1 1 3 3]	8×16	41,3	48,0	28,563	77,383
19.	RN_sep_1115	[1 1 1 5]	8×16	61,3	**48,7**	27,909	77,250
20.	RN_sep_2330	[2 3 3 0]	16×32	10,7	42,9	36,452	81,834
21.	RN_sep_1160	[1 1 6 0]	16×32	18,8	44,7	35,172	81,018
22.	RN_sep_4400	[4 4 0 0]	32×64	3,8	39,1	39,990	83,894
23.	RN_sep_2600	[2 6 0 0]	32×64	5,0	40,1	39,529	83,125
24.	RN_sep_8000	[8 0 0 0]	64×128	1,8	34,5	38,010	84,062

3.4.1. Регистрација слика

Регистрација слика представља процес трансформације слика генерисаних у истој сцени са два различита положаја, све док се те две слике не преклопе у највећој могућој мери. Преклапање се извршава применом процеса трансформације једне слике (покретна слика **I**), све док разлика између слика не постане минимална [54]. Функција на основу које се одређује разлика између слика назива се функција циља. Алгоритми за регистрацију могу се поделити у две категорије: алгоритми базирани на карактеристичним објектима и алгоритми базирани на интензитету пиксела. За алгоритме базиране на карактеристичним објектима, потребно је из обе слике издвојити па затим правилно упарити карактеристичне објекте, а након тога извршити трансформације док се слике (тј. карактеристични објекти) не поклопе. Алгоритми базирани на интензитету пиксела користе целокупну слику за процес регистрације и обезбеђују већу тачност. Такође, предност алгоритма за регистрацију на бази интензитета пиксела огледа се у томе што нема потребе за детекцијом и упаривањем карактеристичних објеката. Како се у оквиру ове докторске дисертације извршава регистрација семантичких мапа, користиће се алгоритми за регистрацију на основу интензитета пиксела.

Друга подела алгоритама за регистрацију слика је према моделу трансформације који се користи. Постоје два модела трансформације који се могу применити: (i) еластични или нелинеарни модел и (ii) линеарни или крути модел. Код еластичног модела трансформације, сликама су дозвољене еластичне деформације, чијом се применом може повећати флексибилност и број параметара потребних за процес регистрације. Међутим, како се у технолошком окружењу ретко транспортују деформабилни објекти (попут платна), одабрани су алгоритми који подразумевају круте трансформације. Алгоритми за регистрацију

дефинисани су као алгоритми оптимизације који минимизују функцију циља (E) између трансформисане (покретне $\mathbf{I}$) и фиксне слике ($\mathbf{I}^*$), а могу се дефинисати једначином (3.13):

$$\boldsymbol{\mu}_{opt} = \arg\max_{\boldsymbol{\mu}} \mathrm{E}\left(\mathbf{I}\left(\mathbf{T}\left(\boldsymbol{\mu}\right)\right), \mathbf{I}^*\right), \tag{3.13}$$

где је са $\mathbf{T}(\boldsymbol{\mu})$ означена трансформациона матрица дефинисана на основу вектора параметара $\boldsymbol{\mu} \in \mathbb{R}^5$, док је резултат оптимизације оптималан скуп параметара $\boldsymbol{\mu}_{\mathrm{opt}}$. Резултат алгоритама регистрације представља матрица трансформације ($\mathbf{T}$) која све пикселе дефинисане у хомогеним координатама покретне слике ($\mathbf{p}$) оптимално пресликава на пикселе фиксне слике ($\mathbf{p}^*$) (3.14):

$$\mathbf{T}\left(\boldsymbol{\mu}\right)\boldsymbol{p} \propto \boldsymbol{p}^*. \tag{3.14}$$

Крута матрица трансформације ($\mathbf{T}$ – једначина 3.15) примењена у оквиру ове докторске дисертације има пет параметара трансформације, транслацију по X оси (t_x), транслацију по Y оси (t_y), скалирање слике по X оси (s_x), скалирање слике по Y оси (s_y), као и ротацију око горњег левог угла слике (φ):

$$\mathbf{T}\left(\boldsymbol{\mu}\right) = \begin{bmatrix} \cos\left(\varphi\right)\cdot s_x & -\sin\left(\varphi\right) & t_x \\ \sin\left(\varphi\right) & \cos\left(\varphi\right)\cdot s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.15}$$

3.4.2. Дефинисање функција циља

Иако постоје различите функције циља које се користе при регистрацији слика, када се врши регистрација семантичких мапа, функције циља се морају адаптирати како би се примениле на класама објеката [54]. Семантичка мапа слике $\mathbf{I}$ означена је са $\hat{\mathbf{I}}$, док је класа пиксела $\boldsymbol{p}$ означена са $\hat{\boldsymbol{p}}$. Како би се дефинисале функције циља за семантичке мапе, потребне су две функције (3.16 и 3.17), које указују на то да ли два пиксела припадају истој или различитој класи:

$$b\left(\mathbf{T}\hat{\boldsymbol{p}}, \hat{\boldsymbol{p}}^*\right) = \begin{cases} 1, & \mathbf{T}\hat{\boldsymbol{p}} = \hat{\boldsymbol{p}}^* \\ 0, & \text{у супортном} \end{cases}, \tag{3.16}$$

$$c\left(\mathbf{T}\hat{\boldsymbol{p}}, \hat{\boldsymbol{p}}^*\right) = \begin{cases} 1, & \mathbf{T}\hat{\boldsymbol{p}} \neq \hat{\boldsymbol{p}}^* \\ 0, & \text{у супортном} \end{cases}. \tag{3.17}$$

Прва и најједноставнија функција циља је тачност поклапања (енгл. *Accuracy* – AC) (3.18):

$$AC = \frac{1}{N}\sum_{i=1}^{N}\left(b\left(\hat{\mathbf{I}}\left(\mathbf{T}\left(\boldsymbol{\mu}\right)\boldsymbol{p}_i\right), \hat{\mathbf{I}}^*\left(\boldsymbol{p}_i^*\right)\right)^2\right). \tag{3.18}$$

где N представља укупан број пиксела слике. Функција тачност поклапања ће имати вредности једнаку 1 када су пиксели две слике идентични. Како функција b може генерисати вредности 1 или 0, уводи се квадрат резултата ради једноставнијег процеса извођења градијента функције. Следећа функција, средњи количник пресека и уније (енгл. *mean Intersection over Union* – mIoU), инспирисана је радовима објављеним у домену семантичке сегментације и дефинисана једначином (3.19):

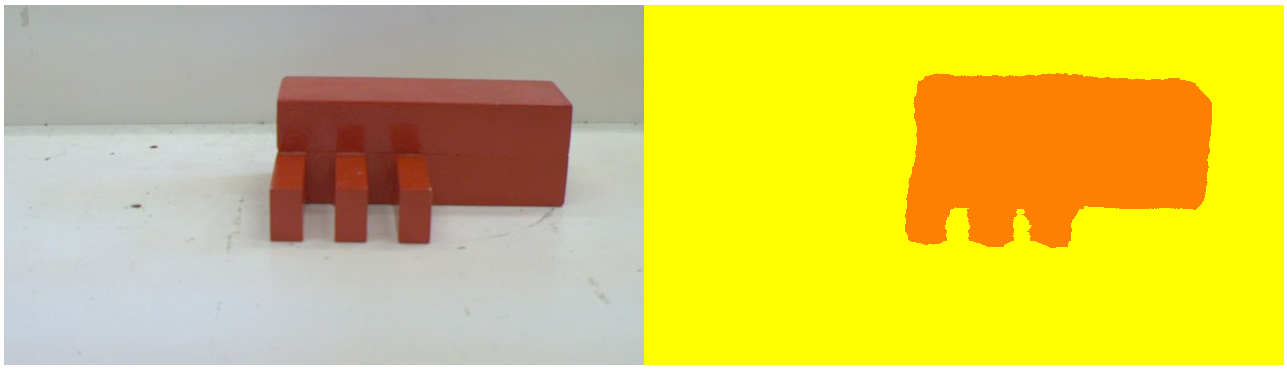
$$mIoU = \frac{1}{k}\sum_{d=1}^{k}\left(\frac{\sum_{i=1}^{N} b\left(\hat{\mathbf{I}}\left(\mathbf{T}(\boldsymbol{\mu})\boldsymbol{p}_i\right), \hat{\mathbf{I}}^*\left(\boldsymbol{p}_i^*\right)\right)}{\sum_{i=1}^{N} b\left(\hat{\mathbf{I}}\left(\mathbf{T}(\boldsymbol{\mu})\boldsymbol{p}_i\right), \hat{\mathbf{I}}^*\left(\boldsymbol{p}_i^*\right)\right) + \sum_{i=1}^{N} c\left(\hat{\mathbf{I}}\left(\mathbf{T}(\boldsymbol{\mu})\boldsymbol{p}_i\right), \hat{\mathbf{I}}^*\left(\boldsymbol{p}_i^*\right)\right) + \sum_{i=1}^{N} c\left(\hat{\mathbf{I}}^*\left(\boldsymbol{p}_i^*\right), \hat{\mathbf{I}}\left(\mathbf{T}(\boldsymbol{\mu})\boldsymbol{p}_i\right)\right)}\right), \quad (3.19)$$

где k представља укупан број класа, а d је индекс тренутне класе. Важно је нагласити да се функција mIoU рачуна за сваку класу понаособ, а затим се рачуна средња вредност за целу слику. Последња анализирана функција циља односи се на „заједничке информације“ (енгл. *Mutual Information* – MI). MI представља статистичку величину која мери зависност две случајно генерисане променљиве. Променљиве у овом случају представљају фиксну и слику која се помера, тј. сваки пиксел тих слика. Ова функција циља је најчешће коришћена у домену регистрације слика, а за семантичке мапе дефинисана је једначином (3.20):

$$MI = \sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j;\boldsymbol{\mu}) \log \frac{p(i,j;\boldsymbol{\mu})}{p_m(i;\boldsymbol{\mu})\, p_{m^*}(j)} \quad (3.20)$$

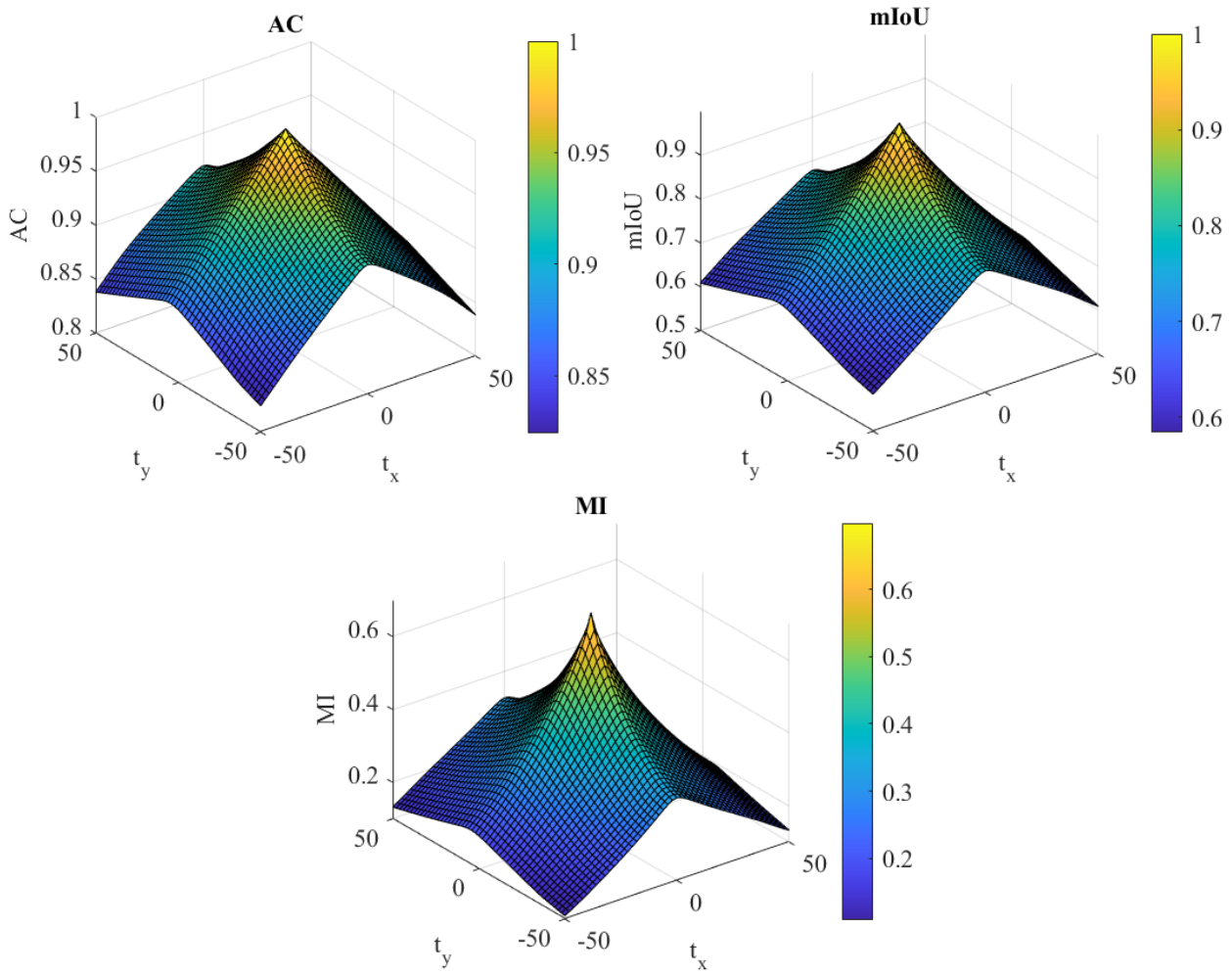
где $p(i,j;\ \boldsymbol{\mu})$ представља заједничку вероватноћу, док су p_m и p_{m^*} вероватноће независних променљивих. Додатна појашњења функције MI могу се пронаћи у литератури [55].

Поређење предложених функција циља биће приказано кроз експерименталну евалуацију зависности вредности функције циља од транслације слике у X и Y правцу. Слика која је коришћена у овој анализи, као и њена семантичка мапа, приказане су на слици 3.11.



Слика 3.11. Пример слике (лево) и њене семантичке мапе (десно).

Почетна семантичка мапа транслирана је инкрементално за два пиксела по једној оси и израчунате су вредности све три функције циља. Максимална вредност померања је 50 пиксела. Резултати су приказани површином креираном повезивањима вредности функција циља за све транслиране слике (слика 3.12). Као што се може уочити, све функције циља имају глобални максимум у позицији где семантичка мапа није трансформисана. Међутим, може се видети да функција тачности поклапања има приближно линеарни карактер, док се друге две функције мењају експоненцијално. Експоненцијални тренд функција mIoU и MI је погодан јер омогућава бржу конвергенцију ка жељеном решењу када се врши оптимизација при регистрацији.



Слика 3.12. Вредности функција циља за различите параметре транслације слике.

3.4.3. Оптимизациони алгоритми за регистрацију слика

У наставку биће анализирана три алгоритма коришћена за регистрацију слика [54]. Два алгоритма припадају групи метахеуристичких алгоритама оптимизације, а поређени су са стандардним градијентним поступком.

3.4.3.1. Градијентни поступак

Након дефинисања оптимизационог проблема и функција циља, потребно је приказати оптимизационе алгоритме потребне ради генерисања решења проблема регистрације слике. Прво је анализиран градијентни поступак (енгл. *Gradient Descent* – GDe), код кога се вредности параметара ажурирају на основу израза дефинисаног у једначини (3.21):

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \cdot \nabla E\left(\boldsymbol{\mu}_t\right) \tag{3.21}$$

где t представља индекс тренутне итерације процеса оптимизације, η је параметар учења, а $\nabla E(\boldsymbol{\mu}_t)$ представља градијент функције циља за тренутне параметре. Како би се извршио процес регистрације слике, потребно је трансформисати позиције пиксела из основног координатног система у хомогени координатни систем слике. Затим је потребно одредити градијент функције циља у односу на промену параметара $\boldsymbol{\mu}$ (једначина 3.22):

$$\nabla E(\boldsymbol{\mu}) = \frac{\partial E}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}). \tag{3.22}$$

Градијенти поступак биће изведен за функцију циља „тачност поклапања“ (3.18). Када се примени правило извода сложене функције, градијент је дефинисан једначином (3.23):

$$\frac{\partial E}{\partial \boldsymbol{\mu}} = -2\sum_{i=1}^{N} b\left(\hat{\mathbf{I}}\left(\mathbf{T}(\boldsymbol{\mu})\boldsymbol{p}_i\right), \hat{\mathbf{I}}^*\left(\boldsymbol{p}_i^*\right)\right)\frac{\partial \hat{\mathbf{I}}_i}{\partial \mathbf{T}\boldsymbol{p}_i}\frac{\partial \mathbf{T}\boldsymbol{p}_i}{\partial \boldsymbol{\mu}}, \tag{3.23}$$

где извод $\partial \mathbf{T}\boldsymbol{p}_i / \partial \boldsymbol{\mu}$ представља зависност промене трансформације пиксела од параметара трансформације, $\partial \hat{\mathbf{I}}_i / \partial \mathbf{T}\boldsymbol{p}_i$ је градијент семантичке мапе и $\sum_{i=1}^{N} b\left(\hat{\mathbf{I}}\left(\mathbf{T}(\boldsymbol{\mu})\boldsymbol{p}_i\right), \hat{\mathbf{I}}^*\left(\boldsymbol{p}_i^*\right)\right)$ представља тачност сваког пиксела тренутно регистрованих слика. Извод $\partial \hat{\mathbf{I}}_i / \partial \mathbf{T}\boldsymbol{p}_i$ генерише се као градијент по обе осе слике, добијен процесом конволуције и стандардним Собеловим филтером, док је извод $\partial \mathbf{T}\boldsymbol{p}_i / \partial \boldsymbol{\mu}$ по x и y координати пиксела дефинисан једначином (3.24):

$$\frac{\partial \mathbf{T}\mathbf{p}_i}{\partial \boldsymbol{\mu}} = \begin{pmatrix} -s_x x\sin(\varphi) - y\cos(\varphi) & x\cos(\varphi) & 1 & 1 & 0 \\ x\cos(\varphi) - s_y y\sin(\varphi) & 1 & y\cos(\varphi) & 0 & 1 \end{pmatrix} \tag{3.24}$$

где $\boldsymbol{\mu} \in \{\varphi, s_x, s_y, t_x, t_y\}$. Применом једначине (3.23), израчунава се вектор градијената 1×5 за сваки пиксел, који указује на то како утиче промена параметара на функцију циља и те вредности користе се у оквиру градијентног поступка. Псеудокод градијентног поступка примењеног за регистрацију слика приказан је у табели 3.2 [54]. Такође, током скалирања резолуције трансформисане слике, мора се применити алгоритам најближег суседа (енгл. *Nearest neighbour*), како пиксели не би добили вредности које не означавају постојеће класе.

Табела 3.2. Псеудокод градијентног поступка.

Улаз: иницијални параметри $\boldsymbol{\mu}$ = (0; 1; 1; 0; 0), семантичке мапе $\hat{\mathbf{I}}$, $\hat{\mathbf{I}}^*$, број итерација M = 400, параметар учења η = 0,1, параметар промене параметра учења η_d = 0,9, максималне промене параметара $\boldsymbol{m}$ = (0,01; 0,001; 0,001; 3; 3).

For (t = 1 **to** M)
- ∇E; (3.22) Израчунати градијенте
- $\boldsymbol{r} = -\eta \cdot \nabla E$;
- **if** $\boldsymbol{r} \geq \boldsymbol{m}$; Уколико је промена параметара већа од максималне
  - $\boldsymbol{r} = \boldsymbol{m}$; Подесити максималну вредност промене параметара
- **end**
- $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \boldsymbol{r}$; Ажурирати параметре трансформације
- $\hat{\mathbf{I}} = \hat{\mathbf{I}}(\boldsymbol{\mu}_t)$; Трансформација слике са новим параметрима
- $E_t(\hat{\mathbf{I}}, \hat{\mathbf{I}}^*)$; (3.18), (3.19) или (3.20) Израчунавање функције циља
- **if** ($E_t < E_{t-1}$) Уколико је вредност функције циља лошија од претходне
  - $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1}$; Подесити вредности претходних параметара
  - $E_t = E_{t-1}$; Подесити вредност претходне функције циља
  - $\eta = \eta \cdot \eta_d$; Смањити параметар учења
- **else** % Уколико је побољшана вредност функције циља
  - $\eta = \eta / \eta_d$; Повећати параметар учења
- **end**

end

3.4.3.2. Алгоритам симулираног жарења

Алгоритам симулираног жарења (енгл. *Simulated Annealing* – SA) представља један од најчешће коришћених метахеуристичких оптимизационих алгоритама. Овај алгоритам припада групи алгоритама базираних на једном решењу и алгоритама инспирисаних законима физике. Алгоритам симултаног жарења [56] настао је на основу аналогије са процесом жарења метала, које подразумева да се метал прво загреје на високу температуру, а затим се врши његово контролисано хлађење. Предност овог алгоритма у односу на градијентни поступак је у могућности избегавања локалних минимума, због карактеристике које му омогућава да се са одређеном вероватноћом (пропорционалној температури) одабере и лошије решење током процеса оптимизације. Псеудокод алгоритма симулираног жарења дат је у табели 3.3 [54].

Табела 3.3. Псеудокод алгоритма симулираног жарења.
Улаз: иницијални параметри $\boldsymbol{\mu}$ = (0; 1; 1; 0; 0), семантичке мапе $\hat{\mathbf{I}}$, $\hat{\mathbf{I}}^*$, број итерација M = 400, иницијална температура T_0 = 200, најнижа температура T_l = 0,01, максималне промене параметара $\boldsymbol{m}$ = (0,1; 0,01; 0,01; 1; 1). cf = (T_0–T_l)/(M–1); Одредити вредност параметра хлађења $\hat{\mathbf{I}} = \hat{\mathbf{I}}(\boldsymbol{\mu}_t)$; Трансформација слике са иницијалним параметрима E_t ($\hat{\mathbf{I}}$, $\hat{\mathbf{I}}^*$); (3.18), (3.19) или (3.20) Израчунавање функције циља **For** (t = 1 **to** M) T_c = (T_0 + cf) – t · cf; Одређивање тренутне температуре $\boldsymbol{r}$ = rand() · $\boldsymbol{m}$; Генерисање вектора случајних бројева у опсегу [-1, 1] који подлеже униформној расподели; $\boldsymbol{\mu}_{temp} = \boldsymbol{\mu}_t + \boldsymbol{r}$; Ажурирати тренутне параметре трансформације $\hat{\mathbf{I}} = \hat{\mathbf{I}}(\boldsymbol{\mu}_{temp})$; Трансформација слике са тренутним параметрима E_t ($\hat{\mathbf{I}}$, $\hat{\mathbf{I}}^*$); (3.18), (3.19) или (3.20) Израчунавање функције циља **if** ($E_{temp} < E_t$) Уколико је вредност функције циља лошија од претходне $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t$; Подесити вредности претходних параметара $E_{t+1} = E_t$; Подесити вредност претходне функције циља r_1= rand(); Генерисање случајног броја у опсегу [0, 1] који подлеже униформној расподели e = 1/(1+e $^{(E_t - E_{temp})/T_c}$); Одређивање вредности вероватноће **if** e > r_1 Уколико се усваја лошије решење $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_{temp}$; Усвајање тренутног лошијег решења $E_{t+1} = E_{temp}$; Усвајање тренутне лошије вредности функције циља **end** **else** Уколико је побољшана вредност функције циља $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_{temp}$; Ажурирати параметре трансформације $E_{t+1} = E_{temp}$; Ажурирати функцију циља **end** **end**

3.4.3.3. Еволуциони алгоритам

Еволуциони алгоритам (енгл. *OnePlusOne Evolutionary algorithm* – EV) представља оптимизациони алгоритам који је инспирисан процесом еволуције. Основни вид ажурирања параметара оптимизације овим алгоритмом заснива се на генерисању случајних бројева који подлежу Гаусовој расподели са очекиваном вредношћу нула и стандардном девијацијом један. Како постоји више различитих модификација еволуционих алгоритама, у оквиру предметне докторске дисертације коришћен је алгоритам предложен у [57]. Главна карактеристика овог алгоритма је у коришћењу матрице коваријансе $\mathbf{A_t}$ ради ажурирања индивидуалних вредности у вектору параметара оптимизације $\boldsymbol{\mu}$. Псеудокод предложеног алгоритма дат је у табели 3.4 [54].

Табела 3.4. Псеудокод еволуционог алгоритма.

Улаз: иницијални параметри $\boldsymbol{\mu}$ = (0; 1; 1; 0; 0), семантичке мапе $\hat{\mathbf{I}}$, $\hat{\mathbf{I}}^*$, број итерација M = 400, параметар повећања c_g = 1,05, иницијална варијанса $A_r = 6{,}25\cdot10^{-3}$, минимална промена параметара $\varepsilon=1{,}5\cdot10^{-6}$, вектор стандардних девијација параметара оптимизације $\boldsymbol{ms}$ = (0,1; 0,01; 0,01; 1; 1).
$c_s = c_g^{-0.25}$; Параметар смањења
$\hat{\mathbf{I}} = \hat{\mathbf{I}}(\boldsymbol{\mu}_t)$; Трансформација слике са иницијалним параметрима
E_t ($\hat{\mathbf{I}}$, $\hat{\mathbf{I}}^*$); (3.18), (3.19) или (3.20) Израчунавање функције циља
$\mathbf{A_t} = \mathbf{I_n} \cdot A_r$; $\mathbf{I_n}$ представља јединичну матрицу
For (t = 2 **to** M)
 $\boldsymbol{r}$ = randn() · $\boldsymbol{ms}$; randn() – генерисање вектора случајних бројева са очекиваном вредношћу 0 и стандардном девијацијом 1 који подлежу нормалној расподели;
 if $\mathbf{A_t} \le \varepsilon$ Уколико је промена параметара мања од минималне
  $\mathbf{A_t} = \varepsilon$; Подесити минималну вредност промене параметара
 end
 $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \boldsymbol{r} \cdot \mathbf{A_t}$; Ажурирати параметре трансформације
 $\hat{\mathbf{I}} = \hat{\mathbf{I}}(\boldsymbol{\mu}_t)$; Трансформација слике са тренутним параметрима
 E_t ($\hat{\mathbf{I}}$, $\hat{\mathbf{I}}^*$); (3.18), (3.19) или (3.20) Израчунавање функције циља
 if ($E_t < E_{t-1}$) Уколико је вредност функције циља лошија од претходне
  $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1}$; Подесити вредности претходних параметара
  $E_t = E_{t-1}$; Подесити вредност претходне функције циља
  $\mathbf{A_t} = \mathbf{I_n} + (c_s - 1)\cdot(r^T\cdot r)./(r\cdot r^T)$; Смањење вредности $\mathbf{A_t}$,
   „./” представља директно дељење елемената две матрице.
 else Уколико је побољшана вредност функције циља
  $\mathbf{A_t} = \mathbf{I_n} + (c_g - 1)\cdot(r^T\cdot r)./(r\cdot r^T)$; Повећање вредности матрице коваријанси $\mathbf{A_t}$
 end
end

Ради побољшања тачности регистрације, за сваки од три алгоритма уведена је методологија регистрације сликама мање резолуције. Наиме, прво се дефинише колико ће нивоа бити коришћено, а затим се резолуција слика које се региструју умањи 2 пута за сваки ниво умањења. У оквиру предметне докторске дисертације, коришћена су два нивоа умањења, тако да је код првог нивоа резолуција слике била смањена два пута, а код другог четири пута. Разлог смањења резолуције је да се пронађе глобално тачно решење на слици мање резолуције, а да се затим добијени параметри трансформације користе код нижих нивоа (где слике имају већу резолуцију) као почетна претпоставка. Ова методологија обезбеђује побољшана својства конвергенције алгоритама, због боље иницијалне претпоставке решења добијене на основу регистрације слика мање резолуције, а истовремено смањује вероватноћу „заробљавања“ алгоритама у локалним минимумима.

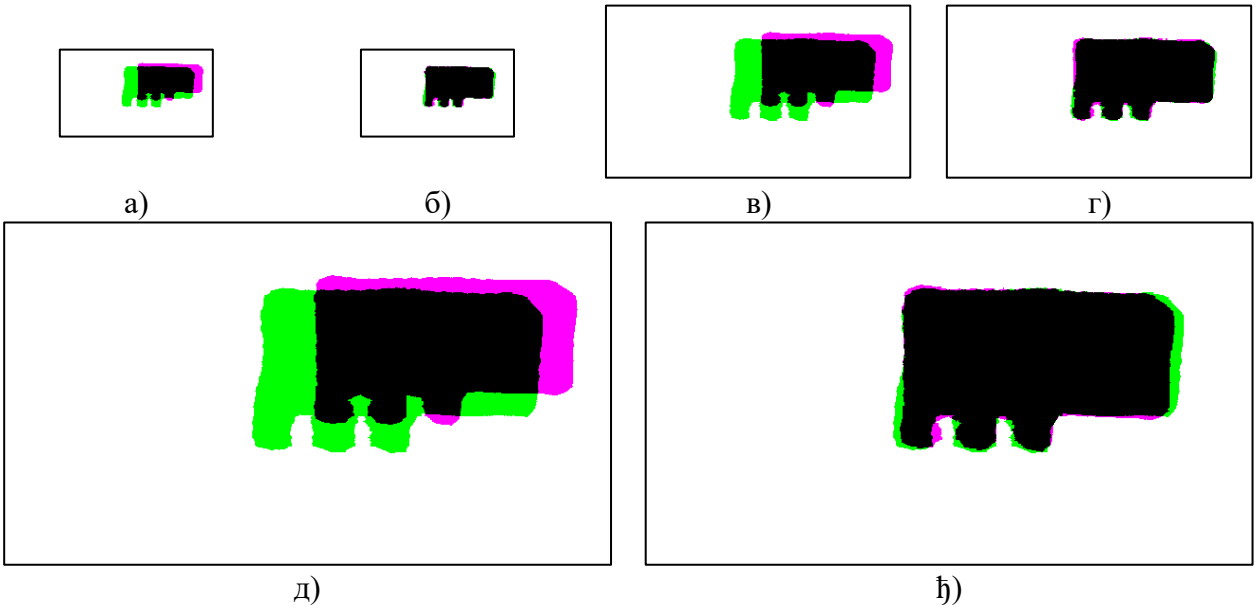
3.4.3.4. Упоредна анализа оптимизационих алгоритама за регистрацију слика

Извршено је експериментално поређење предложених оптимизационих алгоритама, као и њихових комбинација са функцијама циља. Мобилни робот постављен је у почетни положај и генерисане су слике са обе камере у стерео пару, након чега је мобилни робот позициониран у 12 различитих положаја, где су забележене и сачуване слике. Тачни положаји робота дати су у табели 3.5.

Табела 3.5. Положаји мобилног робота.

Р.б.	Положај (x [cm], y [cm], θ [°])	Р.б.	Положај (x [cm], y [cm], θ [°])
1.	(2, 0, 0)	7.	(0, 2, 0)
2.	(-2, 0, 0)	8.	(0, 4, 0)
3.	(-4, 0, 0)	9.	(-2, 2, 0)
4.	(-6, 0, 0)	10.	(-4, 4, 0)
5.	(0, -4, 0)	11.	(0, 0, 5)
6.	(0, -2, 0)	12.	(0, 0, -5)

Извршен процес регистрације семантичких мапа генерисаних у свим положајима са семантичким мапама генерисаном у почетном положају. Примери поклапања семантичких мапа на сва три нивоа умањења резолуције дати су на слици 3.13. На сликама 3.13а), в) и д) представљено је поклапање семантичких мапа пре почетка регистрације. Зеленом бојом су приказани пиксели покретне слике који се не поклапају, тј. они код којих су другачије класе на обе семантичке мапе. Розе бојом означени су пиксели фиксне слике који нису добро регистровани, док су белом и црном бојом приказани добро регистровани пиксели.



Слика 3.13. Тачност поклапања пре и после регистрације. Слике а) и б) представљају поклапање семантичких мапа пре и после регистрације за ниво 2, респективно, док су в) и г) слике за ниво 1, а слике д) и ђ) су за ниво 0.

Параметри подешени за сваки алгоритам дати су у оквиру табела 3.2, 3.3 и 3.4. У оквиру табеле 3.6, дати су резултати оптимизације за сва три оптимизациона алгоритма при коришћењу све три функције циља. Као што се може уочити, за све три функције циља, алгоритми GDe и EV показали су значајно боље резултате него алгоритам SA. Финална селекција најбоље комбинације функције циља и алгоритма оптимизације извршена је на основу резултата првог експеримента где је функција MI показала најбоље карактеристике, а када су поређени алгоритми оптимизације на овој функције као најбољи показао се EV па је та комбинација имплементирана у оквиру даљег развоја система за стерео визуелно управљање.

Табела 3.6. Резултати поређења оптимизационих алгоритама и функција циља; GDe – градијентни поступак, SA – симулирано жарење, EV – еволуциони алгоритам.									
Р.б.	AC (3.18)			mIoU (3.19)			MI (3.20)		
	GDe	SA	EV	GDe	SA	EV	GDe	SA	EV
1	**0.9950**	0.9780	**0.9950**	0.9830	0.8990	**0.9840**	**0.6510**	0.5600	**0.6510**
2	0.9940	0.9790	**0.9950**	**0.9800**	0.9150	**0.9800**	0.6470	0.5770	**0.6570**
3	**0.9940**	0.9680	0.9930	0.9760	0.9030	**0.9820**	0.6420	0.4800	**0.6500**
4	**0.9920**	0.9270	0.9910	**0.9730**	0.8150	0.9680	0.6100	0.4180	**0.6300**
5	**0.9900**	0.9770	**0.9900**	0.9660	0.9530	**0.9720**	0.6180	0.6020	**0.6200**
6	0.9930	0.9830	**0.9940**	**0.9790**	0.9440	0.9770	0.6440	0.5810	**0.6460**
7	**0.9940**	0.9900	**0.9940**	**0.9820**	0.9320	0.9800	**0.6520**	0.6340	0.6410
8	**0.9900**	**0.9900**	0.9840	**0.6460**	0.6340	0.6390	**0.6270**	0.6020	0.5970
9	0.9920	0.9810	**0.9940**	0.9750	0.9220	**0.9790**	0.6330	0.4730	**0.6390**
10	**0.9890**	0.9700	0.9840	**0.9680**	0.8850	0.9500	0.5270	0.4860	**0.5800**
11	**0.9870**	0.9790	0.9760	**0.9500**	0.9400	0.9230	**0.6040**	0.4660	0.5380
12	**0.9880**	0.9830	0.9860	**0.9620**	0.9530	0.9330	**0.6040**	0.5790	0.5890

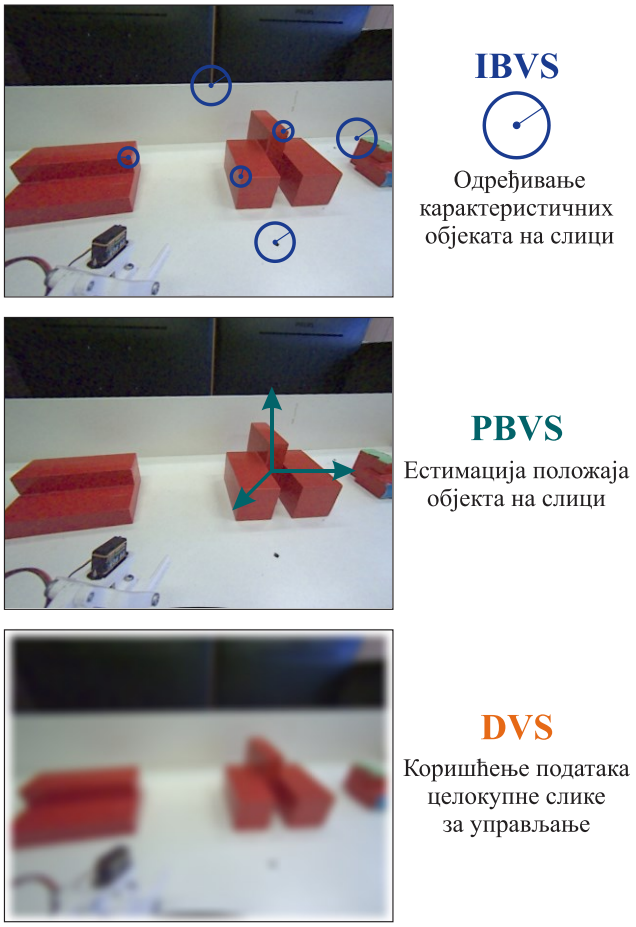
Као што се може уочити из табеле 3.6, за метрике AC и mIoU алгоритам GDe остварио је најбоље резултате. За метрику MI алгоритам EV има највише најбољих резултата. Међутим, како је на основу претходне анализе функција циља установљено да је метрика MI најпогоднија за примену у оквиру процеса регистрације семантичких мапа, за даљи рад коришћен је EV алгоритам интегрисан са метриком MI.

4. Визуелно управљање мобилних роботских система

Методе визуелног управљања представљају скуп алгоритама код којих се кретање динамичких система (обично роботских) генерише искључиво на основу информација добијених од камере. Основна подела метода визуелног управљања [58] је на (слика 4.1):

- Управљање на бази грешака у параметрима слике (енгл. *Image-Based Visual Servoing – IBVS*) [59],
- Управљање на бази издвојених карактеристика са слике неопходних за естимацију положаја изабраног објекта (енгл. *Position-Based Visual Servoing – PBVS*) [59],
- Хибридне системе управљања (енгл. *Hybrid Visual Servoing – HVS)* [60],
- Системе директног визуелног управљања (енгл. *Direct Visual Servoing – DVS*) [61].

Код IBVS алгоритама, управљање се врши на основу грешке дефинисане параметрима са слике који могу представљати тачке, контуре, линије итд. Овај тип визуелног управљања се карактерише великом робустношћу на грешке при калибрацији камере. У оквиру PBVS алгоритма, користе се визуелне информације како би се одредио положај објекта на слици, а те информације директно дефинишу систем управљања. Предност PBVS-а у односу на IBVS огледа се у томе да је грешка дефинисана у 3D простору, што директно утиче на генерисање повољнијих трајекторија при остваривању жељеног положаја. HVS алгоритми интегришу IBVS или PBVS са другим алгоритмима управљања, а њиховом хибридизацијом остварују се системи управљања бољих перформанси. На крају, DVS методе користе целокупне слике, тј. све податке са слике и на основу њих се дефинише управљачки алгоритам. Предност DVS метода огледа се у високој тачности у односу на све остале методе визуелног управљања.



Слика 4.1. Подаци који се користе у оквиру различитих метода визуелног управљања.

4.1. Преглед стања у области визуелног управљања

Преглед стања у области визуелног управљања подељен је у три целине. У потпоглављу 4.4.1 анализирани су радови везани за директно визуелно управљање. Међутим, како већину DVS метода није могуће имплементирати на мобилне роботе, приказане су методе визуелног управљања које се примењују за управљање нехолономних мобилних робота (потпоглавље 4.4.2), а који се крећу помоћу точкова. На крају (потпоглавље 4.4.3) биће дате методе визуелног управљања код којих је извршена интеграција алгоритама машинског учења у процес управљања роботских система.

4.1.1. Директно визуелно управљање

У оквиру табеле 4.1 може се уочити следствени развој метода директног визуелног управљања на основу информација о години објављивања, о подацима са слике који се користе при управљању и врсти роботског система примењеног у сврху тестирања.

Табела 4.1. Методе директног визуелног управљања.

Референца	**Година**	**Подаци са слике**	**Роботски систем**
[62]	1996.	Сопствене вредности	Индустријски робот
[63]	2007.	Кернели	Индустријски робот са 6 степени слободе
[64],[61],[65]	2008, 2010, 2011.	Интензитет осветљености пиксела	Индустријски робот са 6 степени слободе
[55]	2011.	Заједничке информације	Индустријски робот са 6 степени слободе
[66]	2012.	Суме условних варијанси	Индустријски робот
[67]	2012.	Регистрација слике	Мобилни робот
[68]	2013.	Интензитет осветљености пиксела за омнидирекционе камере	Мобилни и индустријски робот са 6 степени слободе
[69]	2013.	Фотометријски моменти	Индустријски робот са 6 степени слободе
[70]	2014.	3D слика	Мобилни и индустријски робот са 6 степени слободе
[71]	2015.	Гаусови мешовити модели	Индустријски робот са 6 степени слободе
[72], [73]	2017, 2015.	Хистограми слике	Мобилни и индустријски робот са 6 степени слободе
[74]	2017.	Честични филтер и интензитет осветљености пиксела	Индустријски робот са 6 степени слободе
[75]	2019.	Гаусови мешовити модели	Индустријски робот са 6 степени слободе
[76]	2019.	Вејвлет и ширлет коефицијенти	Индустријски робот са 6 степени слободе
[77]	2020.	Дискретни косинусни коефицијенти	Индустријски робот са 6 степени слободе
[78]	2023.	Фреквентна трансформација 3D слика	Индустријски робот са 7 степени слободе
[79]	2023.	Ортогонални моменти	Индустријски робота са 6 степени слободе

Истраживање спроведено у раду [62] једно је од првих које се односи за област директног визуелног управљања. У оквиру предложеног алгоритма управљања, коришћени су интензитети пиксела целе слике ради визуелног управљања. Аутори разликују предложену врсту визуелног управљања од IBVS-а и PBVS-а тако што је за управљање коришћен изглед сцене (цела слика), а не одређене геометријске карактеристике слике. Предложени алгоритам управљања има предност у погледу робустности на промену рефлексије, текстуре, као и осталих својстава површина. Такође, робустност се осигурава у процесу обучавања који је потребно извршити пре процеса управљања. Коришћен је робот са пет степени слободе, који на себи има постављену камеру и извор светлости. Примарна идеја анализираног рада је да се на основу довољног броја слика генерисаних у окружењу са познатим положајем камере може обезбедити кретање робота од старта до циља. Како је претходни задатак немогуће било извршити са целим сликама због тадашњих ограничења везаних за меморијске и прорачунске потребе, аутори су смањили димензије проблема прешавши у сопствени простор (енгл. *Eigenspace*), који је био представљен са 20 сопствених вредности (енгл. *Eigenvalues*). Пре преласка у сопствени простор, потребно је издвојити одређени сегмент слике који обухвата жељени објекат. На поменутим сегментима врши се нормализација и скалирање у циљу остваривања робустности на промену положаја по удаљености. Када су одређени сопствени простор и сопствени вектори свих слика, може се одредити где се робот тренутно налази и израчунати тренутни сопствени вектор. Затим, врши се визуелно управљање до жељеног положаја. Развијени систем аутори не предлажу само за управљање, већ и за визуелну инспекцију. Као и код осталих метода директног визуелног управљања, управљачки систем формулисан је као оптимизациони проблем. На крају, поред стандардних алгоритама за оптимизацију, предложен је нови начин оптимизације базиран на првом изводу Јакобијан матрице, за који је доказано да брже конвергира од стандардних метода.

У раду [63] анализира се директно визуелно управљање на основу кернела (енгл. *Kernel*). Понаособ су анализиране три врсте кретања: транслације у равни слике, ротација око оптичке осе и транслација у правцу оптичке осе. Потребне транслације у равни слике рачунају се коришћењем кернела, док се транслација у правцу оптичке осе и ротација око оптичке осе генеришу на основу интензитета амплитуде Фуријеове трансформације две узастопно генерисане слике. У ранијим радовима из ове области доказано је да интензитет поменуте амплитуде не зависи од транслација у равни камере, већ да зависи само од удаљености и ротација. За сва кретања понаособ доказана је стабилност процеса визуелног управљања. Предност ове методе управљања је у интеграцији процеса праћења карактеристичних објеката у сам систем управљања.

Како би се избегло комплексно генерисање и упаривање карактеристичних објеката на сликама, у референцама [61], [64] и [65] представљене су методе које експлоатишу интензитет пиксела за управљање роботским системима. Како би предложени систем било могуће имплементирати, потребно је моделирати начин осветљења сцене. Доказане су могућности примене једноставног дифузног модела осветљења, као и комплексних модела који подразумевају да се извор светлости налази иза камере. Са аспекта грешке положаја робота, ови системи имају виши ниво тачности у поређењу са системима визуелног управљања базираним на карактеристичним објектима. Чињеница да се користи велики број пиксела за генерисање управљачких величина омогућава предложеном систему управљања да буде робустан на промене у сцени. У експерименту је приказано да додавање нових предмета, као и промена њиховог положаја унутар сцене у току управљања, не утиче значајно на конвергенцију грешке, а финална грешка положаја роботског система (која настаје услед ових поремећаја) је у дозвољеним границама. Први недостатак предложеног система огледа се у осетљивости на промене у осветљењу које се дешавају при управљању у реалним индустријским условима. Такође, други недостатак предложеног система је узак опсег конвергенције. Уколико је разлика почетног и жељеног положаја (у експериментима је у почетном тренутку грешка позиције износила ~ 24 cm, а оријентације ~ 20°) већа од домена

конвергенције, није могуће извршити управљање применом овог система. Методе директног визуелног управљања најбоље су се показале у проблемима који захтевају висок ниво тачности, а подразумевају мала померања робота или за интеграцију са неком другом методом визуелног управљања.

Радови [55] и [66] су наставак истраживања предложених у раду [65], тако да ће ова два рада бити упоредо анализирана у наставку. Метода директног визуелног управљања на бази заједничких информација (енгл. *Mutual information* – MI) између две слике представљена је у референци [55]. Визуелно управљање на бази параметара MI је робустно на промене осветљења и заклањање дела сцене. Предложени модел показује најбоље резултате од две тренутно поређене методе директног визуелног управљања. Такође, доказано је да има већи домен конвергенције, као и задовољавајућу тачност оствареног положаја. Повећани домен заузима запремину од приближно 1 m^3 са великим иницијалним ротацијама (до 60°) око X и Y осе. Поред експеримената и симулација, у раду [55] извршено је поређење са стандардном IBVS методом и алгоритмом развијеним у [65], а доказано је да је модел базиран на заједничким информацијама има већи домен конвергенције од алгоритма фотометријског визуелног управљања [65] и већу тачност оствареног положаја од IBVS-а. Такође, предложен је и хибридни систем у оквиру референце [55], добијен интеграцијом IBVS-а и алгоритма DVS на бази заједничких информација. Код хибридног система, IBVS се користи за управљање док се камера не нађе унутар домена конвергенције који има систем базиран на заједничким информацијама, након чега се активира алгоритам DVS на бази заједничких информација [55]. У референци [66] анализира се директно визуелно управљање базирано на суми условних варијанси (енгл. *Sum of conditional variance*). Као и код већине управљачких система директног визуелног управљања, управљачки систем дефинисан је као оптимизациони процес, при чему се за решавање овог проблема користи Левенберг-Маркеов алгоритам [7]. Представљени метод показао се бољим од метода из референце [65] у погледу робустности на промене у осветљењу. Такође, предност алгоритма [66] у односу на метод предложен у [55] је у томе што су потребни далеко мањи процесорски ресурси за управљање у реалном времену, иако систем из [55] има боље перформансе са аспекта тачности.

У оквиру рада [67] предложен је систем директног визуелног управљања где се на основу процеса регистрације слика врши визуелно управљање. Регистрацијом на основу интензитета осветљености пиксела могуће је дефинисати математичку зависност позиције пиксела на тренутној и циљној слици, а на основу перспективне трансформације и епиполова повезује се тренутни и циљни положај робота. Систем визуелног управљања реализује се директном применом параметара перспективне трансформације. Недостатак предложеног система је у потреби за великим процесорским ресурсима потребним за регистрацију слика у реалном времену. Адаптација алгоритма визуелног управљања предложеног у [64] за омнидирекционе камере извршена је у оквиру референце [68]. Алгоритам директног визуелног управљања имплементиран је на нехолономни мобилни роботски систем. Недостатак предложеног система огледа се у имплементацији на омнидирекционе камере које нису често заступљене. Због претходно наведених недостатака везаних за домен конвергенције метода директног визуелног управљања, развијен је систем управљања [69] код кога се користе фотометријски моменти, добијени на основу интензитета пиксела, за оцењивање разлике почетног и жељеног положаја робота. Коришћењем ове врсте момената, Јакобијан матрица има декупловану форму, која раздваја компоненте транслаторне и угаоне брзине. За управљачки систем у овом раду није потребно вршити комплексну обраду слике, већ само релативно једноставно рачунање самих момената. Такође, потребно је напоменути да се у овом раду разматра управљање само када сцена представља раван и нема значајних разлика у удаљености од камере. Извршена су два експеримента, са роботима који имају четири и шест степени слободе. Експеримент са роботом који поседује четири степена слободе кретања извршен је са тачношћу од неколико милиметара, али и са комплексном путањом коју робот остварује. Други експеримент, са роботом који има шест степени слободе, био је успешнији са аспекта тачности

оствареног положаја, док се једини недостатак огледао у неадекватној путањи којом је робот остварио жељени положај. При овом експерименту извршено је поређење са фотометријским директним визуелним управљањем [65].

У оквиру рада [70] предложен је систем визуелног управљања на основу 3D слике. Систем визуелног управљања је формулисан тако да минимизује разлику између дистанце пиксела тренутне и жељене слике, а не интензитета осветљености пиксела као код других метода директног визуелног управљања. Шум сензора је минимизован применом Гаусовог филтера, а пиксели за које сензор није генерисао удаљеност нису коришћени у управљачком систему. Такође, коришћен је М-естиматор ради одређивања делова слике за које постоји вероватноћа да имају висок ниво шума (нпр. при заклањању дела сцене). Експерименти су извршени са индустријским и мобилним роботом, чиме је доказана валидност предложеног система.

Како би се побољшао систем директног визуелног управљања, аутори рада [71] предлажу имплементацију Гаусових мешовитих модела (енгл. *Gaussian Mixture Models* – GMM). Ови модели омогућавају да се интензитети пиксела моделирају Гаусовом расподелом, чиме се повећава домен конвергенције у односу на DVS алгоритам предложен у референци [61]. Експерименталном верификацијом утврђено је да се домен конвергенције знатно повећао у односу на фотометријско визуелно управљање [61]. За разлику од приступа датих у радовима [61] и [69], у раду [71] доказано је да није потребно рачунање градијената слике, које је процесорски захтевно. Недостатак предложног система је потреба за значајним компјутерским ресурсима ради брзог израчунавања GMM-а. Иако се применом GMM-а (који се генеришу на деловима слике који износе 80×64) смањи количина података коришћених у управљачком алгоритму (у поређењу са коришћењем целе слике), управљање у реалном времену није могуће јер је за израчунавање GMM-а потребно око 1,2 s. Додатни недостатак предложеног система је потреба за естимацијом параметра проширења (енгл. *Expansion parameter*), чију вредност је потребно подесити за сваки експеримент понаособ. Извршено је неколико симулација и два реална експеримента. Слика циља је у сваком експерименту била слика раванске сцене, а удаљеност од камере до равни била је претпостављена. Резултати експеримената показују да се применом ове методологије остварује висока тачности позиционирања (0,129 mm; 1,01 mm; 4,28 mm; 0,60°; 0,77°; 0,35°), при чему је разлика тренутног и жељеног положаја на почетку експеримента износила (8,22 cm; 5,75 cm; 5,1 cm; 35,17°; 8,50°; 5,75°). Такође, још један значајан недостатак овог система је генерисање неправилног облика путање којом се робот креће при остваривању жељеног положаја.

Визуелно управљање мобилним и индустријским роботом применом различитих хистограма је предмет анализе рада [72]. Предложена су три управљачка система, базирана на следећим хистограмима: хистограм оријентације градијената (енгл. *Histogram of Oriented Gradients* – HOG), *Hue-Saturation* хистограм, и комбиновани *Hue-Saturation* хистограм. Остварени управљачки системи поређени су са управљачким системом који је базиран на хистограму интензитета пиксела [73]. Вршена је конвексна оптимизација коришћењем Левенберг-Маркеовог алгоритма. Како се на основу једног хистограма може управљати само једним степеном слободе робота, у овом раду слика је подељена у сегменте и коришћени су хистограми различитих сегмената. Разлог примене нових хистограма (у односу на рад [73]) је креирање управљачког система који је робустан на промене осветљења. Након извршене симулације, добијени резултати указују да се управљачки систем базиран на HOG-у најбоље понаша када постоје велике промене осветљења. Експеримент са мобилним роботом извршен је само за управљачки систем базиран на HOG-у и резултати су показали да је овај систем робустан на промене осветљења у реалним условима рада. У референци [74] примењен је секвенцијални Монте Карло метод (честични филтер енгл. *Particle filter*) за проширење домена конвергенције метода DVS-а. Метод развијен у [65] има најмању област конвергенције од свих алгоритама DVS-а и висок ниво тачности па је из тог разлога баш тај алгоритам интегрисан са честичним филтером. Честични филтер састоји се од роја честица случајно распоређених од

почетног до крајњег положаја камере. Честице представљају потенцијалне положаје камере. Виртуалне слике добијене на основу тих случајних положаја генеришу се коришћењем хомографије. Да би се одабрало у који потенцијални положај ће се робот померити, извршено је вредновање свих потенцијалних положаја. Метрика за вредновање заснивала се на заједничким информацијама добијеним на основу слика потенцијалног положаја и жељене слике. Затим, робот се помера из тренутног у положај дефинисан честицом, а ова процедура се наставља док се робот не нађе у положају блиском жељеном, када се активира DVS алгоритам предложен у [65]. Извршен је експеримент са реалним роботом, а након тога и симулација чиме је доказано да се предложеним алгоритмом остварује проширени опсег конвергенције у односу на алгоритам представљен у [65]. Недостатак предложеног система је потреба за естимацијом хомографије.

Још један значајан алгоритам директног визуелног управљања предложен је у [75]. Аутори су на основу Гаусових мешовитих модела представили слику помоћу Гаусових функција. Предложени алгоритам има високу тачност и у случају значајне удаљености између почетног и крајњег положаја. Извршени су експерименти на реалном роботу и у симулацији, у оквиру којих је доказана валидност предложеног метода. Примарни недостатак предложеног система је потреба да се подесе параметри алгоритма за другачије сцене и моделе камера. Примена слика трансформисаних у вејвлет и ширлет коефицијенте, које се користе у оквиру система директног визуелног управљања предложена је у раду [76]. Детаљан опис вејвлет трансформација може се пронаћи у литератури [80], [81]. Четири визуелна управљачка система предложена су и поређена са алгоритмом из референце [65]. Извршено је неколико експерименталних верификација са различитим поремећајним факторима (као што су промена осветљења или парцијално заклањање сцене) за сваки систем визуелног управљања. Током експеримената са поремећајним факторима, визуелни систем базиран на вејвлет трансформацији показао се као бољи са аспекта тачности. У оквиру рада [77], анализиран је систем директног визуелног управљања у фреквентном домену. Аутори су предложили трансформацију слике у дискретни косинусни домен и применили добијене коефицијенте за визуелно управљање. Извршени су експерименти са индустријским роботом са 6 степени слободе у оквиру различитих раванских и 3D сцена. Предност предложеног метода огледа се у великом домену конвергенције и погодним трајекторијама које роботски систем оствари.

Алгоритам визуелног управљања код кога се за управљање користи брза Фуријеова трансформација 3D облака тачака, предложен је у раду [78]. Након регистрације 3D облака тачака, резултати су трансформисани у фреквентни домен ради оцене транслације, чиме се остварује робустност на шум генерисан сензором. Предност овог метода у односу на остале које користе 3D облаке тачака (нпр. [70]) је да није потребно имати густи облак тачака и да је могуће извршити задовољавајућу регистрацију (а затим и визуелно управљање) чак и када је део сцене заклоњен. Применом ове методе могуће је извршити визуелно управљање на основу регистрације читаве сцене или одређеног објекта у њој. Експериментом са редундантним роботом приказана је валидност предложеног система. Визуелни систем управљања базиран на три типа ортогоналних момената предложен је у раду [79]. Развијена је адаптивна процедура за оптимизацију параметара везаних за трансформацију слика у домен ортогоналних момената, на основу чега се постиже форма Јакобијан матрице. Након многобројних експеримената у симулацији, којима је показана робустности на различите поремећајне факторе, предложени систем тестиран је на реалном индустријском роботу у оквиру раванске и 3D сцене. Два од три система визуелног управљања конвергирају ка жељеном положају у свим експериментима, а најбољи алгоритам је остварио грешку мању од 1 mm за 3D сцену.

4.1.2. Визуелно управљање мобилних робота

У референци [82] дефинисана је оптимална путања кретања мобилног робота на основу почетног положаја робота, позиције карактеристичних објеката, као и услова да карактеристични објекти остану у видном пољу камере. Трајекторије изведене у анализираном раду усвојене су као оптималне за многе развијене алгоритме визуелног управљања. У референци [83] анализиран је систем управљања мобилне роботске платформе са интегрисаном роботском руком у оквиру које је камера постављена на ендефектор робота. Прво је развијен систем визуелног управљања којим је извршено позиционирање ендефектора робота, без захтева везаних за мобилну платформу. У првој симулацији успешно је добијена конвергенција положаја камере до жељеног положаја. Други експеримент извршен је са циљем остваривања положаја мобилне платформе, док је све време робот са камером оријентисан ка положају објекта ка коме се креће. У раду [84] анализирано је управљање мобилног робота на основу визуелне меморије. Визуелна меморија представља скуп слика које су повезане у граф. Поменуте слике су генерисане пре почетка управљања у „*offline*“ режиму који представља процес обучавања робота. Са роботом су остварене трајекторије на основу даљинског управљања уз генерисање слика, након чега су слике повезане у граф тако да њиховим праћењем робот може да понови путању. Уколико слика у тренутном положају има довољно повезаних карактеристичних објеката са тренутном сликом из меморије, робот ће бити у стању да прати генерисану путању. Визуелно управљање, базирано на хомографији, вршено је све док се не остваре положаји дефинисани свим узастопним сликама из меморије. Камера је постављена вертикално навише на мобилни робот и све слике које се користе за управљање су слике плафона. Такође, потребно је нагласити да се применом ове методе не врши локализација, тј. робот нема информацију о свом положају у глобалном координатном систему, али може да прати путању дефинисану скупом слика.

Још један приступ у области визуелног управљања нехолономних мобилних робота предложен је у раду [85]. Аутори су развили систем визуелног управљања мобилних робота који се крећу помоћу точкова на бази епиполарне геометрије. Коришћени су карактеристични објекти на тренутној и циљној слици како би се одредили епиполови. У оквиру управљачког система није било потребно оцењивати положај робота, већ је само на основу параметара дефинисаних епиполова остварено визуелно управљање. Предложени управљачки систем имао је три фазе: фазу ротације ка жељеној позицији, транслацију до жељене позиције и ротацију ради остваривања жељеног положаја. У референци [86] анализирано је управљање мобилним роботом помоћу матрице хомографије са додатним ограничењем које је подразумевало да карактеристични објекти морају остати у видном пољу камере. За разлику од других радова, није коришћена хомографска матрица за естимацију положаја робота, већ су само њени елементи били употребљени за пројектовање система визуелног управљања. У поменутом раду доказана је управљивост и стабилност наведеног система визуелног управљања. Извршене су симулације и реални експеримент у коме је робот након кретања, пратећи путању дужине око 80 cm, остварио циљ са тачношћу од неколико центиметара. У овом раду оптимално планирање путање нехолономног мобилног робота извршено је на основу анализе представљене у [82].

У раду [87] анализирано је праћење путање мобилног робота на основу фузије сензорских информација са енкодера, камере и жироскопског сензора. Енкодер је служио за одређивање положаја и брзина мобилног робота, применом жироскопа је мерена угаона брзина, док су помоћу дискретне и континуалне хомографске матрице одређени положај, транслаторна и угаона брзина робота. Ради фузије поменутих сензорских информација коришћен је линеаризовани Калманов филтер. Камера је постављена на робот и оријентисана ка плафону. Путања је формирана од кривих вишег реда и извршено је неколико експеримената. У првом експерименту била је задата удаљеност камере од карактеристичних објеката, док је у другом тај параметар добијен естимацијом. Недостатак предложеног

алгоритма огледа се у потреби да транслаторна брзина увек буде константна. У референци [88] анализирана је интеграција камере, компас сензора, енкодера и IMU (енгл. *Inertial Measurement Unit*) сензора за естимацију промене положаја мобилног робота. Циљ овог рада био је анализа могућности фузије сензорских информација са поменутих сензора, уз перманентно отклањање кумулативне грешке која се јавља применом само једног сензора. Развијен је PBVS систем управљања за праћење путање и на основу њега је анализирано кретање робота. Извршен је експеримент праћења кружне и праволинијске путање. Недостатак предложеног алгоритма огледа се у чињеници да је позната само релативна промена положаја мобилног робота.

У референци [89] предложен је систем визуелног управљања, који узима у обзир непознати положај камере на роботу и непознату удаљеност карактеристичних објеката. Прва фаза овог управљачког алгоритма је померање робота транслаторно уназад, чиме се одређује разлика оријентације робота и камере. Када је тај параметар одређен, робот има две фазе које извршава. Прво се изврши ротација, у оквиру које се робот оријентише ка позицији коју треба да оствари, након чега робот почиње да се транслаторно креће ка свом циљу до одређеног задатог положаја. Предност предложене методе огледа се у робустности на грешке при калибрацији и позиционирању камере у оквиру роботског система. Анализа могућности праћења трајекторије на основу трифокалног тензора (енгл. *Trifocal tensor*) тема је рада [90]. Праћење трајекторије врши се на основу остваривања положаја дефинисаних сликама генерисаним током процеса обучавања робота. Трифокални тензор је коришћен уз ограничења која подразумевају да почетна и крајња сцена буду у видном пољу камере, међутим, у раду [90] претходно ограничење је отклоњено увођењем „кључних слика“ (енгл. *Key frames*) за естимацију положаја. За управљање користе се оријентација и скалирана удаљеност које се директно добијају применом трифокалног тензора. Такође, у управљачки систем је интегрисана процедура за естимацију положаја камере на роботу и естимацију удаљености карактеристичних објеката. Извршена је симулација и потврђена је ефикасност предложеног алгоритма.

У радовима [91], [92] и [93] анализиран је модел визуелног управљања базиран на интеграцији IBVS-а и PBVS-а, коришћењем модела предиктивног управљања (енгл. *Model Predictive Control*). Поменути модел коришћен је ради стабилизације ограничавајућих фактора IBVS-а у комбинацији са нехолономним ограничењима мобилног робота са диференцијалним погоном. Ограничавајући фактори анализирани у овом раду су: опсег управљачких величина, опсег промене управљачких величина, као и грешке система. Функција циља коју је потребно оптимизовати ради стабилизације поменутих утицаја дефинисана је Еуклидском дистанцом. За решавање овог оптимизационог проблема у реалном времену коришћена је техника вештачке интелигенције базирана на неуронским мрежама. Извршена су четири експеримента која потврђују валидност оваквог приступа.

Како је један од недостатака визуелног управљања потреба да се формира слика циља, у раду [94] анализира се визуелно управљање без слике циља. Да би се претходни задатак извршио, потребно је дефинисати три координатна система: базни, координатни систем прве и друге фазе. У првој фази робот се приближава циљу на претходно задато растојање, док је координатни систем друге фазе у жељеном положају робота. Да би се извршила промена положаја робота, естимира се висина карактеристичног објекта и користи се алгоритам за адаптивну промену те величине. Извршена је симулација и експеримент, на основу чега су аутори утврдили да је могуће користити естимацију висине карактеристичног објекта уместо слике циља за потребе визуелног управљања. У раду [95] представљен је систем управљања који врши естимацију удаљености карактеристичних објеката. Естимација удаљености извршена је на основу адаптивног управљачког система који модификује појачања на основу претходних стања. Такође, управљачки систем је пројектован у поларном координатном систему. Након доказивања стабилности система, извршена је симулација и експериментална верификација предложене методологије. Недостатак предложеног приступа је потреба за извршавањем процеса естимације хомографије.

Аутори рада [96] предлажу PBVS систем за визуелно управљање нехолономног мобилног робота при чему се у истој управљачкој петљи врши и естимација положаја камере у односу на робот. Предност предложеног система огледа се у могућности континуираног визуелног управљања чак и када спољашњи утицаји промене положај камере на роботу. Положај робота у PBVS управљачком систему добијен је на основу ArUco маркера. Након симулације и реалног експеримента, доказана је валидност предложеног система, као и његова робустност на шумове. Алгоритам визуелног управљања нехолономног мобилног робота који укључује ограничење дозвољених управљачких величина представљен је у раду [97]. Аутори физичка ограничења везана за актуаторе мобилног робота директно инкорпорирају у управљачки систем чиме се остварује тачније моделирање целокупног система, а на основу чега се директно утиче на тачност управљања. Применом представљеног управљачког система гарантује се да карактеристични објекти остају у видном пољу камере. Експерименталном евалуацијом на реалном роботу доказана је валидност предложеног система.

У раду [98] предложен је визуелни управљачки систем са подсистемом за детекцију поремећаја. Паралелно са системом визуелног управљања базираним на предиктивном моделу, имплементиран је нелинеарни систем за детекцију поремећаја чије дејство директно учествује у рачунању грешке, а самим тим и одређивању управљачких величина. Предност предложеног система огледа се у могућности компензације грешака и неочекиваних спољашњих утицаја који нису били моделирани у иницијалном управљачком систему. Симулацијом је приказана успешност предложеног управљачког система да компензује различите негативне спољашње утицаје при визуелном управљању. Како се различите технике вештачке интелигенције, а посебно технике машинског учења, све више примењују у области роботике, у наставку извршен је преглед стања примене машинског учења у визуелном управљању робота.

4.1.3. Машинско учење у визуелном управљању робота

Свеобухватна анализа могућности и ограничења примене дубоког машинског учења у области роботике приказана је у раду [99]. Установљена су три главна правца у оквиру којих дубоко машинско учење може да оствари значајна унапређења: учење, физичка интеграција и резоновање. У оквиру учења, највиши степен развоја довео би до активног учења, које подразумева да робот све време у току свог радног века учи и унапређује своју тачност на основу искуства и инструкција људи. Физичка интеграција подразумева коришћење временских и геометријских информација ради остваривања процеса активног гледања и манипулације. Активно гледање подразумева да уколико робот у тренутном положају није сигуран у стање окружења (нпр. ниска поузданост препознавања објеката манипулације), може предузети одређене акције како би повећао сигурност пре доношења одлука о извршавању нових акција. У оквиру резоновања, дубоко машинско учење омогућује интеграцију геометријских (мапе, позиције) и семантичких информација ради интегрисаног резоновања о новим акцијама које је потребно донети ради испуњења постављеног задатка. Иако ови циљеви нису остварени у радовима који су наведени у наставку, они представљају добар основ и перспективу за даља истраживања.

Алгоритам машинског учења ојачавањем интегрисан са вештачким неуронским мрежама примењен у сврху визуелног управљања представљен је у раду [100]. Хибридни управљачки систем састоји се од два алгоритма. Први алгоритам користи се за остваривање жељене оријентације робота на основу машинског учења ојачавањем и вештачких неуронских мрежа, а док други алгоритам представља стандардни IBVS управљачки систем. У зависности од параметра селекције, оба алгоритма се смењују у току визуелног управљања. Различите симулације и експерименти са реалним роботом извршени су ради валидације развијених алгоритама. У референци [101] предложена је интеграција визуелног управљања, неуронских мрежа и машинског учења ојачавањем, ради обучавања робота да изврши задатак проналажења и хватања задатог објекта. Роботски систем има две камере, једну постављену на робот, а другу

која се налази унутар хватача. Прва камера коришћена је за проналажење дела у оквиру сцене, а другом камером вршено је тачно позиционирање ради хватања објекта. Неуронска мрежа је обучена да као излаз генерише сигурност хватања на основу слике. На основу тог излаза, робот је обучен, применом Q-учења, да изврши позиционирање ради тачног хватања дела. У раду [102] анализира се могућност обучавања мобилног робота применом вештачких неуронских мрежа и учења демонстрацијом (енгл. *Learning from Demonstration*). У фази обучавања („*offline*“ фаза) се применом вештачких неуронских мрежа и учења демонстрацијом обучава робот да на основу разлике карактеристичних објеката у равни слике генерише вредност угаоне брзине потребне за достизање жељеног положаја. У „*online*“ фази користи се модел визуелног управљања базиран на епиполарној геометрији за одређивање транслаторне брзине потребне за остваривање жељеног положаја. На основу вредности функције циља карактеристичних објеката, активира се систем за одређивање транслаторне или угаоне брзине у сваком инкременту путање, чиме се обезбеђује кретање робота ка жељеном положају. Извршени су експерименти који показују робустност и тачност позиционирања предложеног система визуелног управљања.

У референци [103] анализира се могућност примене дубоког Q-учења ојачавањем за проблем манипулације, применом робота са три степена слободе који се креће у равни. Развијено је симулационо окружење и моделиран је роботски систем, како би се извршило обучавање робота. Модел робота представљен је из два сегмента, а виртуелном камером су снимана померања. У симулацију је имплементирана функција циља која роботу додељује позитивну вредност уколико се креће ка циљу, негативну ако се удаљава од циља, а не даје оцену уколико се није померио. Обучено је пет различитих модела, а сваком је додат одређени ниво поремећајних фактора (повећање шума, случајно иницијално померање камере, померање почетног положаја робота и промена дужине сегмента робота). Након обучавања, извршена је симулација у којој су сви модели тестирани са свим поремећајним факторима. Затим, модел који је остварио највећу тачност тестиран је на реалном роботу, али због мале тачности није било могуће извршити експеримент. Након тога, извршен је експеримент на реалном роботу, али са синтетичким сликама из симулације где је доказано да је на тај начин могуће управљати роботом. Анализирани рад представља увод у проблематику обучавања роботских система за задатке манипулације применом дубоког машинског учења ојачавањем. У раду [104] анализирано је визуелно управљање коришћењем IBVS система. Тема рада је развој система визуелног управљања за индустријски робот који има могућност да манипулише великим бројем различитих објеката. Генерисана је база података у којој се налазе поменути објекти, а сваки објекат има скуп слика другачијих димензија, текстура и оријентација. Након тога, формира се граф у оквиру кога су слике једног објекта чворови, а вредност метрике, која генерише дужину трајекторије остварену визуелним управљањем између те две слике, дефинише грану графа. Решавање проблема манипулације своди се на проналажење најкраће путање у графу између тренутне и жељене слике. Проналажење најкраће путање извршено је помоћу Дајкстриног алгоритма. Граф који је приказан као репрезентативан обухватао је 100.000 слика. Извршене су три симулације. Прва симулација извршена је коришћењем случајно одабраних слика са интернета. Друга симулација је извршена са сликама из базе за обучавање, са циљем одређивања тачности алгоритма. Трећа симулација извршена је ради детектовања утицаја величине базе података на тачност визуелног управљања. Просечна грешка износила је 23 cm на пређеном путу од 1,1 m, а са повећавањем броја слика у графу, повећава се и тачност визуелног управљања.

Примена конволуционих неуронских мрежа (енгл. *Convolutional Neural Networks* – CNN) за генерисање разлике положаја на основу две слике, анализира се у раду [105]. Предност предложеног алгоритма је да није потребно уводити претпоставке везане за камеру или окружење. За обучавање коришћена је већ постојећа база података, са додатним вештачким сликама добијеним хомографским трансформацијама. Оптички ток (енгл. *Optical Flow*) користи се за предикцију кретања камере у односу на стационарну 3D сцену (енгл. *Camera ego-*

motion). На крају обучавања се, на основу две слике, добија њихов узајамни положај који се користи у оквиру PBVS-а за остваривање жељеног положаја робота. Извршене су две симулације и један експеримент на реалном квадкоптеру, где је показано да предложени систем има задовољавајућу тачност. Највећи недостатак система су велики компјутерски ресурси потребни за извођење ове методе у реалном времену, као и велики меморијски захтеви ради примене CNN-а. Додатни допринос овог рада је и формирање базе података за тестирање алгоритама визуелног управљања.

У раду [106] аутори су развили методологију за учење експлицитног динамичког модела система управљања и карактеристичних објеката које је потребно користити за визуелно управљање. Анализа је вршена у симулационом окружењу, а задатак је био визуелно праћење возила. Коришћен је обучени CNN модел, да би се на основу излаза из скривених слојева издвојили сегменти слике који су коришћени за визуелно управљање. Како постоји више различитих излаза из скривених слојева, на основу апроксимације Q-функције извршено је дефинисање њихових тежинских коефицијената. Развијен је систем за учење динамике кретања возила на основу Адам оптимизационог алгоритма [107]. Након учења, извршава се предикција следећег вектора стања возила који се директно користи у визуелном управљању. Предложена методологија показала се као најбоља са аспекта тачности и брзине процесирања у поређењу са другим сличним приступима. У раду [108] анализира се примена CNN-а интегрисана са PBVS алгоритмом. Процес управљања врши се на основу разлике положаја тренутне и жељене слике. Разлика се генерише на основу излаза CNN-а, при чему је улаз у CNN тренутна слика. CNN модел се обучава на основу иницијално генерисане слике и синтетичким сликама добијеним на основу хомографских и оклузивних трансформација уз промене локалног и глобалног осветљења. Синтетичке слике омогућавају CNN моделу да буде робустан на промене осветљења и на парцијално заклањање сцене. Мрежа је дообучавана на основу *AlexNet* архитектуре, док је последњи слој дефинисан као потпуно повезани слој за одређивање положаја. Коришћењем претходно обучене *AlexNet* архитектуре смањено је потребно време обучавања. Предложени алгоритам тестиран је у симулацији и поређен са методом из [65], као и методом на бази честичног филтера [74]. Резултати поређења показали су да је алгоритам на бази CNN-а остварио проширени интервал конвергенције у односу на поређене методе. Два експеримента су извршена на индустријском роботу, при чему је у првом сцена представљала раван, а у другом 3D сцену. Главни недостатак анализиране методологије огледа се у потреби рада са сценом за коју је мрежа претходно обучавана. Из тог разлога, обучена је још једна мрежа која се може користити и за непознате сцене. Поменути тип мреже имплементиран је и тестиран, а добијена тачност износила је неколико центиметара, након чега је искоришћено фотометријско директно визуелно управљање ради добијања милиметарске тачности. Недостатак предложеног алгоритма представља мали интервал конвергенције јер је највећа дистанца помераја по једној оси износила 30 cm.

Стандардне методе визуелног управљања попут IBVS-а и PBVS-а није могуће применити у окружењима у којима се не могу детектовати карактеристични објекти па је у раду [109] предложен систем дубоког машинског учења који помаже при њиховој детекцији и упаривању. Коришћен је систем машинског гледања ради локализације објеката од значаја на слици, а затим је примењен систем дубоког машинског учења за њихово препознавање. Када је локализован и препознат објекат од значаја, извршена је детекција и упаривање карактеристичних објеката само на тим деловима слика. Овим процесом омогућено је робустније упаривање, које није било могуће извршити због униформности текстура и боја у окружењу у коме се робот налази. Затим је примењен стерео визуелни систем управљања ради остваривања задатог положаја. Директно визуелно управљање извршено на основу карактеристика генерисаних дубоким машинским учењем односно, конволуционим аутоенкодером, представљено је у раду [110]. Модел је обучен да изврши компресију слика, како би се одредили тежински коефицијенти који имају могућност смањења резолуције слика. Затим, аутори доводе у везу научене карактеристике модела са померањима камере и на основу

тога врши се директно визуелно управљање. Предложена репрезентација слике обезбеђује конвексну функцију циља погодну за оптимизацију Левенберг-Маркеовим алгоритмом. Експериментална верификација у симулацији и на реалном роботу показала је знатно већи домен конвергенције од метода директног визуелног управљања и сличну тачност остваривања жељеног положаја. Недостатак предложеног алгоритма огледа се у потреби за обучавањем мреже на сличним сценама на којима ће бити и коришћен систем. Наставак истраживања на поменуту тему резултирао је новом методом хибридног визуелног управљања [111] која интегрише претходно предложени систем са PBVS алгоритмом. Применом метричког учења, аутори предлажу обучавање конволуционог аутоенкодера како би слике генерисане на сличним положајима биле блиске у простору тежинских коефицијената. Предложеним системом знатно се повећава домен конвергенције визуелног управљања, а остварена је могућност тачног остваривања положаја и са великим ротацијама око Z осе од око 130°. Робустност предложеног алгоритма показала се при утицају поремећајних фактора који се односе на промене осветљења и заклањање сцене, међутим ови поремећајни фактори имају негативан утицај на грешку оствареног положаја. Недостатак предложеног метода огледа се у потреби да се метрика коришћена за обучавање аутоенкодера дефинише на основу скупа података, што може бити ограничавајући фактор за реалну примену.

Анализирани приступи у области визуелног управљања превасходно се ослањају на коришћење карактеристичних геометријских објеката (тачке, линије, контуре) или неких других података добијених са слике. Такође, у поређењу са *state-of-the-art* методама визуелног управљања, које се развијају као индивидуална алгоритамска решења, у докторској дисертацији се предлаже приступ код кога се информације коришћене за визуелно управљање могу такође применити у оквиру других управљачких алгоритама у оквиру интегрисаних система за остваривање аутономног понашања робота у динамичним окружењима. У том смислу, развијен је нови метод за стерео визуелно управљање који се заснива на семантичким информацијама, а које се могу применити за реализацију задатака избегавања колизије, разумевања сцене, локализације и аутономне манипулације. Семантичка сегментација је изабрана због својих карактеристика које обезбеђују значајне и структуиране информације о сцени, укључујући тачну класификацију и позиционирање свих објеката на нивоу пиксела. У оквиру новог алгоритма, фаза упаривања карактеристичних објеката замењује се процесом регистрације семантичких мапа, што омогућава управљање засновано на вишем нивоу репрезентације сцене, односно на семантички значајним објектима и њиховом распореду. Такође, један од недостатака визуелног управљања огледа се у потреби за претходним дефинисањем жељеног положаја робота, најчешће његовим постављањем у жељени положај и генерисањем циљних слика. Такав недостатак значајно ограничава степен аутономије и примењивост роботских система са визуелним управљањем у реалним индустријским условима. Са друге стране, предложени алгоритам заснован на семантичкој сегментацији има могућност аутоматског генерисања жељених слика у симулираном окружењу, што се може извршити с обзиром на то да су семантичке мапе базиране искључиво на класама објеката. Овакав приступ чини роботски систем робустнијим на реалне поремећајне факторе, као што су варијације у осветљењу, шум камере или промене у динамичком окружењу, а омогућава континуирано прилагођавање жељених положаја (односно генерисање нових парова жељених слика) и реализовање сложених транспортних задатака у потпуности без људске интервенције.

4.2. Визуелно управљање роботских система – математичко-алгоритамске основе

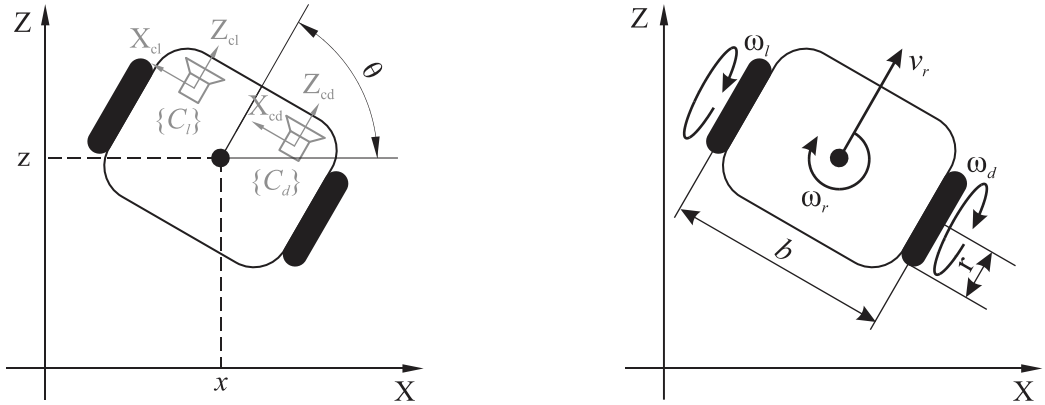
Како би се детаљно приказали алгоритми визуелног управљања мобилних робота, неопходно је дефинисати стање мобилних роботских система. У општем случају, стање система може бити дефинисано у зависити од различитих физичких величина које описују систем попут позиције, оријентације, брзине, убрзања система, нивоа батерија итд. У оквиру ове докторске дисертације, стање система биће дефинисано положајем (позицијом и оријентацијом (4.1)).

$$\mathbf{x} = \left(x, y, \mathrm{z}, \psi, \theta, \varphi\right)^T \tag{4.1}$$

Међутим, како је разматрани мобилни роботски систем пројектован за задатке транспорта у оквиру интелигентног технолошког окружења и креће се у равни, неопходне су три компоненте вектора стања $\mathbf{x} \in \mathbb{R}^{3\times1}$ (4.2) како би се у потпуности дефинисао његов положај.

$$\mathbf{x} = \left(x, z, \theta\right)^T \tag{4.2}$$

Мобилни робот у произвољном положају, који је дефинисан позицијом (x и z координате) и оријентацијом (угао θ), може се уочити на слици 4.2а). Поред самог положаја мобилног робота, приказани су и координатни системи леве $\{C_l\}$ и десне $\{C_d\}$ камере. Усвојена је стандардна конвенција при раду са камерама, која налаже да је X хоризонтална оса слике, вертикална оса слике је Y, док је Z оса нормална на раван слике. За означавање сваког од координатних система камера, као и њихових оса, додате су ознаке d и l, чиме се указује на то да ли се ради о десној или левој камери у стерео пару.



а) б)

Слика 4.2. Дефинисање а) координатних система мобилног робота и камера б) транслаторне брзине, угаоних брзина и параметара мобилног робота.

Модел кретања на основу пређеног пута, тј. одометрија (енгл. *Odometry*), усвојен је као модел промене стања мобилног робота. Управљачке величине у оквиру овог модела кретања су пређени пут левог и десног точка (Δs_l и Δs_d), а једначина промене стања је дата у (4.3):

$$\mathbf{x} = \begin{pmatrix} x \\ z \\ \theta \end{pmatrix} + \begin{pmatrix} \dfrac{\Delta s_d + \Delta s_l}{2} \cos\left(\theta + \dfrac{\Delta s_d - \Delta s_l}{2b}\right) \\ \dfrac{\Delta s_d + \Delta s_l}{2} \sin\left(\theta + \dfrac{\Delta s_d - \Delta s_l}{2b}\right) \\ \dfrac{\Delta s_d - \Delta s_l}{b} \end{pmatrix}, \tag{4.3}$$

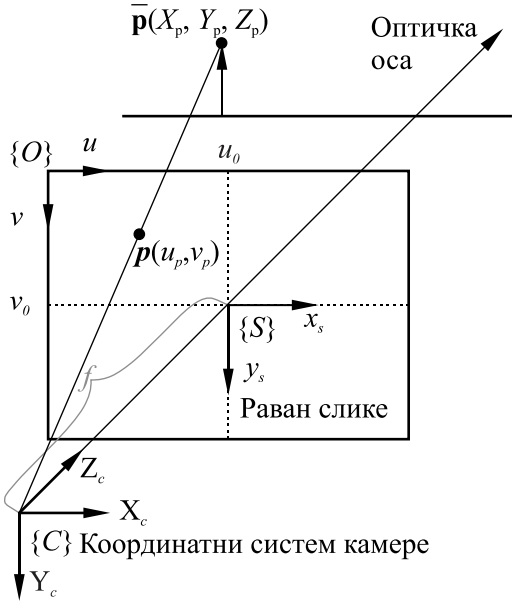
где b представља растојање између точкова. Детаљно извођење једначина модела кретања на основу пређеног пута може се пронаћи у оквиру литературе [3]. Такође, како би се остварила

угаона и транслаторна брзина (видети слику 4.2б) мобилног робота (ω_r и v_r), морају се задати одговарајуће угаоне брзине левом (ω_l) и десном точку (ω_d), на основу израза (4.4):

$$\begin{pmatrix} v_r + 0,5\omega_r b \\ v_r - 0,5\omega_r b \end{pmatrix} (1/r) = \begin{pmatrix} \omega_d \\ \omega_l \end{pmatrix}, \tag{4.4}$$

где је са r означен полупречник точка.

Након дефинисања основних параметара и једначина везаних за модел промене стања система мобилног робота, потребно је дефинисати слику и модел камере. Слика је дефинисана као ненегативни целобројни тензор $\mathbf{I} \in \mathbb{Z}^{H \times W \times D}$, где H и W представљају резолуцију слике, а D дефинише да ли је слика монохроматска (D=1) или у боји (D≥3). Вредности осветљености пиксела у оквиру слике су у опсегу од 0 до 255. У специфичним случајевима, вредности интензитета пиксела слике могуће је скалирати на опсег од нула до један. Усвојен је општеприхваћени модел инфинитезимално малог отвора бленде (енгл. *Pinhole camera model*). Овај модел користи се за математичко моделирање перспективне пројекције 3D координата објеката у сцени на раван слике (слика 4.3).



Слика 4.3. Модел инфинитезимално малог отвора бленде.

Нека је један детектовани карактеристични објекат у сцени дефинисан вектором познатих 3D координата $\bar{\mathbf{p}} = \left(X_p, Y_p, Z_p\right)^T$, тада је могуће извршити перспективну трансформацију и дефинисати координате детектоване тачке у равни слике (у координатном систему слике $\{S\}$), применом Талесове теореме (4.5):

$$\mathbf{p} = \begin{pmatrix} x_{sp} \\ y_{sp} \end{pmatrix} = \frac{f}{Z_p} \begin{pmatrix} X_p \\ Y_p \end{pmatrix}, \tag{4.5}$$

где f представља фокалну (жижну) даљину у милиметрима, која је физички параметар камере, односно сочива. Међутим, када се генерише слика на камери и препозна објекат на слици, позиција тачке $\mathbf{p}$ у равни слике дефинисана је позицијом пиксела (u_p, v_p) у основном координатном систему слике $\{O\}$. Уколико се изврши процес калибрације камере и добије матрица калибрације $\mathbf{K}$ (која ће детаљно бити објашњена у даљем тексту), могуће је трансформисати позиције пиксела (u_p, v_p) из основног координатног система {O} у координатни систем слике $\{S\}$, односно у координате x_{sp} и y_{sp}. Пресликавања из основног у координатни систем слике дато је једначином (4.6):

$$\begin{pmatrix} x_{sp} \\ y_{sp} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -u_0\rho_u \\ 0 & 1 & -v_0\rho_v \end{pmatrix} \begin{pmatrix} u_p\rho_u \\ v_p\rho_v \\ 1 \end{pmatrix}, \tag{4.6}$$

где ρ_u и ρ_v представљају димензије пиксела дуж u и v оса (што је такође параметар сензора), респективно, а u_0 и v_0 дефинишу позицију где оптичка оса пресеца раван слике (видети слику 4.3) која у општем случају не мора бити у самом центру слике. Као што је већ поменуто, процес креирања слике је перспективна трансформација тродимензионог простора на дводимензиону раван слике, чиме се подаци о удаљености губе. Из тог разлога, уводи се математички формализам у виду хомогених координата које обезбеђују да се тачка $\tilde{\mathbf{p}} = (x', y', z')^T$ на слици у тродимензионом простору налази на било којој удаљености од равни слике. Са z', уводи се додатна оса у координатном систему камере {C}. Веза хомогених и нехомогених координата дата је једначином (4.7).

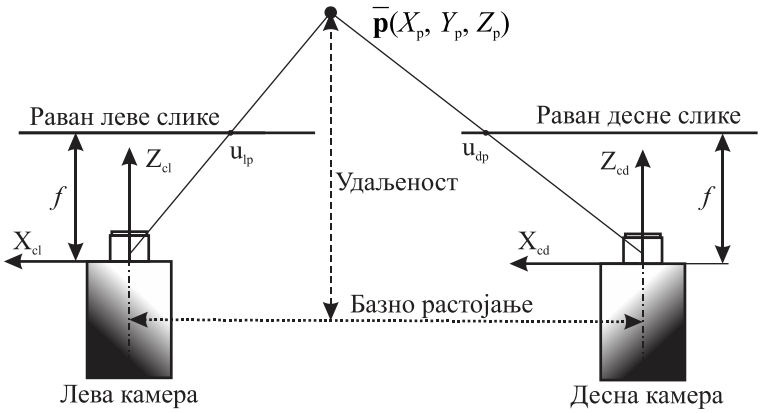
$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} / z', \tag{4.7}$$

Целокупна перспективна трансформација, која директно повезује 3D координате тачке и њихову позицију на равни слике у хомогеним координатама основног координатног система, дата је једначином (4.8):

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f/\rho_x & 0 & u_0 \\ 0 & f/\rho_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{K} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{4.8}$$

где **K** представља калибрациону матрицу, f је фокална дужина сочива, ρ_x и ρ_y су димензије пиксела дуж x и y осе. Овако дефинисана калибрациона матрица подразумева да не постоји дисторзија при креирању слика, што и јесте случај за сочива која имају велику фокалну дужину. Иако параметри f, ρ_x и ρ_y представљају карактеристике сочива, односно сензора камере, вредности ових параметара добијају се процесом калибрације камере. Поред ова три параметра, калибрацијом камере добијају се вредности и за параметре u_0 и v_0. У литератури су предложени различити алгоритми за калибрацију камере, а у оквиру ове докторске дисертације коришћен је стандардни метод са шаблоном шаховске табле [112], [113].

Недостатак визуелног система са једном камером је немогућност одређивања удаљености карактеристичних објеката од камере без додатних информација о окружењу. Овај недостатак може се отклонити додавањем још једне паралелно постављене камере, чиме се остварује стерео визуелни систем (слика 4.4) инспирисан људским визуелним системом. Потребно је напоменути да је процес калибрације стерео визуелног система значајно комплекснији од калибрације једне камере. Поред већ поменутих параметара који се генеришу калибрацијом (у овом случају за сваку камеру понаособ), у стерео визуелном пару потребно је добити и тачан положај једне камере у односу на другу, дефинисан матрицом ротације и вектором транслације. Када је извршен процес калибрације и детектован исти објекат на обе камере (тачка $\bar{\mathbf{p}}$, дефинисана координатама u_{lp}, v_{lp}, u_{dp}, v_{dp} у равни слике), могуће је оценити његову удаљеност (Z_p) применом процеса триангулације [113].



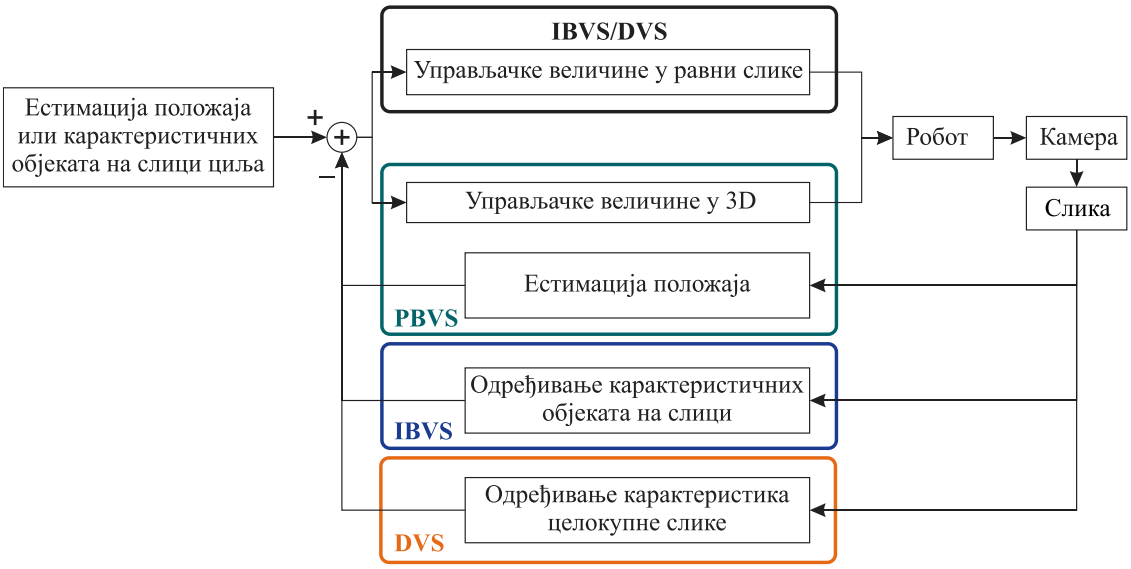
Слика 4.4. Стерео визуелни систем – основне карактеристике.

Након дефинисања координатних система робота и процеса формирања слике, биће дефинисане основе методе визуелног управљања. Процес визуелног управљања подразумева коришћење информација добијених од камере ради остваривања жељеног положаја роботског система [59], [114], [115]. Пре самог визуелног управљања, потребно је поставити роботски систем у жељени положај и генерисати циљну слику. Такође, важно је напоменути да жељени положај није познат управљачком систему, већ је једина информација коју систем поседује о жељеном положају циљна слика. Када се у тренутном положају генерише тренутна слика, потребно је израчунати грешку (4.9), тј. разлику у параметрима ове две слике. У току визуелног управљања, на основу израчунате грешке, генеришу се брзине и угаоне брзине које робот треба да оствари како би постигао жељени положај, а које уједно обезбеђују конвергенцију грешке ка нули.

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^* \tag{4.9}$$

У оквиру једначине (4.9), $\mathbf{m}(t)$ представља вектор измерених визуелних података са слике, $\mathbf{a}$ је матрица која додатно описује систем (нпр. 3D модел објекта, калибрациона матрица или подаци о ArUco маркеру који треба детектовати), док су $\mathbf{s}$ и $\mathbf{s}^*$ вектори карактеристичних објеката на тренутној и циљној слици, респективно.

Основна разлика између метода визуелног управљања је у подацима који се користе у управљачком систему, што је приказано на слици 4.5. Хибридни системи визуелног управљања представљају методе код којих се примарно интегришу IBVS и PBVS или један од ова два метода са различитим алгоритмима који обезбеђују поједине погодности за целокупни систем управљања [116].



Слика 4.5. Разлике метода визуелног управљања.

Код управљања на основу грешке у параметрима слике (IBVS), карактеристични објекти могу бити тачке, праве, кружнице или неки други геометријски примитиви, дефинисани у равни слике, док матрица **a** представља калибрациону матрицу. Како је грешка дефинисана у равни слике, овај метод управљања се такође назива и 2D визуелно управљање. Брзина која се генерише применом овог метода визуелног управљања дефинисана је својом транслаторном и угаоним компонентом у координатном систему камере (4.10).

$$\mathbf{v} = \left(\mathbf{v}_c, \boldsymbol{\omega}_c\right)^T \tag{4.10}$$

Након дефинисања брзине, потребно је дефинисати математичку зависност брзине камере (тј. робота) и грешке карактеристичних објеката (4.11),

$$\dot{\mathbf{e}} = \mathbf{J}\mathbf{v}\,, \tag{4.11}$$

где $\dot{\mathbf{e}} = \lambda\left(\mathbf{s} - \mathbf{s}^*\right)$ представља промену грешке, λ је појачање, а **J** је Јакобијан матрица слике (енгл. *Jacobian image matrix* или *Interaction matrix*, која се у литератури често означава и са $\mathbf{L_s}$). Затим, ради извођења Јакобијан матрице слике, потребно је увести математичку зависност 3D позиције карактеристичног објекта у сцени $\bar{\mathbf{p}} = \left(X_p, Y_p, Z_p\right)^T$ и брзине кретања роботског система (4.12):

$$\dot{\bar{\mathbf{p}}} = -\boldsymbol{\omega} \times \bar{\mathbf{p}} - \mathbf{v}\,, \tag{4.12}$$

пројектовањем једначине (4.12) на три осе координатног система камере, добија се једначина (4.13):

$$\begin{aligned} \dot{X}_p &= Y_p\omega_z - Z_p\omega_y - v_x \\ \dot{Y}_p &= Z_p\omega_x - X_p\omega_z - v_y\,. \\ \dot{Z}_p &= X_p\omega_y - Y_p\omega_x - v_z \end{aligned} \tag{4.13}$$

Коришћењем једначине (4.13) и трансформације 3D координата сцене у 2D простор слике из једначине (4.5), може се представити први извод брзине координата у равни слике као (4.14):

$$\dot{x}_{cp} = \frac{\dot{X}_p Z_p - X_p \dot{Z}_p}{Z_p^{\,2}}, \dot{y}_{cp} = \frac{\dot{Y}_p Z_p - Y_p \dot{Z}_p}{Z_p^{\,2}}, \tag{4.14}$$

затим, могуће је једначину (4.13) уврстити у (4.14) и трансформисати у матрични облик приказан у једначини (4.15) [115], где су због прегледности изостављене ознаке карактеристичног објекта **p**.

$$\begin{pmatrix} \dot{\mathrm{x}} \\ \dot{\mathrm{y}} \end{pmatrix} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -\left(1+x^2\right) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}. \tag{4.15}$$

Применом једначине (4.16), могуће је извршити трансформацију из координатног система камере (4.15) у нормализовани координатни систем слике дефинисани у пикселима (4.17) [117]:

$$x = \frac{\rho_u}{f}\bar{u}, y = \frac{\rho_y}{f}\bar{v}, \tag{4.16}$$

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{v}} \end{pmatrix} = \begin{pmatrix} -\frac{f}{\rho_u Z} & 0 & \frac{\bar{u}}{Z} & \frac{\rho_u \bar{u}\bar{v}}{f} & -\frac{f^2+\rho_u^2\bar{u}^2}{\rho_u f} & \bar{v} \\ 0 & -\frac{f}{\rho_v Z} & \frac{\bar{v}}{Z} & \frac{f^2+\rho_v^2\bar{v}^2}{\rho_v f} & -\frac{\rho_v \bar{u}\bar{v}}{f} & -\bar{u} \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}, \tag{4.17}$$

где је грешка позиције карактеристичног објекта на слици у координатном систему слике {S} дефинисана као $\bar{u} = u - u_0, \bar{v} = v - v_0$. Једначина (4.15) представљена у векторском облику дата је у једначини (4.18):

$$\dot{\mathbf{p}} = \mathbf{J_p}\mathbf{v}. \tag{4.18}$$

Како је за визуелно управљање неопходно детектовати минимално четири независна карактеристична објеката на слици, једначина (4.18) проширује се за сваки индивидуални карактеристични објекат (4.19):

$$\begin{pmatrix} \dot{\mathbf{p}}_{p1} \\ \dot{\mathbf{p}}_{p2} \\ \dot{\mathbf{p}}_{p3} \\ \dot{\mathbf{p}}_{p4} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{p1} \\ \mathbf{J}_{p2} \\ \mathbf{J}_{p3} \\ \mathbf{J}_{p4} \end{pmatrix} \mathbf{v}. \tag{4.19}$$

Затим, применом регулатора пропорционалног дејства $\dot{\mathbf{p}} = \lambda(\mathbf{p} - \mathbf{p}^*)$ у оквиру једначине (4.19) остварени су сви захтеви потребни за формализацију визуелног управљачког система (4.20):

$$\mathbf{v} = \lambda \cdot \begin{pmatrix} \mathbf{J}_{p1} \\ \vdots \\ \mathbf{J}_{pk} \end{pmatrix}^{-1} \cdot \begin{pmatrix} \mathbf{p}_1 - \mathbf{p}_1{}^* \\ \vdots \\ \mathbf{p}_k - \mathbf{p}_k{}^* \end{pmatrix}, \tag{4.20}$$

где k представља број детектованих карактеристичних објеката. Јакобијан матрицу $\mathbf{J_p}$ у својој иницијалној форми није могуће директно одредити због недостатка информација везаних за удаљеност карактеристичних објеката Z. Међутим, могуће је извршити естимацију непознатих параметара и користити апроксимацију матрице $\hat{\mathbf{J}}_p \in \mathbb{R}^{k\times 6}$. Како матрица $\hat{\mathbf{J}}_p$ није квадратна, није могуће стандардном процедуром одредити њену инверзну форму, већ се сингуларном декомпозицијом матрица може одредити псеудо-инверзна матрица $\hat{\mathbf{J}}_p^+ \in \mathbb{R}^{6\times k}$ [118]. Коначни векторски облик управљачког система базираног на IBVS методи визуелног управљања дат је једначином (4.21).

$$\mathbf{v} = \lambda \cdot \hat{\mathbf{J}}_p^+ \cdot (\mathbf{p} - \mathbf{p}^*) \tag{4.21}$$

Основна предност IBVS метода огледа се у робустности на грешке при естимацији удаљености карактеристичних објеката од равни слике и параметара калибрационе матрице. Први недостатак односи на генерисање вектора брзина на основу параметара дефинисаних у оквиру 2D слике, чиме може доћи до стања у коме се карактеристични објекти не могу детектовати. У том случају, процес визуелног управљања није могуће наставити и потребно је применити

одређене процедуре за наставак управљања. Утицај поменутог недостатка може се минимизирати интегрисањем додатних процедура у алгоритам управљања, које обезбеђују да карактеристични објекти остану у сцени током управљања [119]. Други недостатак IBVS-а огледа се у томе да се не генерише експлицитна разлика у тренутном и жељеном положају робота па брзине које се добијају нису оптималне са аспекта најкраћег пута између положаја. Из овог разлога, предложен је други метод визуелног управљања, назван управљање на бази издвојених карактеристика са слике неопходних за естимацију положаја изабраног објекта (PBVS).

Код основне једначине за рачунање грешке визуелног управљања (4.9) PBVS алгоритма, у оквиру вектора **s** и **s*** дефинисан је положај камере, тј. робота у тренутном и жељеном тренутку. Дакле, грешка представља разлику естимираних положаја, на основу које се може извршити планирање путање роботског система. Вектор **a** у оквиру једначине за дефинисање грешке (4.9) може бити било који скуп података који омогућава оцену положаја камере у односу на препознати карактеристични објекат на слици нпр. 3D модел објекта. Уколико је грешка дефинисана вектором положаја (**t**) и матрицом ротације (**R**), односно разликом у Ојлеровим угловима (θ**u**) између тренутног и жељеног положаја, брзине које треба задати мобилном роботу како би остварио жељени положај приказане су једначином (4.22):

$$\begin{aligned} \boldsymbol{v}_c &= -\lambda \boldsymbol{R}^T \boldsymbol{t} \\ \boldsymbol{\omega}_c &= -\lambda \theta \boldsymbol{u} \end{aligned}, \tag{4.22}$$

где λ представља појачање. Како би се имплементирао PBVS метод, потребно је применити неки од алгоритама за оцену положаја мобилног робота. У оквиру предметне докторске дисертације примењен је алгоритам који на основу матрице хомографије врши оцену положаја мобилног робота. Уколико је на тренутној и слици циља позната позиција минимално четири карактеристична објекта који припадају истој равни (π) у простору, могуће је извршити естимацију матрице перспективне хомографије (**G**). Поменута матрица директно повезује позиције карактеристичних објеката на слици циља $\mathbf{p}^* = (u^*, v^*, 1)$ и тренутној слици $\mathbf{p} = (u, v, 1)$ у хомогеним координатама основног координатног система дефинисаног у пикселима (4.23).

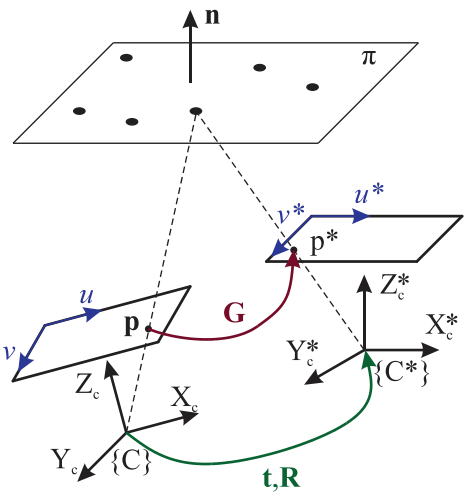
$$\mathbf{p}^* = \mathbf{G}\,\mathbf{p}\,. \tag{4.23}$$

Алгоритам за естимацију матрице хомографије на основу сингуларне декомпозиције матрице и детектованих карактеристичних објеката који припадају једној равни у простору дат је у литератури [113] (алгоритам 4.2), док се имплементација може пронаћи у различитим библиотекама за обраду дигиталне слике отвореног приступа [117], [120].

Поред поменуте трансформације, перспективна матрица хомографије (**G**) такође може бити дефинисана помоћу матрице ротације $\mathbf{R} \in \mathbb{R}^{3\times3}$ и вектора транслације $\mathbf{t} = (t_x, t_y, t_z)^T$ између жељеног и тренутног положаја мобилног робота (4.24):

$$\mathbf{G} = \gamma \mathbf{K}\left(\mathbf{R} + \mathbf{t}\mathbf{n}^T\right)\mathbf{K}^{-1}\,. \tag{4.24}$$

где је са $\mathbf{n} = (n_x, n_y, n_z)^T$ представљен вектор који дефинише нормалу равни π, **K** је калибрациона матрица (једначина 4.8), док је γ фактор скалирања. Примена матрице хомографије из једначина (4.23) и (4.24), приказана је на слици 4.6 [121], где су са {C} и {C*} означени координатни системи камере у тренутном и циљном положају, респективно.



Слика 4.6. Примена матрице хомографије.

Како је позната матрица калибрације и оцењена матрица хомографије, могуће је извршити декомпозицију матрице **G** ради одређивања вектора **t**, **n** и матрице **R**. Алгоритам за декомпозицију перспективне матрице хомографије предложен је у литератури [122], док се имплементација може наћи у [120]. Резултат декомпозиције су четири решења (4.25) за поменуте векторе и матрице,

$$\begin{aligned} Rtn_a &= \{\mathbf{R}_a, \mathbf{t}_a, \mathbf{n}_a\} \\ Rtn_b &= \{\mathbf{R}_b, \mathbf{t}_b, \mathbf{n}_b\} \\ Rtn_{a-} &= \{\mathbf{R}_a, -\mathbf{t}_a, -\mathbf{n}_a\} \\ Rtn_{b-} &= \{\mathbf{R}_b, -\mathbf{t}_b, -\mathbf{n}_b\} \end{aligned} . \tag{4.25}$$

Као што се може уочити, постоје два потпуно различита решења (означена са *a* и *b*) и њихове негативне вредности. На основу визуелног ограничења [122], могу се одбацити два нетачна решења применом неједначине (4.26) на карактеристичне објекте на слици циља (**p***) у нормализованим хомогеним координатама:

$$\mathbf{p}^{*T}\,\mathbf{n} < 0. \tag{4.26}$$

Како је остало још два решења од којих је потребно одабрати право, могуће је искористити погодност кретања у равни које гарантује да се координатни систем камере неће значајно померати по Y оси између почетног и крајњег положаја, на основу чега је могуће одредити индекс правог решења применом (4.27):

$$idx = min\left(\left[\mathbf{dt}_1(2), \mathbf{dt}_2(2)\right]\right), \tag{4.27}$$

где су $\mathbf{dt}_1(2)$ и $\mathbf{dt}_2(2)$ други елементи вектора разлике између почетне и крајње позиције, добијени на основу два решења која нису елиминисана претходно примењеним ограничењем, док *min* представља функцију која селектује најмањи елемент у вектору. Када су одређене праве вредности матрице ротације **R**, као и вектор транслације **t** и нормале **n** равни π, могуће је применити PBVS алгоритам. Међутим, постоји недостатак у коришћењу ограничења (4.27) у тренутку када се мобилни робот нађе близу жељеног положаја. Тада разлика у позицији мобилног робота, па самим тим и Y координати, постаје мала чиме због шума може доћи до избора погрешног решења. Овај недостатак могуће је елиминисати применом ограничења (4.28) које се може применити само након тачно одређеног решења у претходној итерацији алгоритма. Ограничење (4.28) је могуће дефинисати због велике фреквенције извршавања управљачке петље која проузрокује инкременталне промене позиције мобилног робота у две

узастопне итерације алгоритма, тако да се на основу норме разлике тренутног и претходног вектора позиције селектује тачно решење.

$$idx = min\left(\left\|\mathbf{t}_{old} - \mathbf{t}_i\right\|_2\right),\ i = 1,...,4 \tag{4.28}$$

Када су одређени вектор транслације **t** и матрица ротације **R** између тренутног и жељеног положаја, могуће је њихове вредности уврстити у једначину (4.22).

Примарни недостатак IBVS и PBVS система огледа се у потреби за алгоритмима који обезбеђују робустност при детекцији, упаривању и праћењу карактеристичних објеката на слици. Ови алгоритми су комплексни [123], склони шуму, што има значајан утицај на успешност, као и тачност визуелног управљања. Имајући то у виду, у последњих неколико година развијени су системи за директно визуелно управљање (енгл. *Direct Visual Servoing* – DVS) који не користе информације о карактеристичним објектима. Ови системи користе разлику интензитета свих пиксела на тренутној и циљној слици [61], [65] или неке друге податке са слике за визуелно управљање. Поменути подаци могу представљати фотометријске моменте (енгл. *Photometric moments*) [69], фотометријске Гаусове мешовите моделе (енгл. *Photometric Gaussian mixture models*) [71], [124], хистограме [72], [73] или кернеле (енгл. *Kernel*) [63]. Главна предност DVS алгоритама је у већој тачности оствареног положаја у односу на IBVS и PBVS. Ова предност настаје због знатно већег броја података који се користе у оквиру DVS-а (нпр. интензитет свих пиксела слике у односу на позицију четири карактеристична објекта) чиме се остварује већа робустност управљачког система и смањују утицаји шумова. Грешка која може бити остварена имплементацијом DVS алгоритма из рада [61] на индустријски робот је мања од једног пиксела. Међутим, главни недостатак ових метода огледа се у уском домену конвергенције, због чега их није могуће користити када је разлика почетног и жељеног положаја већа од поменутог домена (величина домена зависи од алгоритма DVS-а). Из тог разлога, истраживачи су почели да развијају алгоритме који имају већи домен конвергенције применом другачијих података прикупљених са слика. Такође, постоје и хибридни DVS алгоритми, код којих би се алгоритам који има мању тачност (а већи домен конвергенције) први применио, а затим би се активирао DVS алгоритам ради остваривања високе тачности жељеног положаја.

Већина управљачких система на бази DVS дефинисана је као оптимизациони проблем (4.29) где је потребно променом тренутног положаја **x** смањити вредност грешке дефинисане разликом између података са слике (који су дефинисани функцијом *f*) у тренутном и жељеном положају:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}}\left(f\left(\mathbf{I}(\mathbf{x})\right) - f\left(\mathbf{I}^*\right)\right). \tag{4.29}$$

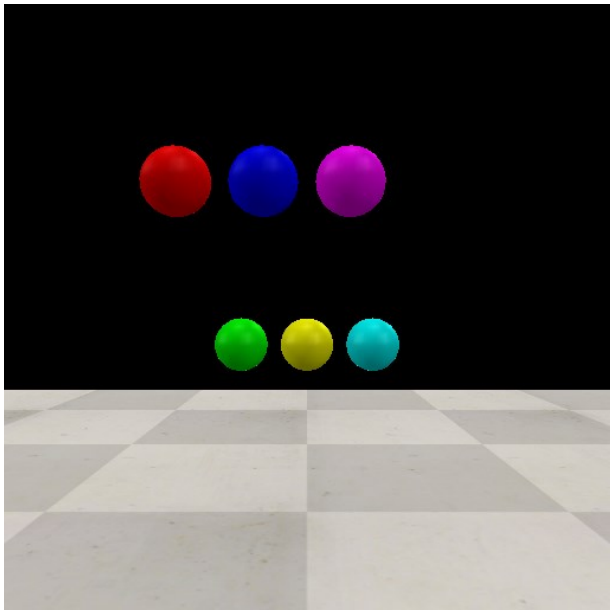
Различити алгоритми визуелног управљања разликују се по подацима, тј. функцији коју користе за представљање слика. Код фотометријског визуелног управљања [61], сви пиксели слике користе се за формирање управљачког система па функција циља представља квадратну разлику целокупних слика. Различити алгоритми конвексне оптимизације могу се користити за решавање овог проблема, али се углавном користе градијентни поступак или Левенберг-Маркеов алгоритам, што је и случај код фотометријског визуелног управљања (4.30):

$$\mathbf{v} = -\lambda\left(\mathbf{H} + \mu\,\mathrm{diag}\left(\mathbf{H}\right)\right)^{-1}\mathbf{J}^T e. \tag{4.30}$$

где је λ појачање, **H** представља Хесијан матрицу, μ представља параметар који дефинише утицај Хесијан матрице, **J** је Јакобијан матрица, а e представља функцију грешке. Како алгоритми DVS имају мали домен конвергенције, њихово поређење са IBVS и PBVS алгоритмима није практично изводљиво, а поготово у домену мобилне роботике.

4.2.1. Поређење IBVS и PBVS алгоритама у симулацији

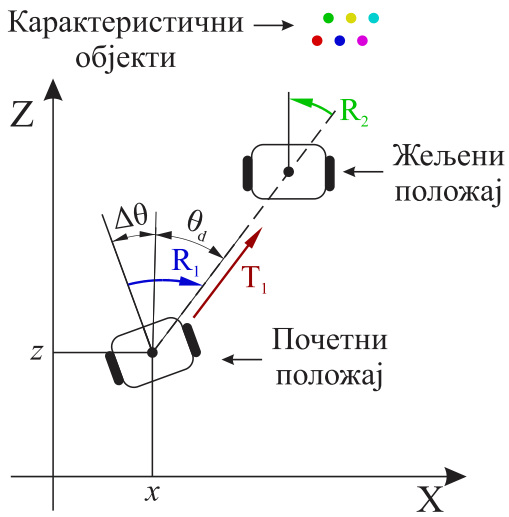
У оквиру овог потпоглавља извршена је упоредна анализа алгоритама IBVS и PBVS. Поређење је извршено применом модела мобилног робота са диференцијалним погоном који је имплементиран у софтверском пакету *CoppeliaSim* [125] (софтвер је раније носио назив V-REP), а у оквиру кога су дефинисана сва динамичка и кинематичка ограничења везана за кретање. *CoppeliaSim* је софтвер који се користи у сврху експерименталне верификације алгоритама са различитим мобилним и индустријским роботима. Интегрисани су различити модели мотора, елементи за повезивање механичке структуре робота и сензори (као што су сензори удаљености, камере и сензори додира). Експеримент је постављен тако да су алгоритми за визуелно управљање имплементирани у MATLAB софтверском пакету, док се подаци са камере прослеђују из софтвера *CoppeliaSim*. Када се изврше алгоритми у MATLAB-у и израчунају управљачке величине, оне се прослеђују моторима мобилног робота у оквиру *CoppeliaSim* окружења. У наставку потпоглавља детаљно су приказани алгоритми IBVS и PBVS имплементирани у оквиру софтверског окружења MATLAB ради њихове компаративне анализе. Карактеристични објекти коришћени у оквиру алгоритама визуелног управљања су центри сфера дефинисаних у окружењу мобилног робота (слика 4.7). Потребно је напоменути да позиција сфера није позната управљачком систему, већ се искључиво на основу генерисане слике и процеса обраде дигиталне слике генеришу центри сфера у равни слике.



Слика 4.7. Слика циља (**I***) у свим експериментима.

Пре почетка извршавања алгоритма, мобилни робот се поставља у жељени положај у оквиру кога се генерише и чува слика циља. Затим, покретањем алгоритма учитава се слика циља из меморије. Резолуција слика коришћених у обе симулације је дефинисана матрицама $\mathbf{I}, \mathbf{I}^* \in \mathbb{Z}^{512\times512\times3}$.

Како мобилни робот са диференцијалним погоном представља нехолономни систем и има ограничења у виду немогућности директне транслације у смеровима другачијим од тренутног, није могуће директно користити брзине генерисане у оквиру визуелног управљања. Из тог разлога, развијен је управљачки алгоритам који се састоји из три фазе. Прва фаза представља ротацију ка жељеној позицији (R_1 – слика 4.8), друга фаза је транслација до жељене позиције (T_1 – слика 4.8) и трећа фаза је ротација ради остваривања жељеног положаја (R_2 – слика 4.8). Брзина и угаона брзина које се задају мобилном роботу у свакој фази могу се уочити у табели 4.2.



Слика 4.8. Илустрација управљачког алгоритма који се састоји из три фазе.

Угао за који мобилни робот треба да се заротира (θ_t) у првој фази одређује се на основу односа транслаторних брзина у X и Z правцу добијених применом IBVS алгоритма визуелног управљања. Потребно је нагласити да се брзине на основу визуелног управљања генеришу у координатном систему камере (видети слику 4.2) па се мора извршити трансформација у координатни систем мобилног робота. Регулатор пропорционалног дејства коришћен је у свакој фази са одређеном вредношћу појачања. Свака фаза завршава се у тренутку када грешка (*err*) те фазе достигне вредност мању од задате вредности грешке (Δ_x), чиме започиње наредна фаза или се управљачки алгоритам завршава. Ред величине задате грешке и појачања у различитим фазама (видети табелу 4.2) је другачији због физичких величина које се користе у тој фази управљања (у првој фази је угао дефинисан у степенима, у другој транслаторна брзина у m/s, а у трећој угаона брзина у rad/s). Вредности угаоних брзина које се задају моторима мобилног робота зависе од појачања λ које се може подешавати и фигурише у једначини (4.21), тако да је потребно подесити максималне и минималне угаоне брзине које се могу проследити роботу. Такође, брзине остварене у оквиру друге фазе, тј. транслаторног кретања могу бити далеко веће него у фазама један и три где се извршава ротација. Из тог разлога, подешени су параметри ub_1, ub_2, lb_1, lb_2, којима се ове границе задају у алгоритму. Како се експерименти врше у симулацији, нема потребе за калибрацијом камере и експерименталним генерисањем вредности у матрици калибрације **K** па се сматра да се оптичка оса налази у центру равани слике и да нема дисторзије слике (једначина 4.31).

$$\mathbf{K} = \begin{pmatrix} 443,405 & 0 & 256 \\ 0 & 443,405 & 256 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.31}$$

Вредности свих осталих параметара који представљају улаз у алгоритам дати су у табели 4.2. Управљачка петља алгоритма се извршава у временским инкрементима од 0,075s.

Алгоритам за управљање нехолономног мобилног робота на бази PBVS-а дат је у табели 4.3. Форма алгоритма је слична као и код IBVS, а примарне разлике су наглашене у наставку. Првенствено, уместо директног израчунавања брзина које мобилни робот треба да оствари, код PBVS-а одређује се разлика између тренутног и жељеног положаја, која је дефинисана вектором транслације (**t**) и матрицом ротације (**R**). На основу те разлике, може се израчунати помоћни угао θ_z за који мобилни робот треба да се заротира (4.32):

$$\theta_z = \Delta\theta + \theta_d = \Delta\theta + atan2\left(\mathbf{t}(1), \mathbf{t}(3)\right), \tag{4.32}$$

где $\Delta\theta$ представља разлику између оријентације тренутног и жељеног положаја, а θ_d угао дефинисан разликом тренутне и жељене позиције (слика 4.8).

Табела 4.2. Алгоритам IBVS за управљање нехолономног мобилног робота.	
1.	Улазни подаци: *Step*=1; $k_{\omega 1}$=0,001; k_v=0,2; $k_{\omega 2}$=1; Δ_1=0,05; Δ_2=0,005; Δ_3=0,0001; b=0,380; r=0,0975; ub_1=0,4; lb_1= –0,4; ub_2=0,1; lb_2= –0,1; λ=0,1; **K**; **I***
2.	Генерисање карактеристичних објеката на слици циља → **p***
3.	**while** *Step* ≠ 4
4.	Аквизиција тренутне слике **I**
5.	Генерисање карактеристичних објеката на тренутној слици → **p**
6.	IBVS алгоритам: Рачунање грешке за све карактеристичне објекте **e** = **p** – **p*** (4.9) Нормализација координата **p**[u, v] → **p**[x, y] (4.6) Рачунање Јакобијан матрице за све карактеристичне објекте (4.15) Рачунање брзина мобилног робота **v** (4.21)
7.	Рачунање помоћног угла у $\theta_t = atan2(\mathbf{v}(1), \mathbf{v}(3))$
8.	**switch** *Step*
9.	**case** 1 (прва фаза) $err_{\omega 1} = -\theta_t$ $\omega_r = k_{\omega 1} \cdot err_{\omega 1}$; $v_r = 0$ **if** $err_{\omega 1} \leq \Delta_1$ *Step* = 2 **end_if**
10.	**case** 2 (друга фаза) $err_{v1} = \mathbf{v}(3)$ $\omega_r = 0$; $v_r = k_{v1} \cdot err_{v1}$ **if** $err_{v1} \leq \Delta_2$ *Step* = 3 **end_if**
11.	**case** 3 (трећа фаза) $err_{\omega 2} = -\mathbf{v}(5)$ $\omega_r = k_{\omega 2} \cdot err_{\omega 2}$; $v_r = 0$ **if** $err_{\omega 3} \leq \Delta_3$ *Step* = 4 **end_if**
12.	**end_switch**
13.	**if** *Step* == 2 $\omega_d = min(ub_1, max(lb_1, (v_r/r + 0.5 \cdot \omega_r \cdot b/r)))$ $\omega_l = min(ub_1, max(lb_1, (v_r/r - 0.5 \cdot \omega_r \cdot b/r)))$ **else** $\omega_d = min(ub_2, max(lb_2, (v_r/r + 0.5 \cdot \omega_r \cdot b/r)))$ $\omega_l = min(ub_2, max(lb_2, (v_r/r - 0.5 \cdot \omega_r \cdot b/r)))$ **end_if**
14.	Прослеђивање управљачких команди моторима мобилног робота (ω_d, ω_l)
15.	**end_while**

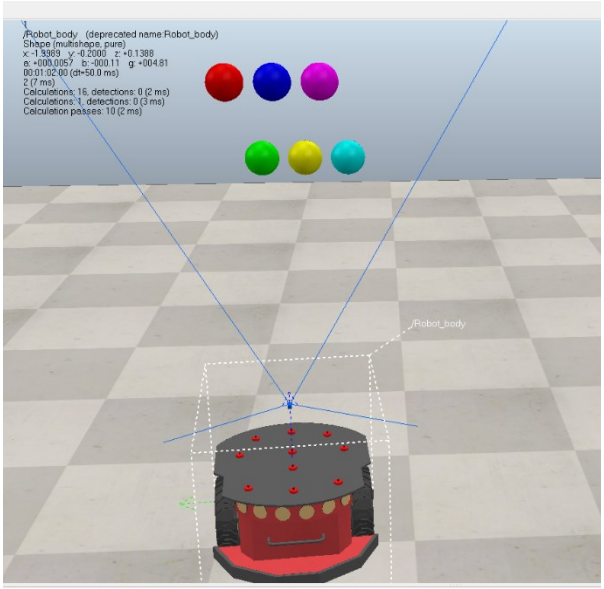
Угао $\Delta\theta$ одређује се на основу вредности у матрици ротације применом једначине (4.33):

$$\Delta\theta = -atan2\left(-\mathbf{R}(3,1), \sqrt{\mathbf{R}(1,1)^2 + \mathbf{R}(2,1)^2}\right), \tag{4.33}$$

У другој фази управљачког алгоритма као управљачка величина користи се пројекција дистанце по Z оси (у координатном систему камере) на Z осу равни π.

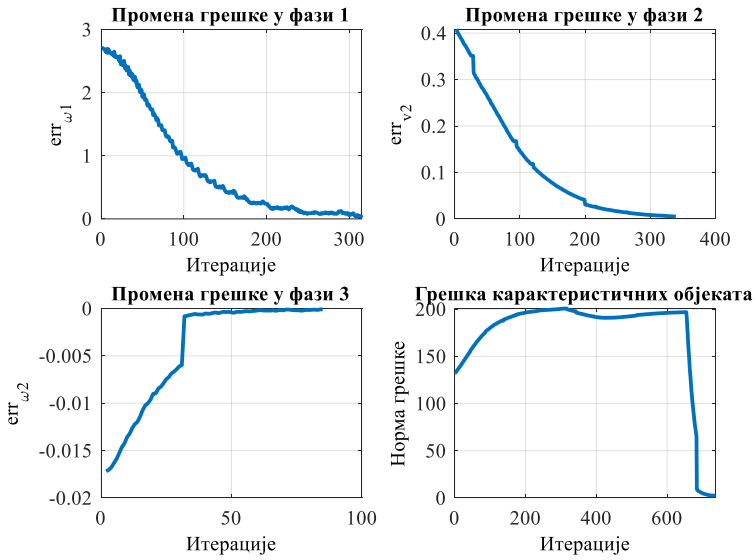
Табела 4.3. Алгоритам PBVS за управљање нехолономног мобилног робота.	
1.	Улазни подаци: *Step*=1; iter=0; $k_{\omega 1}$=0,001; k_v=0,2; $k_{\omega 2}$=0,001; Δ_1=0,05; Δ_2=0,005; Δ_3=0,1; b=0,380; r=0,0975; ub_1=0,4; lb_1= –0,4; ub_2=0,1; lb_2= –0,1; **K**; **I***
2.	Генерисање карактеристичних објеката на слици циља → **p***
3.	**while** *Step* ≠ 4
4.	Аквизиција тренутне слике **I**, iter++
5.	Генерисање карактеристичних објеката (центра сфера) на тренутној слици → **p**
6.	PBVS алгоритам: Одређивање матрице хомографије: **p**, **p*** → **G** (4.23) Декомпозиција матрице хомографије **G**, **K** → $\{\mathbf{R}_i, \mathbf{t}_i, \mathbf{n}_i\}$, $i = 1,...,4$, (4.25) **if** *iter* ==1 Одређивање правог решења на основу услова (4.27) и (4.28) → $\{\mathbf{R}, \mathbf{t}, \mathbf{n}\}$ Одређивање угла ротације мобилног робота (4.32) → θ_z **else** Одређивање правог решења на основу услова мале промене положаја (4.29) → $\{\mathbf{R}, \mathbf{t}, \mathbf{n}\}$ **end**
7.	**switch** *Step*
8.	**case** 1 (прва фаза) $err_{\omega 1} = \Delta\theta - \theta_z$ $\omega_r = k_{\omega 1} \cdot err_{\omega 1}$; $v_r = 0$ **if** $err_{\omega 1} \leq \Delta_1$ *Step* = 2 **end_if**
9.	**case** 2 (друга фаза) $err_{v1} = \mathbf{t}(3) \cdot \mathbf{n}(3)$ $\omega_r = 0$; $v_r = k_{v1} \cdot err_{v1}$ **if** $err_{v1} \leq \Delta_2$ *Step* = 3 **end_if**
10.	**case** 3 (трећа фаза) $err_{\omega 2} = -\Delta\theta$ $\omega_r = k_{\omega 2} \cdot err_{\omega 2}$; $v_r = 0$ **if** $err_{\omega 3} \leq \Delta_3$ *Step* = 4 **end_if**
11.	**end_switch**
12.	**if** *Step* == 2 $\omega_d = min(ub_1, max(lb_1, (v_r/r + 0.5 \cdot \omega_r \cdot b/r)))$ $\omega_l = min(ub_1, max(lb_1, (v_r/r - 0.5 \cdot \omega_r \cdot b/r)))$ **else** $\omega_d = min(ub_2, max(lb_2, (v_r/r + 0.5 \cdot \omega_r \cdot b/r)))$ $\omega_l = min(ub_2, max(lb_2, (v_r/r - 0.5 \cdot \omega_r \cdot b/r)))$ **end_if**
13.	Прослеђивање управљачких команди моторима мобилног робота (ω_d, ω_l)
14.	**end_while**

На основу предложених алгоритама за визуелно управљање нехолономних мобилних робота, извршено је поређење IBVS-а и PBVS-а у симулационом окружењу приказаном на слици 4.9.

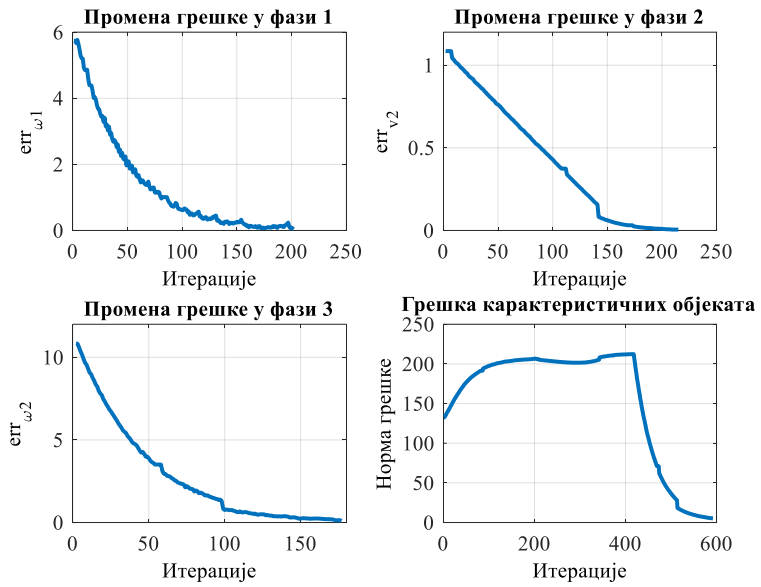


Слика 4.9. Симулационо окружење за поређење алгоритама IBVS и PBVS.

Прво су анализиране промене грешака по фазама и норма грешке карактеристичних објеката оба алгоритма и приказане на сликама 4.10 и 4.11, респективно. Као што се може уочити, фазе један оба алгоритма су сличног тренда, са главном разликом да код PBVS-а ова фаза траје значајано већи број итерација. Главна разлика која се може уочити између трендова промене грешке је у фази два, где код IBVS постоји повољан експоненцијални пад грешке, док се код PBVS-а може уочити приближно линеаран тренд смањења вредности грешке. Експоненцијални карактер криве конвергенције грешке је повољан јер се у мањем броју итерација може остварити жељена грешка. Такође, код IBVS-а у фази три уочава се неповољан нагли пад грешке што може довести до нестабилног понашања робота који користи IBVS управљачки систем, док је код PBVS-а тренд промене грешке повољан. На крају, промена грешке карактеристичних објеката слична је за оба алгоритма.

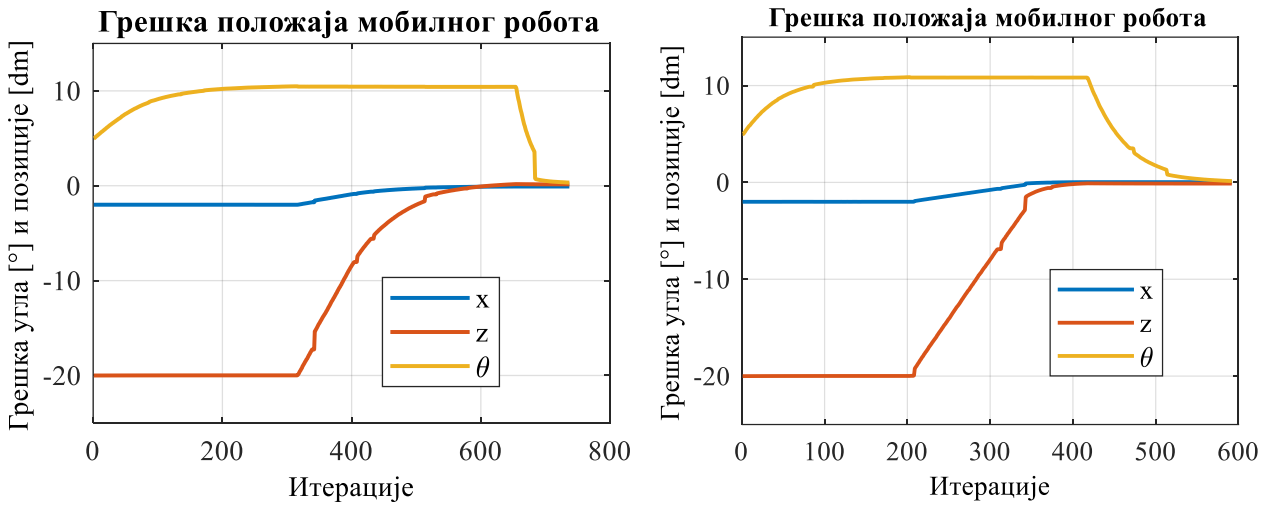


Слика 4.10. IBVS промена грешака по фазама.



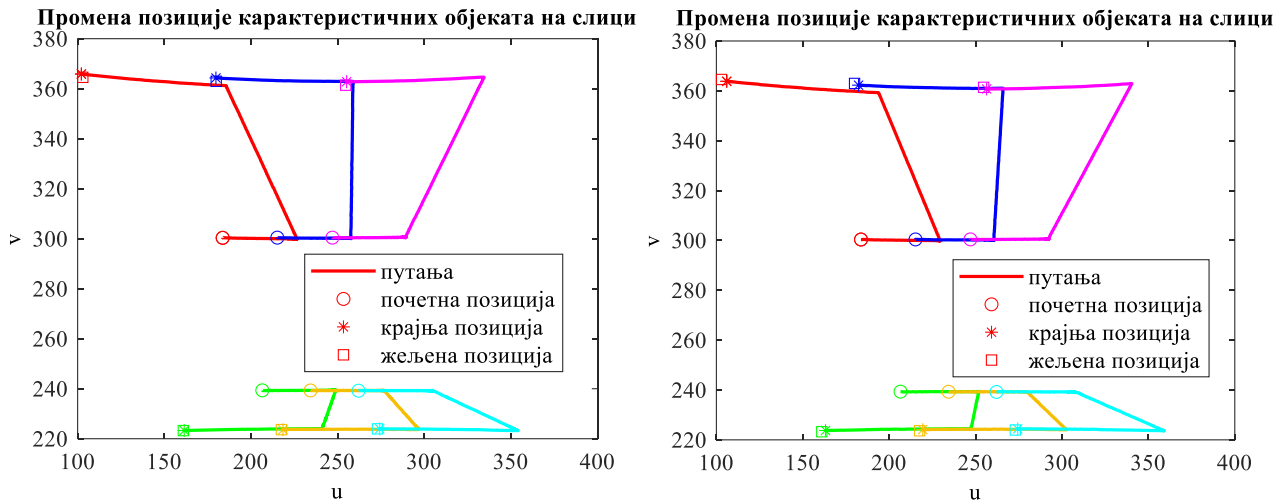
Слика 4.11. PBVS промена грешака по фазама.

Наредно поређење извршено је на основу промене грешке компоненти вектора стања, тј. x и z координата, као и угла оријентације θ (слика 4.12). Као што се може приметити, промена грешке је слична са јасном разликом у тренду промене позиције по z оси, где је код IBVS-а тренд експоненцијалан, а код PBVS-а линеаран. Како PBVS користи грешке карактеристичних објеката у 3D, а имплементиран је регулатор пропорционалног дејства, овакав тренд је и очекиван.



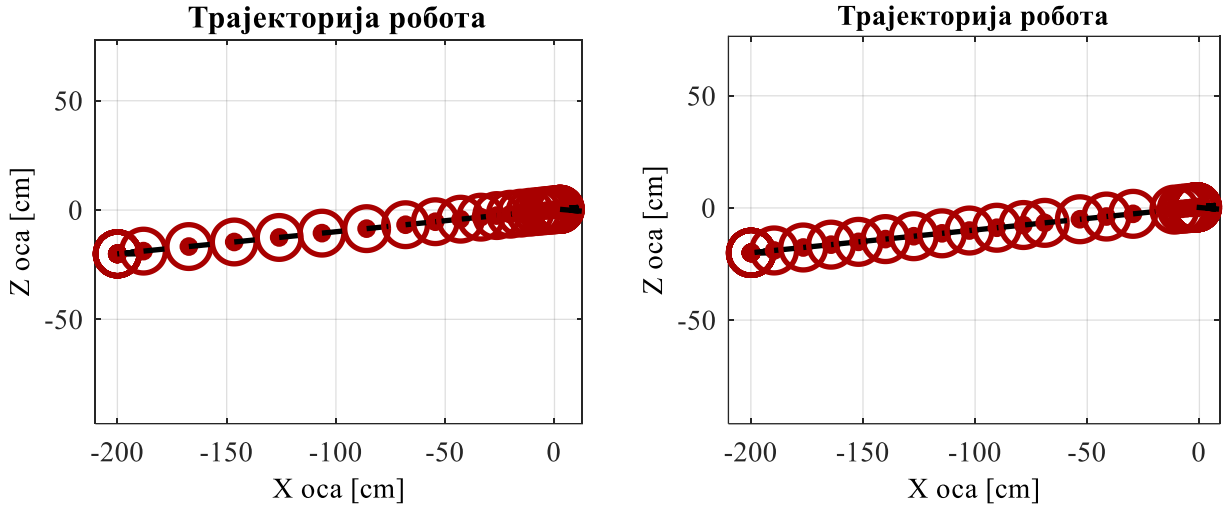
Слика 4.12. Промена положаја мобилног робота применом алгоритама IBVS (лево) и PBVS (десно).

Као што се може уочити на слици 4.13, оба алгоритма успешно мењају позиције свих шест карактеристичних објеката све док оне не достигну жељену вредност на слици са минималном грешком од пар пиксела. Код IBVS-а, разлика остварене и жељене позиције карактеристичних објеката је у правцу v осе, док је код PBVS-а грешка уочљива по u оси, што се касније и директно одражава на остварену финалну грешку положаја мобилног робота.



Слика 4.13. Промена позиције карактеристичних објеката на слици применом алгоритама IBVS (лево) и PBVS (десно).

Трајекторије мобилног робота остварене на основу примене алгоритама за визуелно управљање дате су на слици 4.14. Као што се може приметити, трајекторије су сличне, са примарном разликом у учестаности приказа промене положаја. Како су алгоритми извршили око 600 (IBVS) и 700 (PBVS) итерација управљачке петље, нису сви узастопни положаји приказани, већ су положаји одабрани са одређеном учестаношћу. На крају кретања мобилног робота применом IBVS алгоритма остварен је положај са грешком (1,50 cm 0,79 cm -0,47°), док је на основу PBVS алгоритма остварена грешка положаја од (1,22 cm 0,31 cm -0,14°).



Слика 4.14. Трајекторије мобилног робота остварене применом алгоритама IBVS (лево) и PBVS (десно).

4.3. Стерео визуелно управљање нехолономних мобилних роботских система базирано на семантичкој сегментацији

Као што је већ напоменуто, експериментална верификација предложеног алгоритма стерео визуелног управљања биће извршена на мобилном роботском систему RAICO чији се сензорски подсистем састоји од стерео визуелног система машинског гледања. Када се изврши семантичка сегментација све четири слике (лева тренутна, десна тренутна, лева циљна и десна циљна), а затим и регистрација између леве тренутне и леве циљне слике, десне тренутне и десне циљне и на крају између леве и десне циљне слике, могуће је на основу матрица трансформације добијених као резултат регистрације довести у математичку зависност позиције пиксела између свих слика. Затим, потребно је одабрати камеру која ће бити примарна (у овом случају лева камера) и са тренутне слике те камере одредити технолошки објекат који је од интереса за управљање. На основу пиксела који припадају том објекту, може се издвојити неколико референтних који ће се користити за управљање. Непозната позиција референтних пиксела на свим осталим сликама одређује се на основу трансформационих матрица добијених на основу регистрације слика. Примарни разлог за коришћење само дела пиксела при управљању је време процесирања које је потребно остварити ради управљања у реалном времену. Секундарни разлог произилази из потребе да се трансформација пиксела добијена на основу матрице регистрације не разликује значајно за суседне пикселе па се целокупна трансформација објекта може апроксимирати са неколико адекватно одабраних пиксела. Након тога, потребно је да се помоћу матрица трансформације доведе у математичку зависност позиција референтних пиксела на левој и десној тренутној слици, што се може извршити на основу једначине (4.34):

$$\boldsymbol{c}_{lk} = \boldsymbol{T}_l^{-1}\boldsymbol{T}_d\boldsymbol{T}_r\boldsymbol{c}_{rk}, \quad k = 1,\ldots,K, \tag{4.34}$$

где је $\boldsymbol{T}_l$ матрица трансформације између левих слика, $\boldsymbol{T}_d$ матрица трансформације између леве и десне жељене слике, $\boldsymbol{T}_r$ матрица трансформације између десних слика, док су са $\mathbf{c}_{lk}$ и $\mathbf{c}_{rk}$ дефинисане позиције референтних пиксела на левој и десној слици, док је са K обележен укупан број референтних пиксела. Када су одређене позиције референтних пиксела на све четири слике, потребно је уврстити их у једначину за одређивање Јакобијан матрице слике (4.17), за сваку од камера понаособ. Тиме се одређују брзине у координатном систему леве и десне камере. Међутим, како су камере постављене под одређеним углом инклинације у односу на хоризонталну раван, потребно је извршити трансформацију тих брзина у координатни систем робота (4.35),

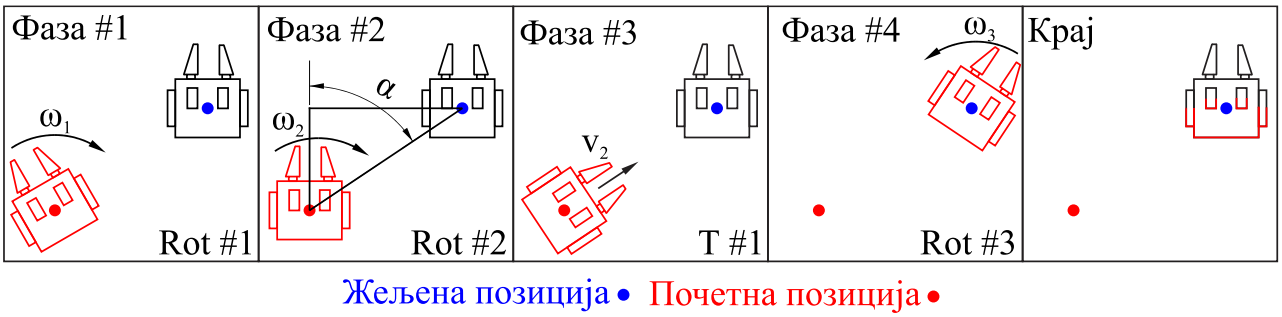
$$\boldsymbol{v}_{mr} = {}^{mr}\boldsymbol{M}_\rho \boldsymbol{v}_{cam,\rho} = \begin{pmatrix} {}^{mr}\boldsymbol{R}_\rho & \left[{}^{r}\boldsymbol{t}_\rho\right]_\times {}^{mr}\boldsymbol{R}_\rho \\ O_3 & {}^{mr}\boldsymbol{R}_\rho \end{pmatrix} \boldsymbol{v}_{cam,\rho}, \quad \rho = \mathrm{l,r}. \tag{4.35}$$

$$\left[{}^{r}\boldsymbol{t}_\rho\right]_\times = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{4.36}$$

где $\boldsymbol{v}_{cam,\rho}$ дефинише брзине у координатним системима камера, ${}^{mr}\boldsymbol{M}_\rho \in \mathbb{R}^{6\times 6}$ је матрица трансформације брзине из координатних система камера у координатни систем мобилног робота, ${}^{mr}\boldsymbol{R}_\rho$ је матрица ротације између координатног система камере и робота, $[{}^{r}\boldsymbol{t}_\rho]_\times$ (4.36) је кососиметрична матрица креирана на основу вектора транслације између координатног система мобилног робота и камере, $\boldsymbol{v}_{mr}=(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^{\mathrm{T}}$ су брзине трансформисане у координатни систем мобилног робота, док ρ дефинише да ли се користи једначина за леву или десну камеру. Веза између брзина мобилног робота и промене позиција референтних пиксела, на основу једначина (4.21) и (4.35), дата је једначином (4.37):

$$\boldsymbol{v}_{mr} = \lambda \begin{pmatrix} \boldsymbol{J}_{l1}\boldsymbol{M}_l \\ \vdots \\ \boldsymbol{J}_{lk}\boldsymbol{M}_l \\ \boldsymbol{J}_{r1}\boldsymbol{M}_r \\ \vdots \\ \boldsymbol{J}_{rk}\boldsymbol{M}_r \end{pmatrix}^{-1} \begin{pmatrix} \dot{\boldsymbol{c}}_{l1} \\ \vdots \\ \dot{\boldsymbol{c}}_{lk} \\ \dot{\boldsymbol{c}}_{r1} \\ \vdots \\ \dot{\boldsymbol{c}}_{rk} \end{pmatrix}. \tag{4.37}$$

У наставку, приказан је алгоритам на основу кога се стерео визуелно управљање имплементира на нехолономни мобилни робот. Како мобилни робот који се креће помоћу точкова има нехолономна ограничења, није могуће директно користити брзине добијене из визуелног управљања ради остваривања жељеног положаја. Из тог разлога, предложен је управљачки алгоритам који се састоји из четири фазе, на основу којих мобилни робот остварује жељени положај, а илустрација кретња мобилног робота на основу овог алгоритма, дата је на слици 4.15:



Слика 4.15. Фазе алгоритма стерео визуелног управљања на бази семантичке сегментације.

Прва фаза подразумева да се мобилни робот ротира све док му оријентација не буде идентична са оном у жељеном положају. У таквој конфигурацији могуће је оценити угао α (слика 4.15), који представља жељену управљачку величину друге фазе. Друга фаза се завршава када тренутни угао оријентације постане исти као угао α. Тада се активира трећа фаза, у оквиру које мобилни робот врши транслаторно кретање све док не оствари жељену позицију. У последњој фази мобилни робот се ротира док не оствари жељени положај. Извођења за управљачки систем сваке фазе дата су у наставку.

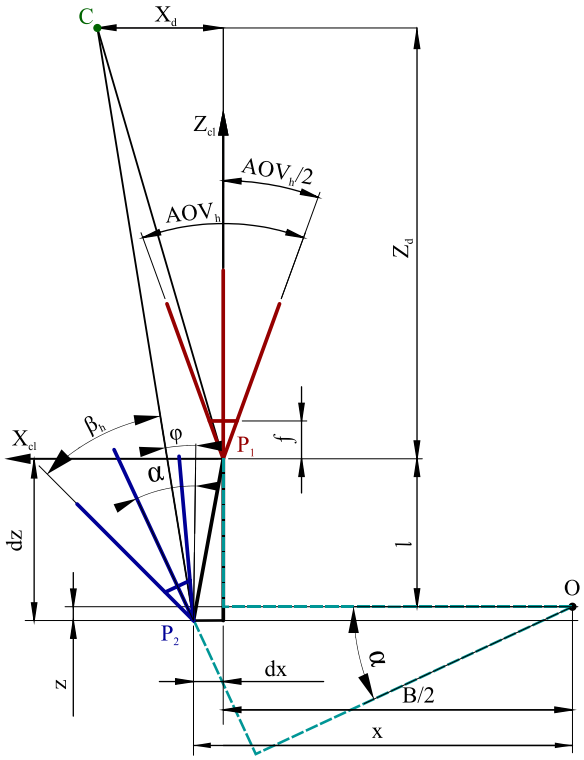
Када се на основу једначине (4.37), односно система стерео визуелног управљања генеришу брзине мобилног робота, угаона брзина ω_y може се употребити за формирање управљачког система прве фазе (4.38). Како се у овој фази врши само ротација, транслаторна брзина неће бити узета у обзир:

$$v_1 = 0; \omega_1 = k_{\omega 1}\omega_y. \tag{4.38}$$

Када се заврши прва фаза, потребно је одредити угао α и отпочети другу фазу. Угао α добија се на основу једначине (4.39), односно коришћењем 3D координата жељених $(X_d, Y_d, Z_d)^T$ и тренутних $(X_c, Y_c, Z_c)^T$ референтних објеката ради естимације угла:

$$\alpha = \operatorname{atan2}\left(X_c - X_d, Z_c - Z_d\right). \tag{4.39}$$

У овој фази алгоритма уочава се потреба за коришћењем стерео визуелног система управљања, јер се на основу њега могу добити 3D координате. Када је израчунат угао α, може се дефинисати управљачки систем за другу фазу, али се првенствено мора оценити ротација референтних објеката у равни слике која би настала када се мобилни робот заротира за угао α, а на основу чега је могуће извршити естимацију позиције референтних пиксела на слици након ротације мобилног робота.



Слика 4.16. Померање координатног система камере у хоризонталној равни услед ротације робота.

Када се мобилни робот RAICO ротира око вертикалне осе за угао α, лева камера ће се померити из положаја P_1 (приказан црвеном бојом на слици 4.16) у положај P_2 (приказан плавом бојом на слици 4.16). Удаљеност камере од центра ротације мобилног робота дуж X, односно Z осе дефинисана је параметрима $B/2$ и l, респективно. Како је позната фокална дужина f, као и 3D позиција референтних објеката, могуће је извршити њихову пројекцију на раван слике у оба положаја камере. Релативна трансформација позиције камере из положаја P_1 у положај P_2 дата је једначином (4.40):

$$\begin{pmatrix} dx \\ dz \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} B/2 \\ l \end{pmatrix} - \begin{pmatrix} B/2 \\ l \end{pmatrix}. \tag{4.40}$$

Након тога, потребно је одредити угао φ (4.41) како би било могуће, применом једначине (4.42), одредити угао β_h, који се користи како би се одредила жељена позиција пројекције референтних објеката на раван слике у положају на крају друге фазе управљачког алгоритма.

$$\varphi = \mathrm{atan2}\left(X_d - dx, \mathrm{Z}_d - dz\right) \tag{4.41}$$

$$\beta_h = \left(AOV_h / 2\right) + \alpha - \varphi \tag{4.42}$$

Коришћењем угла β_h, угла видног поља камере (AOV_h), као и резолуције слике у хоризонталном правцу (N), могуће је директно одредити позицију референтног објекта на слици након ротације (x_d), што представља жељену вредност у управљачком алгоритму, а израчунава се применом једначине (4.43):

$$x_d = \left(N \cdot \beta_h / AOV_h\right) - \delta_h . \tag{4.43}$$

Како место пресека оптичке осе и равни слике није увек у самом центру слике, уведен је додатни параметар δ_h, а који се израчунава једначином (4.44):

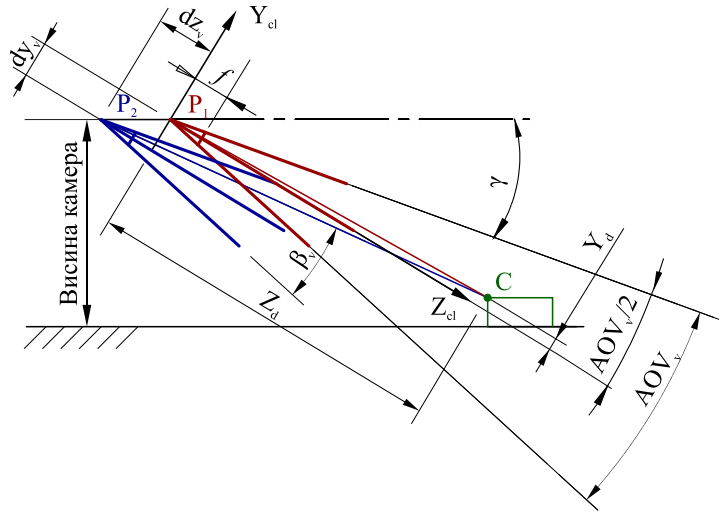
$$\delta_h = PP_{hc} - PP_{ht}, \tag{4.44}$$

где PP_{hc} и PP_{ht} представљају калибрисану и теоријску позицију где оптичка оса пресеца раван слике, респективно. Када је израчуната вредност x_d, може се креирати управљачки алгоритам за другу фазу, дефинисан једначином (4.45):

$$v_2 = 0; \omega_2 = k_{\omega 2}\left(x_d - x_c\right); e_2 = \sqrt{\left(x_d - x_c\right)^2}. \tag{4.45}$$

Грешка која се користи за заустављање извршавања друге фазе је e_2.

Трећа фаза подразумева транслацију до жељене позиције. Развој управљачког алгоритма сличан је као и за другу фазу, а главна разлика је у томе да се дефинисање параметара врши у вертикалној равни (слика 4.17).



Слика 4.17. Померање координатног система камере у вертикалној равни услед ротације робота.

Додатни параметар који утиче на оцену промене позиције референтног објекта у равни слике је угао инклинације камера γ, који представља конструкциони параметар роботског система. У случају мобилног робота RAICO, угао инклинације је константан и износи 20°. Дистанца померања позиције камере у Z правцу (dz), израчунато из једначине (4.40), може се пројектовати на осе камере Y_{cl} и Z_{cl}, чиме се генеришу померања у координатном систему леве камере (4.46).

$$\left(dz_v \quad dy_v\right) = \left(\cos(\gamma) \quad \sin(\gamma)\right) dz. \tag{4.46}$$

Помоћу дистанци dz_v и dy_v, могуће је одредити угао β_v коришћењем једначине (4.47):

$$\beta_v = \mathrm{AOV_v}/2 + \mathrm{atan2}\left(Y_d + dy_v, Z_d + dz_v\right). \tag{4.47}$$

Затим, угао β_v директно се користи у одређивању позиције референтног објекта на слици након ротације мобилног робота применом једначине (4.48):

$$y_d = \left(M \cdot \beta_v / AOV_v\right) - \delta_v \tag{4.48}$$

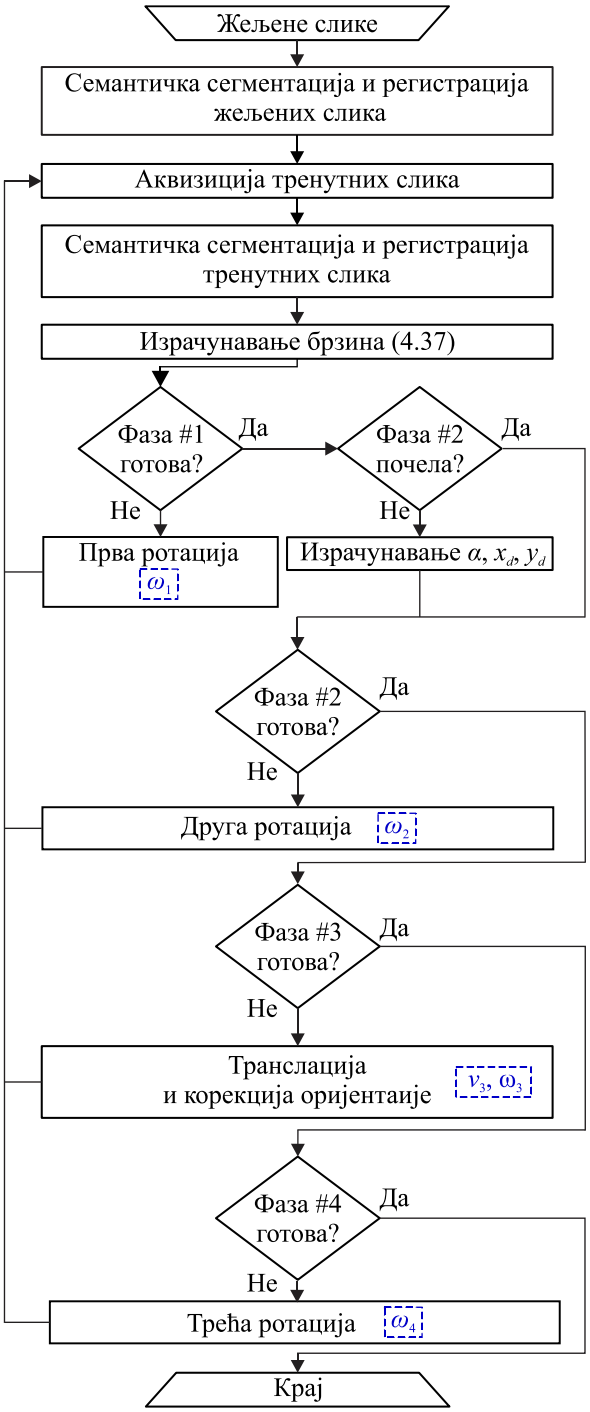
Тиме су дефинисани сви параметри који су потребни ради креирања управљачког алгоритма за трећу фазу (једначина 4.49):

$$v_3 = k_t\left(y_d - y_c\right); \omega_3 = \begin{cases} 0, & \sqrt{\left(x_d - x_c\right)^2} \le R_t \\ k_{\omega 3}\left(x_d - x_c\right), & \sqrt{\left(x_d - x_c\right)^2} > R_t \end{cases}; e_3 = \sqrt{\left(y_d - y_c\right)^2}. \tag{4.49}$$

Управљачки систем треће фазе примењује се док мобилни робот не оствари жељену позицију. Поред транслаторног кретања, уведено је и додатно ротационо кретање уколико дође до повећања грешке из фазе два, које може настати услед шума, проклизавања точкова или разлике у брзинама точкова. Управљачки алгоритам се зауставља када грешка e_3 постане мања од задате вредности прага. Четврта и финална фаза активира се када мобилни робот оствари жељену позицију и потребно је остварити жељену оријентацију. Управљачки алгоритам (4.50) је сличан као и код прве фазе, са другачијим вредностима појачања:

$$v_4 = 0; \omega_4 = k_{\omega 4}\omega_y . \tag{4.50}$$

Комплетан блок дијаграм стерео визуелног управљачког алгоритма на бази семантичке сегментације дат је на слици 4.18.



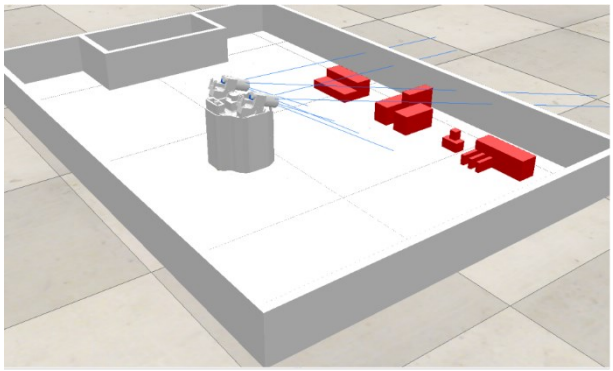
Слика 4.18. Предложени алгоритам стерео визуелног управљања.

У оквиру експерименталне верификације биће извршено укупно пет експеримената са одређеним бројем понављања и другачијим почетним условима, како би се доказала валидност предложеног алгоритма. Прва два експеримента биће изведена у симулационом окружењу, а преостала три ће бити изведена у реалном окружењу са мобилним роботским системом RAICO.

4.3.1. Први експеримент – симулација

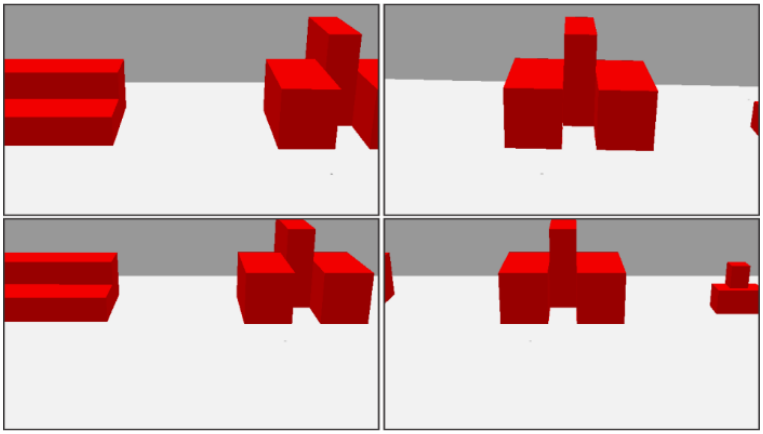
Пре самог тестирања и експерименталне верификације предложеног алгоритма, било је потребно извршити калибрацију стерео визуелног система, као и подешавање параметара алгоритма. Како би се извршила адекватна регистрација циљних слика, уведена је иницијална естимација параметара регистрације. С обзиром на то да је стерео визуелни систем непокретан, увек ће се слика коју генерише лева камера односити на леви део сцене, док ће десна камера опажати десни део сцене, што значи да постоји очекивана разлика у сликама везана за транслацију у X правцу, тако да је уведена иницијална естимација при регистрацији која износи t_x=250. Параметри еволуционог алгоритма коришћеног за регистрацију су: иницијална варијанса износи $1{,}5 \cdot 10^{-7}$, параметар повећања је 1,025 и максимални број итерација је 400. Три нивоа умањења слике су примењена при регистрацији, док је финална трансформација коришћена као иницијална естимација параметара трансформације за наредни ниво.

Прва експериментална верификација изведена је у 3D симулационом окружењу софтверског пакета *CoppeliaSim* који је предвиђен за симулацију роботских система. Мобилни робот RAICO моделиран је по угледу на реални роботски систем са интегрисаним виртуелним камерама и постављен је у 3D окружење (слика 4.19) које представља модел реалног експерименталног технолошког окружења. Камере у симулационом окружењу немају дисторзију, тако да за овај експеримент није било потребно извршити калибрацију. У сцени мобилног робота налазе се четири модела машина алатки различите геометрије. Сваком моделу машина додељена је различита класа, која ће бити генерисана применом семантичке сегментације сцене.



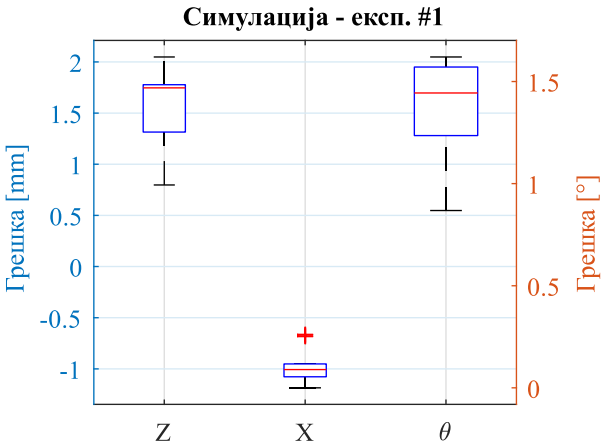
Слика 4.19. Мобилни роботски систем RAICO у симулационом окружењу.

Мобилни роботски систем прво је постављен у циљни положај дефинисан вектором (200 mm, 600 mm, 0°), где су генерисане жељене слике, а затим је постављен у почетни положај (220 mm, 700 mm, 0°) и активиран је стерео визуелни систем управљања на бази семантичке сегментације. Слике генерисане у почетном и циљном положају приказане су на слици 4.20, где су у првом реду представљене слике циља, а у другом слике у почетном положају.



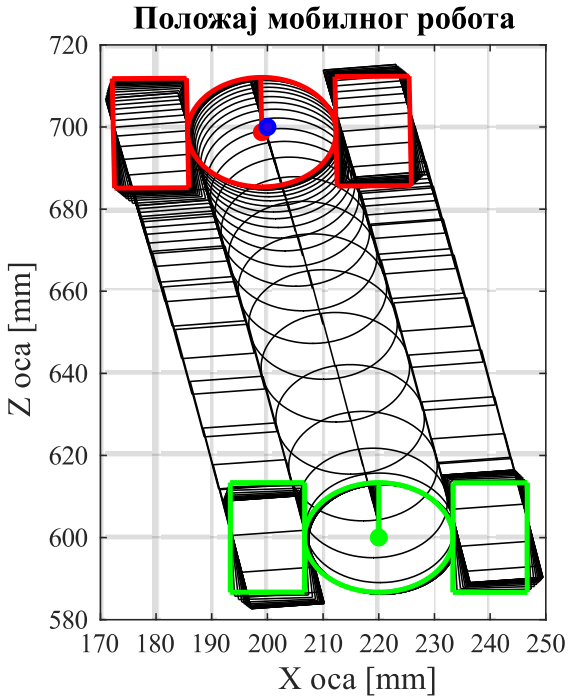
Слика 4.20. Слике генерисане стерео паром у жељеном положају (прва врста) и почетном положају (друга врста).

Експеримент је поновљен 10 пута ради анализе поновљивости резултата, а грешке оствареног положаја могу се уочити на слици 4.21. На левој ординати могу се очитати грешке позиције (у Z и X правцу), док се на десној ординати може видети грешка оријентације. Медијана остварене грешка позиције у X правцу износи око 1 mm, док је најмања грешка износила око 0,6 mm. Грешка позиције по Z оси има већу вредност и њена медијана износи око 1,75 mm. Такође, стандардна девијација грешке по Z оси има већу вредност, а тиме је и поновљивост по овој координати значајно мања. На крају, медијана грешке оријентације износи око 1,5°, са оствареним распоном грешке од око 0,8°. Анализом дијаграма са слике 4.21, може се закључити да је резултат симулације изузетно задовољавајући са високим нивоом остварене тачности и поновљивости.



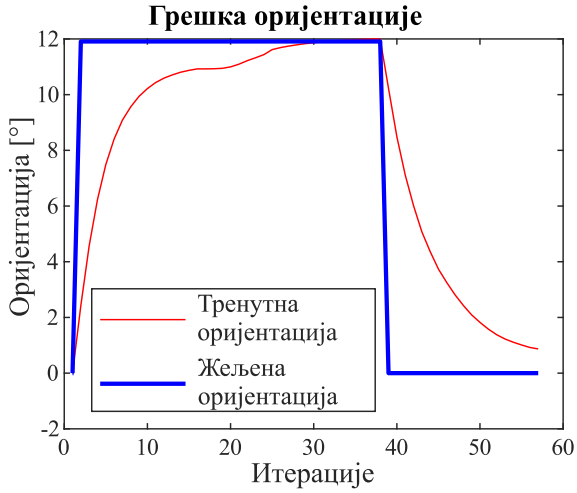
Слика 4.21. Резултати првог експеримента у симулацији.

Путања коју је остварио мобилни робот током најбољег понављања анализираног експеримента приказана је на слици 4.22. Зеленом контуром приказан је почетни положај, црном међуположаји остварени током кретања, црвеном финални положај, док је плавим кругом означена жељена позиција.



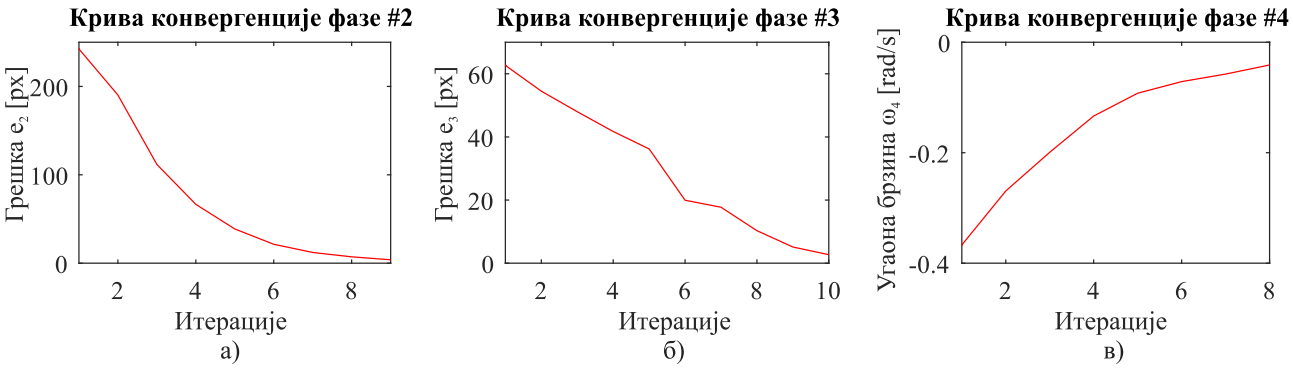
Слика 4.22. Путања коју је остварио мобилни робот за најбоље понављање првог експеримента у симулацији.

За мобилни робот са диференцијалним погоном и предложеним алгоритмом стерео визуелног управљања, финална позиција зависи значајно од тачности оријентације током кретања. Из овог разлога, анализирана је и приказана промена оријентације на слици 4.23, као и жељене вредности оријентације коју мобилни робот треба да оствари.



Слика 4.23. Промена оријентације мобилног робота током визуелног управљања.

Како би се анализирале перформансе фаза предложеног алгоритма, на слици 4.24 приказане су криве конвергенције грешке или угаоне брзине за све фазе управљачког алгоритма. Оријентација мобилног робота у првој фази приближне је вредности као и жељена оријентација, тако да није било потребно извршити прву фазу алгоритма. За фазе два и три, приказана је конвергенција грешке дефинисане у равни слике, тј. у пикселима, док је крива конвергенције фазе четири дефинисана интензитетом угаоне брзине. Крива конвергенције фазе два има експоненцијални тренд, што је повољно за управљачки систем. Код фазе три може се уочити да постоје сегменти криве где се нагло мења нагиб, што настаје због компензације оријентације (једначине 4.49) при кретању мобилног робота. Крива конвергенције фазе четири такође има повољан експоненцијални карактер са мањим (неповољним) променама брзине, које настају услед шума у сегментацији и регистрацији слика.



Слика 4.24. Криве конвергенције грешака и угаоне брзине, током различитих фаза реализације алгоритма стерео визуелног управљања.

4.3.2. Други експеримент – симулација

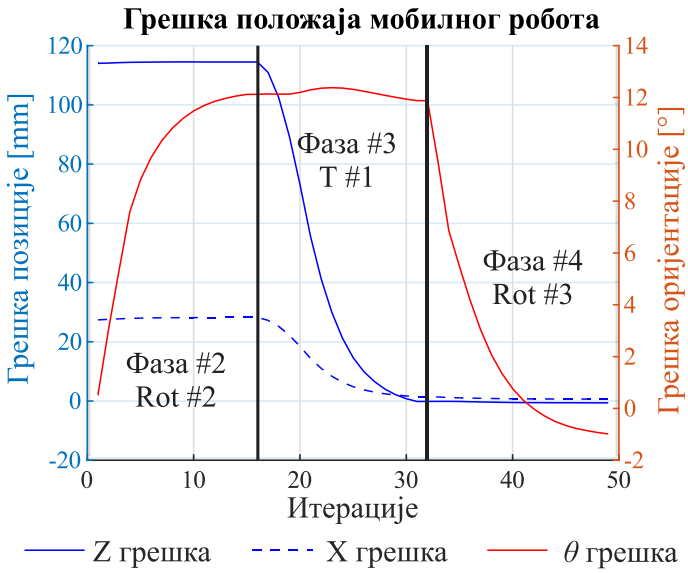
Други експеримент изведен у симулацији састоји се од 10 понављања алгоритма визуелног управљања, али са случајно генерисаним почетним положајима мобилног робота. Мобилни робот је постављен у нови жељени положај (600 mm, 500 mm, 0°), где су генерисане и сачуване жељене слике. Затим је дефинисан иницијални положај (600 mm, 600 mm, 0°) који је представљао очекивану вредност при генерисању случајних нових стартних положаја. Стандардна девијација при генерисању случајних положаја износила је (2, 1, 3) за сваки елемент вектора положаја. Највећа вредност стандардне девијације задата је за оријентацију, тј. за угао θ, јер је у првом експерименту почетна оријентација била идентична као и циљна. Стандардна девијација по X оси је већа јер постоји иницијална разлика између очекиване вредности почетног и жељеног положаја по Z оси. Такође, стандардне девијације су дефинисане тако да се у свим случајно генерисаним положајима виде машине са жељених слика. Случајно генерисани почетни положаји, као и грешка остварена након управљања предложеним алгоритмом, приказани су у табели 4.4.

Табела 4.4. Резултати другог експеримента у симулацији.

Ред. бр.	Почетни положај			Остварена грешка		
	X [mm]	Z [mm]	θ [°]	X [mm]	Z [mm]	θ [°]
1.	617,77	586,50	–1,85	–5,16	1,98	**0,22**
2.	577,06	630,35	–2,24	1,36	–8,28	–1,67
3.	578,62	607,25	–0,58	4,04	–5,34	–1,68
4.	583,81	599,37	2,67	–15,32	3,61	–1,51
5.	561,11	607,15	–2,29	**–0,08**	–16,66	–1,66
6.	628,77	597,95	–4,21	–3,86	6,91	–0,35
7.	606,50	598,76	–4,27	0,56	–1,11	–0,73
8.	584,90	614,90	1,46	–14,03	3,45	–1,63
9.	627,41	614,09	–0,53	**0,65**	**–0,60**	**–0,98**
10.	565,77	614,17	–0,59	5,19	–3,17	–1,86

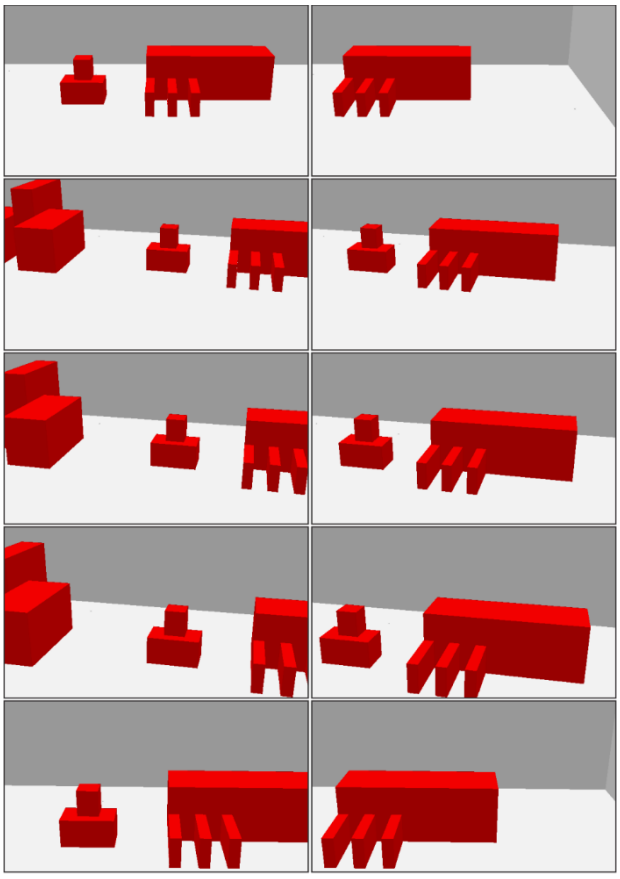
Најбоље понављање другог експеримента у симулацији је под редним бројем девет. Такође, минималне индивидуалне грешке у табели 4.4 приказане су подвученим вредностима. Резултати остварени у другом експерименту знатно су лошији од оних из првог експеримента, међутим, то је и очекивано јер су сви параметри управљачког алгоритма подешени за први експеримент. Током другог експеримента, уочен је недостатак предложеног алгоритма у виду нетачне естимације угла α на крају фазе један. Када је угао α нетачно оцењен, финални положај неће имати очекивану тачност. Међутим, у реалним условима могуће је извршити поновно покретање алгоритма и тиме минимизирати грешку остварену на крају кретања. Даља анализа овог експеримента дата је на примеру деветог понављања. Тачна итерација у којој се мења фаза

алгоритма, као и промена грешака позиције и оријентације у тим фазама, приказане су на слици 4.25.



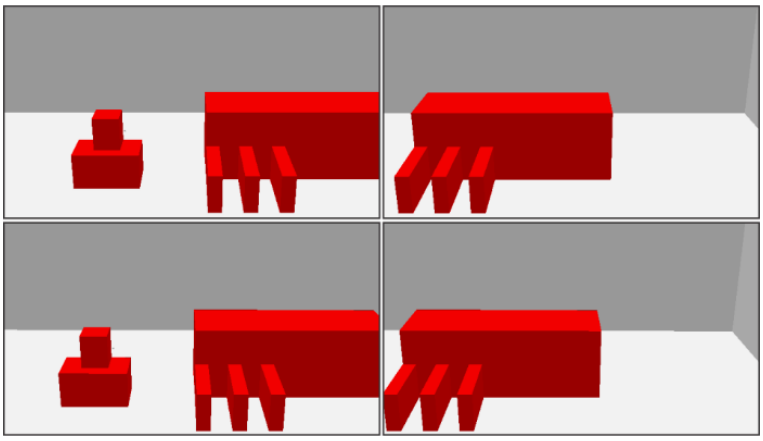
Слика 4.25. Грешка компоненти вектора стања мобилног робота током визуелног управљања.

Затим, на слици 4.26 приказане су слике генерисане стерео визуелним системом мобилног робота током визуелног управљања. У левој колони налазе се слике генерисане левом камером, док се у десној налазе слике добијене са десне камере. У првој врсти су слике генерисане у иницијалном положају, а затим је приказан сваки десети стерео пар слика и на крају су дате слике генерисане у оствареном положају.



Слика 4.26. Слике генерисане стерео визуелним системом током визуелног управљања.

Тачност оствареног положаја на крају симулације може се визуелно оценити на основу разлике жељених слика и слика остварених у крајњем положају (слика 4.27).



Слика 4.27. Слике генерисане у крајњем (горе) и жељеном (доле) положају.

Као што се може приметити са слике 4.27, тачност по Z оси је изузетно висока, јер су машине на сликама у оба положаја на сличној позицији по висини слике. Међутим, грешка позиције по X оси, као и грешка оријентације, резултују разликом између слика у хоризонталном правцу.

4.3.3. Први експеримент у реалном окружењу

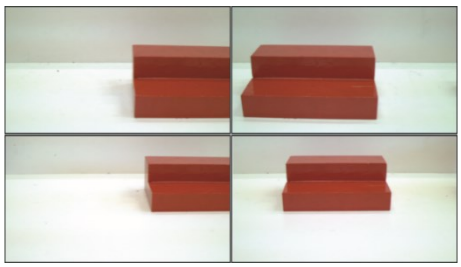
Експериментална верификација предложеног алгоритма спроведена је са иницијалном конфигурацијом стерео визуелног система која је подразумевала две *Basler acA1920-25uc* камере и сочива *Fujinon DF6HA-1B* и системом перцепције предложеним у [126]. Фиксни параметри везани за мобилни роботски систем RAICO су: осно растојање камера износи B = 130 mm, удаљеност између осе ротације мобилног робота и камера у Z правцу је l = 27,5 mm. Интеграцијом поменутих камера и сочива остварено је видно поље камера представљено хоризонталним и вертикалним углом видног поља сваке камере од AOV_h = 38,8° и AOV_v = 22,4°, респективно. Угао инклинације износио је γ=20°, док је резолуција генерисаних слика била 640×360 px. Након процеса калибрације, остварене су следеће матрице унутрашњих (4.51) и спољашњих (4.52) параметара стерео система:

$$\boldsymbol{K}_l = \begin{bmatrix} 926,23 & 0 & 308,50 \\ 0 & 924,12 & 170,87 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{K}_r = \begin{bmatrix} 924,57 & 0 & 322,84 \\ 0 & 924,82 & 162,60 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.51}$$

$$\boldsymbol{T}_{cl} = \begin{bmatrix} 0,999 & -0,006 & 0,017 & -131,07 \\ 0,006 & 0,999 & -0,009 & 0,22 \\ -0,017 & 0,01 & 0,999 & -4,96 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.52}$$

где су $\mathbf{K}_l$ и $\mathbf{K}_r$ лева и десна унутрашња калибрациона матрица, док је $\mathbf{T}_{cl} \in SE(3)$ спољашња матрица која доводи у математичку зависност координатни систем леве и десне камере. Вредност параметра појачања за визуелно управљање износи λ=10 (једначина 4.37), док за сваку од четири фазе вредности појачања износе: $k_{\omega 1}$=0,2; $k_{\omega 2}$=0,00014; k_t=0,4; $k_{\omega 3}$=0,0005 и $k_{\omega 4}$=0,1.

Слике генерисане у жељеном положају (0 mm, 0 mm, 0°) и почетном положају (20 mm, –100 mm, 0°) приказане су на слици 4.28. Експеримент је поновљен пет пута како би, поред тачности, могла да се одреди и поновљивост остваривања финалног положаја.



Слика 4.28. Слике стерео пара генерисане у жељеном положају (прва врста) и почетном положају (друга врста) за први експеримент у реалном окружењу.

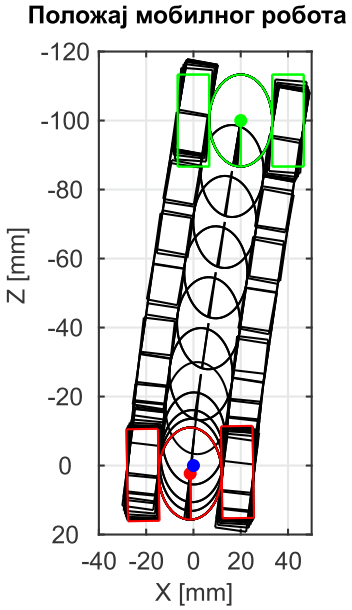
Грешке положаја које мобилни роботски систем RAICO оствари након сваког понављања експеримента приказане су у табели 4.5.

Табела 4.5. Грешке положаја при првом експерименту.

Понављање	Грешке положаја			Тачност
	X [mm]	Z [mm]	θ [°]	
1	2	–11	3	0,9180
2	0	9	5	0,9133
3	**–2**	**4**	**2**	**0,9618**
4	–5	6	3	0,9145
5	–3	5	1	0,9375

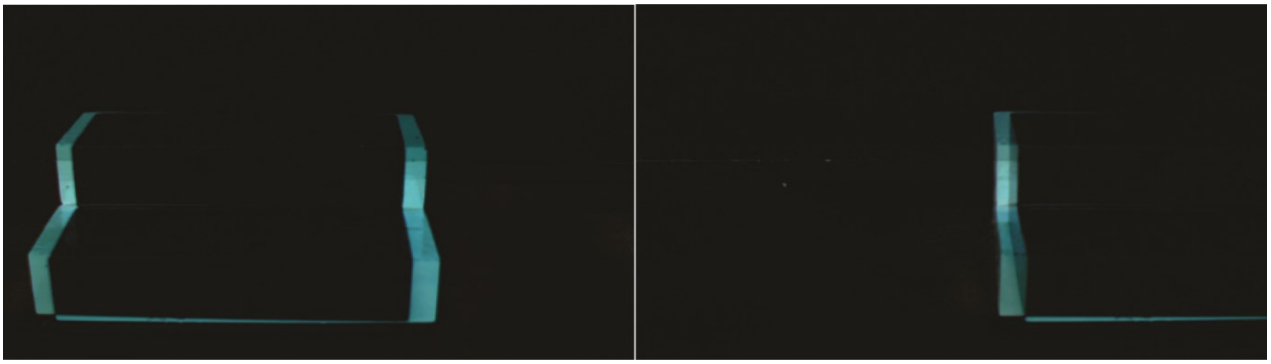
Као што се може уочити из табеле 4.5, грешке по X оси и грешке оријентације су изузетно мале, док су грешке по Z оси знатно веће. Такође, опсег грешака по Z оси је од –11 mm до 9 mm, што истиче ниску поновљивост остваривања позиције по овој оси. Један од кључних разлога за поменуте грешке односи се на стохастичку природу еволуционог алгоритма који је примењен за регистрацију слика. Када се на основу регистрације остваре другачији параметри матрице **T**, у оквиру алгоритма стерео визуелног управљања долази до генерисања различитих интензитета брзина, а самим тим и другачијег положаја. Поред грешке положаја, приказана је и тачност семантичке сегментације израчуната на основу крајњих и жељених слика.

При трећем понављању експеримента остварена је највећа тачност положаја, као и највећа тачност сегментације па је то понављање коришћено за наредне анализе експеримента. Путања којом се мобилни робот кретао током трећег понављања приказана је на слици 4.29. Како се управљање врши на основу визуелних информација, примењен је стандардни модел кретања на основу пређеног пута [117] за одређивање путање којом се мобилни робот кретао.



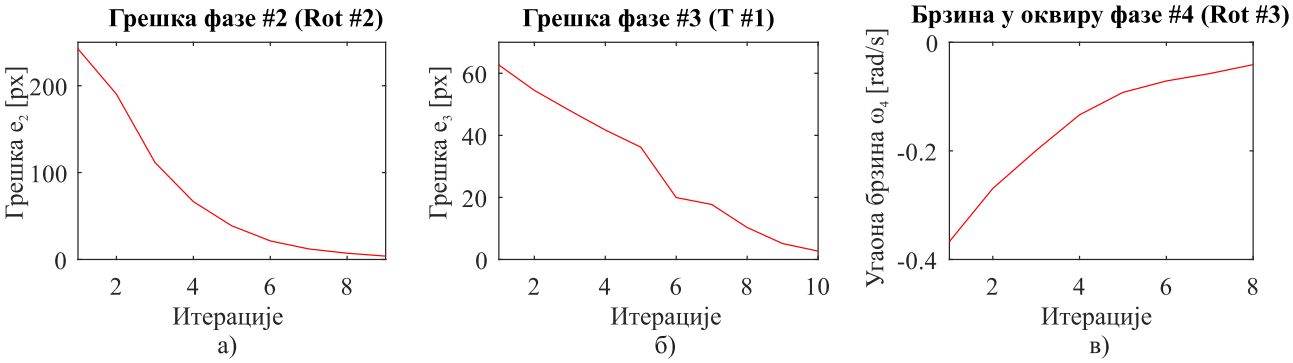
Слика 4.29. Путања мобилног робота у најбољем понављању првог експеримента.

Наредни показатељ тачности оствареног положаја је апсолутна разлика пиксела жељених и остварених слика (слика 4.30). Црном бојом приказани су пиксели код којих апсолутна разлика не постоји, док је су плавом бојом приказани пиксели где постоји разлика. Као што се може уочити, модели машина на две слике су померени у хоризонталном правцу, те се може видети да постоји разлика између оствареног и жељеног положаја, док је разлика у вертикалном правцу слике минимална.



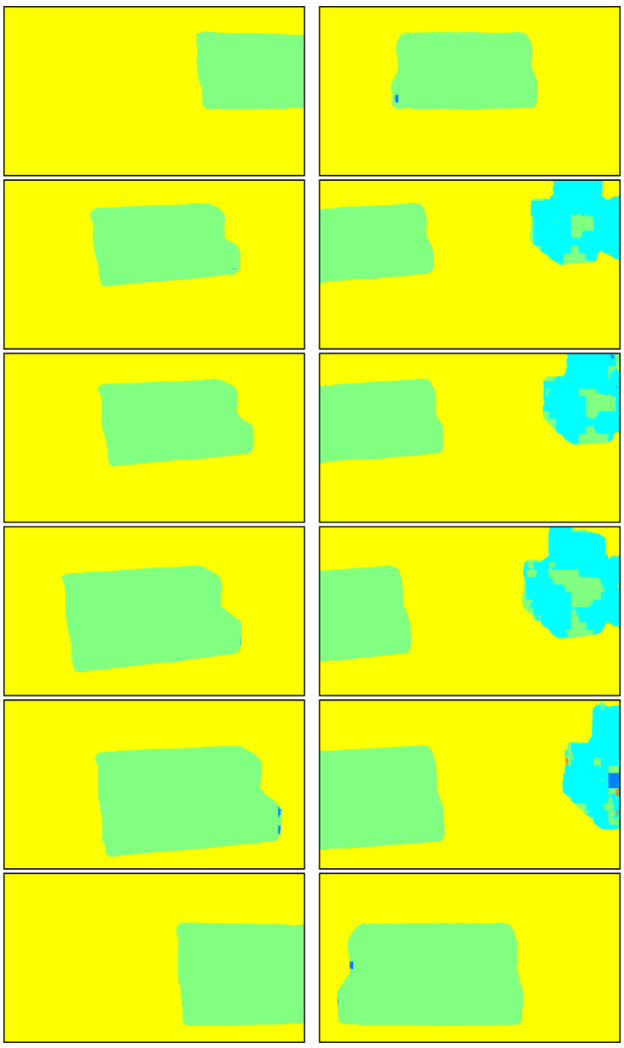
Слика 4.30. Слике апсолутне разлике интензитета пиксела између жељених и крајњих слика.

Конвергенције грешака и брзина током трећег понављања првог експеримента са мобилним роботским системом RAICO приказане су на слици 4.31. Како је разлика између иницијалне и жељене оријентације мала, прва фаза није била неопходна. Током друге фазе, тренд конвергенције грешке је експоненцијалан, што представља повољан тренд ка коме се тежи при креирању визуелних система управљања. У трећој фази, грешка опада линеарно са малим скоком у шестој итерацији. Овакав тренд није повољан, међутим, повећањем максималних брзина које мобилни робот остварује, могло се додатно утицати на побољшање тренда конвергенције. Такође, нагли пад грешке у шестој итерацији проузрокован је додатном ротацијом коју мобилни робот може да изврши у фази три, како би кориговао тренутну оријентацију. На крају, тренд конвергенције брзине у четвртој фази је такође изразито повољан.



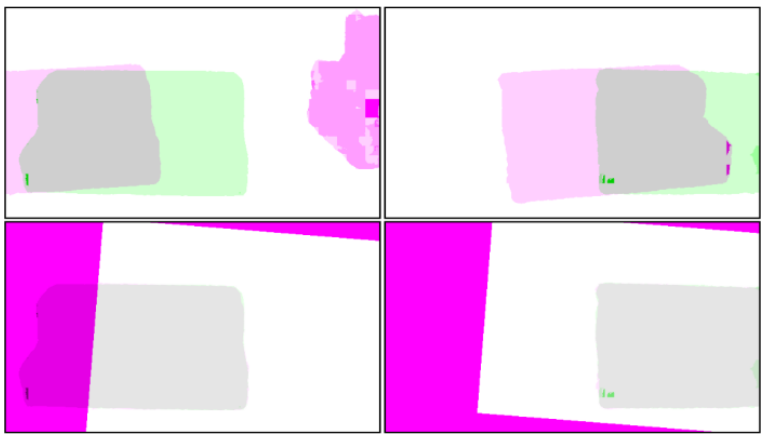
Слика 4.31. Криве конвергенције грешке и угаоне брзине за различите фазе алгоритма визуелног управљања.

Семантичке мапе слика генерисаних током процеса визуелног управљања приказане су на слици 4.32. Као што се може уочити, током сегментације лева машина је препозната са високом тачношћу, док друга машина приказана на десним сликама нема високу тачност сегментације. Међутим, као што је већ приказано, визуелно управљање је извршено адекватно чак и са поменутим грешкама у сегментацији, што доводи до закључка да је предложени алгоритам робустан на мале грешке у сегментацији сцене.



Слика 4.32. Семантичке мапе слика генерисаних током првог експеримента у реалном окружењу.

Наредни анализирани део експеримента је процес регистрације слика. Тачност регистрације директно се одражава на генерисање интензитета брзина, што утиче на тачност самог визуелног управљања и оствареног положаја мобилног робота. Тачна регистрација подразумева да се машине, као и позадина тачно подударају на сликама на којима се регистрација врши. Семантичке информације о класи и позицији модела машина алатки користе се у процесу регистрације. На слици 4.33 приказане су преклопљене слике, пре (прва врста) и после (друга врста) регистрације. Машина на тренутној слици приказана је розе бојом, док је машина на жељеној слици представљена зеленом бојом. Када се адекватно подударају, оне су представљене сивом бојом. Адекватно преклопљена (регистрована) позадина је приказана белом бојом, док је љубичастом бојом приказана област где уопште нема преклапања слика (јер је једна слика померена током трансформације). Као што се може уочити, процес регистрације је тачан и машине, као и позадина, се подударају у великој мери.



Слика 4.33. Разлика тренутних и жељених семантичких мапа пре и после процеса регистрације.

На крају, изведено је поређење тачности предложеног стерео визуелног алгоритма управљања на бази семантичке сегментације са неколико стандардних визуелних система управљања. Резултати су приказани у оквиру табеле 4.6, а у циљу поређења коришћено је треће понављање првог експеримента. Сви експерименти су изведени под истим следећим условима. Поређени су алгоритми који су имплементирани на нехолономним роботским системима са диференцијалним погоном, развијеним у сврхе истраживања. Експерименти су изведени у лабораторијским условима где постоји комбинација природног и вештачког осветљења сцене. Иницијални положај мобилних робота је дефинисан на уобичајен начин, где је сцена на жељеним сликама видљива па нису били потребни иницијални маневри пре визуелног управљања. Путање које су остварили мобилни роботи су такође биле сличне. На крају, брзине које су прослеђене мобилним роботима генерисане су искључиво на основу прикупљених визуелних информација. Како је код анализираних метода извршен различит број понављања експеримента, из сваког рада је одабрано најбоље понављање и искоришћено за поређење.

Табела 4.6. Поређење предложеног алгоритма са методама из литературе.

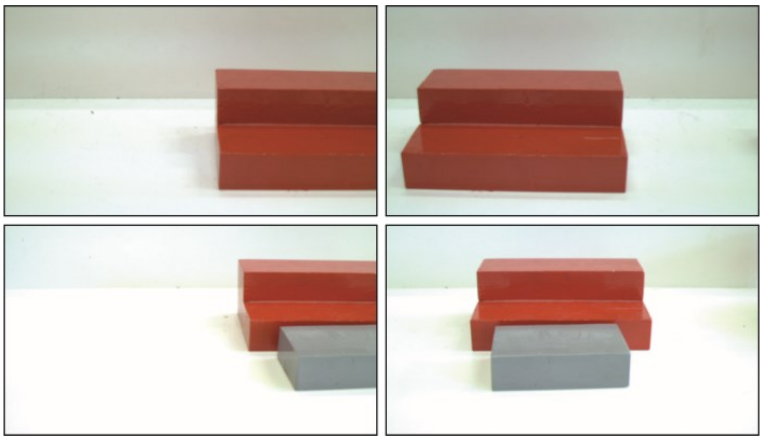
Алгоритам	Грешка положаја		
	X [mm]	Z [mm]	θ [°]
Стандардни стерео IBVS [127]	17	2	/
Визуелно управљање без калибрације камере [89]	17	1	–2.9
Визуелно управљање на бази хомографије [86]	30+	30+	/
Визуелно управљање на бази епиполова [85]	80	110	≈0
Предложени алгоритам на бази семантичке сегментације	–2	4	2

Разлог остваривања веће тачности предложеног алгоритма огледа се у већој количини података који се користе приликом визуелног управљања, као и степену њихове семантичке значајности. Наиме, код осталих алгоритама, позиције одабраних карактеристичних објеката на слици користе се за управљање. За разлику од тих алгоритама, сви подаци везани за позицију и класу објеката у сцени користе се код алгоритма предложеног у оквиру предметне докторске дисертације. Ови подаци садрже много већу количину информација о сцени, тј. окружењу у коме се мобилни робот налази, што резултира високом тачношћу финалног положаја.

4.3.4. Други експеримент у реалном окружењу

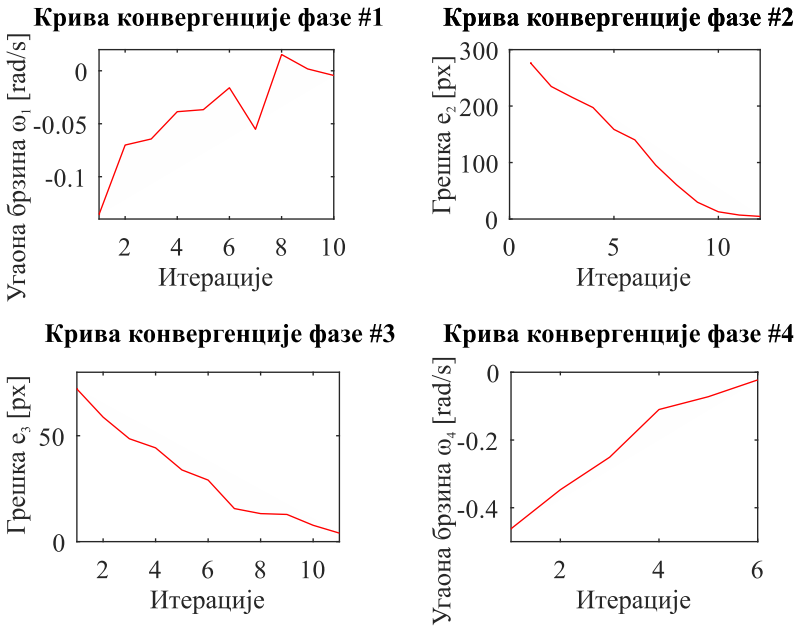
Други експеримент изведен је ради доказивања робустности предложеног алгоритма на парцијалну заклоњеност сцене (што представља поремећајни фактор). Након генерисања жељених слика, на сцену је додат објекат који заклања око 10% сцене. Додатни објекат постављен је тако да заклања део позадине и део машине алатке. У исто време, у овом експерименту анализирана је робустност на промену осветљења сцене (што такође представља

поремећајни фактор). Као што се може уочити са слике 4.34, постоји значајна разлика у осветљености сцене на жељеним (прва врста) и тренутним (друга врста) сликама.



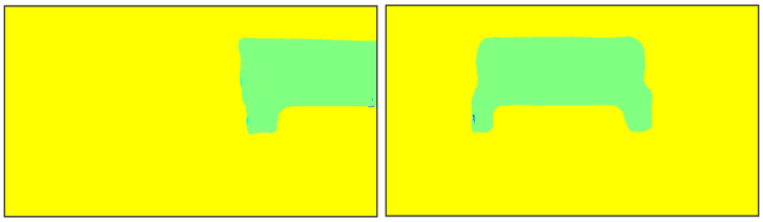
Слика 4.34. Слике у жељеном и почетном положају за други експеримент у реалном окружењу.

У другом експерименту, мобилни робот извршио је све четири фазе предложеног алгоритма. Криве конвергенције све четири фазе приказане су на слици 4.35. Као што се може уочити за прву фазу, заклоњеност сцене, као и промена осветљења, негативно утичу на одређивање брзине па се зато у оквиру седме итерације, када се мобилни робот приближи жељеној оријентацији, нетачна регистрација негативно одражава тако да долази до смањења угаоне брзине. Након тог смањења, у осмој итерацији, може се уочити да мобилни робот компензује насталу грешку тако што изврши ротацију којом долази до прекорачења жељене оријентације. Након тога, ротацијом у супротном смеру брзина конвергира ка нултој жељеној вредности. Уведени поремећајни фактори експеримента негативно утичу и на тачност управљачког алгоритма у оквиру друге фазе, што се одражава линеарним трендом конвергенције грешке. Негативан утицај на другу фазу је знатно мањи, али генерално успорава цео процес визуелног управљања. Трећа и четврта фаза су робустне на заклоњеност сцене и промену нивоа осветљења, јер су њихови трендови конвергенције сличног карактера као и код првог експеримента.



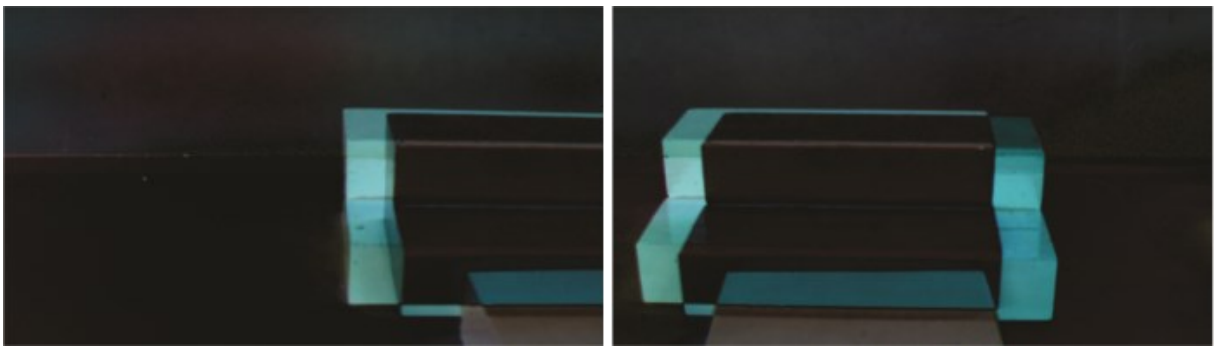
Слика 4.35. Криве конвергенције грешака и угаоних брзина свих фаза стерео визуелног алгоритма генерисане током другог експеримента у реалном окружењу.

Ради додатне анализе, приказан је изглед семантичких мапа почетних слика (слика 4.36). Као што се може уочити, додатни објекат сврстан је у класу позадине, а семантичке мапе са додатним објектом нису у значајној мери утицале на тачност процеса регистрације слика.



Слика 4.36. Семантичке мапе почетних слика у другом експерименту.

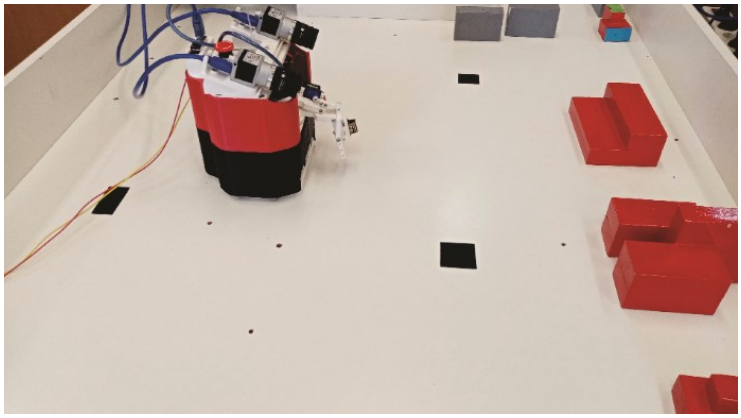
Грешка финалног положаја може се оценити и на основу апсолутне вредности разлике слика генерисане на основу жељених и финалних слика (слика 4.37). У поређењу са сликом 4.30 из првог експеримента, уочава се да је финални положај ниже тачности, али да се не разликује значајно од експеримента без поремећајних фактора.



Слика 4.37. Слике апсолутне разлике интензитета пиксела између жељених и крајњих слика у другом експерименту.

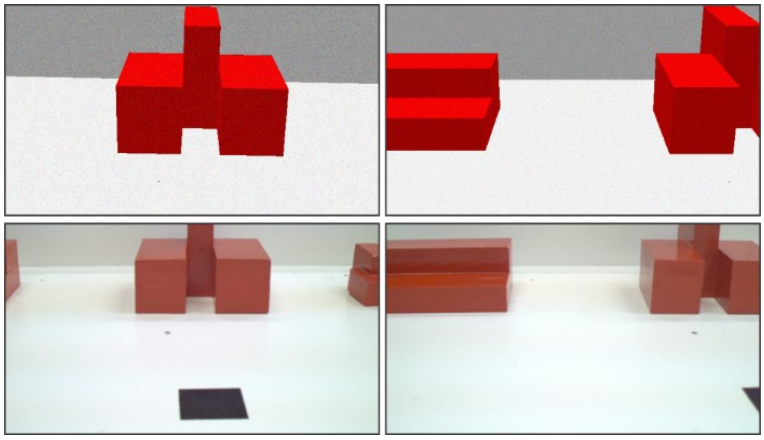
4.3.5. Трећи експеримент у реалном окружењу

Један од недостатака визуелног управљања је потреба за генерисањем циљних слика. Роботски систем потребно је иницијално поставити у жељени положај како би се генерисала циљна слика, а поменути недостатак доводи до мање флексибилности при коришћењу алгоритама визуелног управљања јер се за сваки нови циљ робот мора поставити у жељени положај и генерисати нове циљне слике. У оквиру трећег експеримента, тестирана је хипотеза да је могуће вршити визуелно управљање на основу слика генерисаних у виртуелном окружењу. Наиме, уколико постоји виртуелно технолошко окружење у коме је робот моделиран, могуће је ефикасно („*offline*“– без робота) генерисати слике циља у било ком положају. Почетни положај мобилног робота приказан је на слици 4.38.



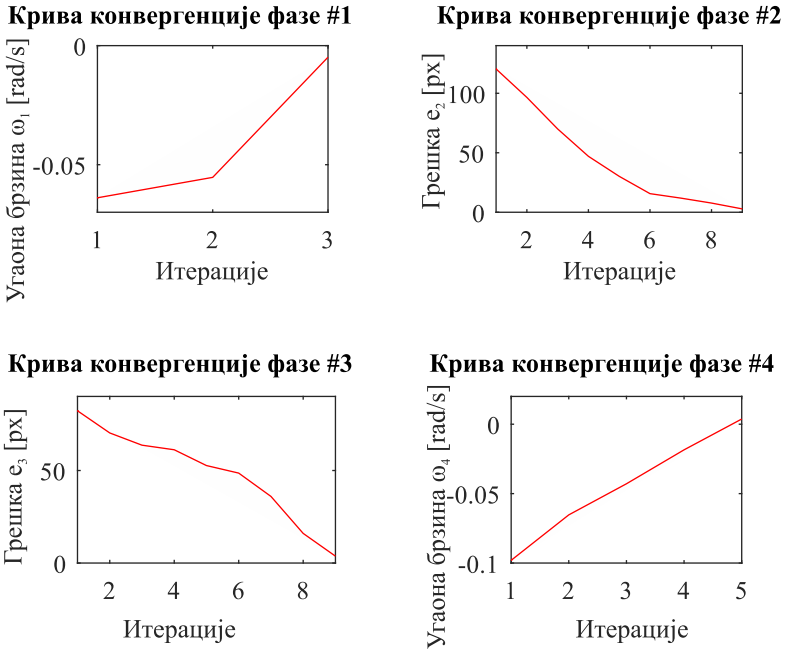
Слика 4.38. Почетни положај мобилног роботског система RAICO у трећем експерименту.

Циљне слике генерисане су у виртуелном окружењу, које је коришћено у прва два експеримента у симулацији. Циљне слике (из симулације) и тренутне слике (генерисане камерама роботског система) приказане су на слици 4.39. Како би процес семантичке сегментације био изведен са довољним степеном тачности помоћу вештачких неуронских мрежа, било је потребно на слике генерисане у симулацији додати вештачки шум, како би слике биле сличније реалним.



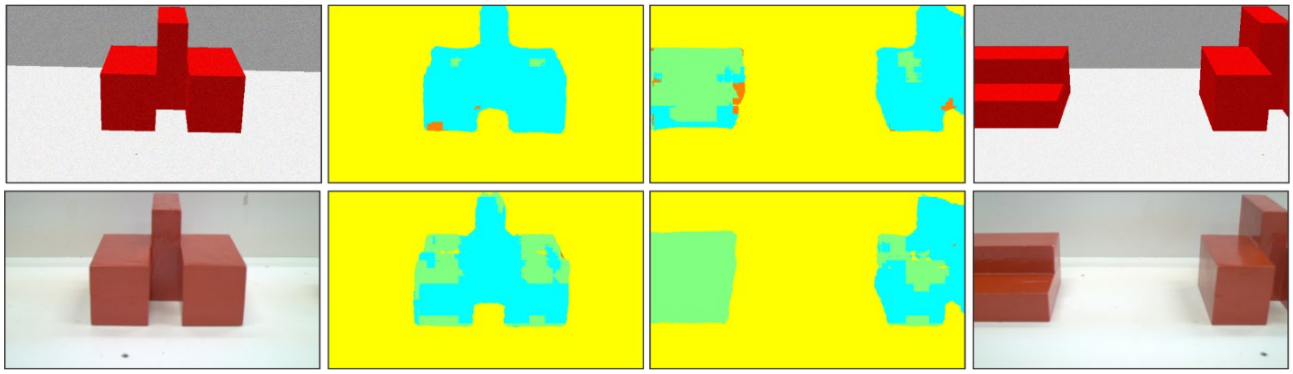
Слика 4.39. Жељене слике у симулацији (прва врста)
и слике генерисане у почетном положају (друга врста).

Као и у претходним експериментима, изведена је анализа кривих конвергенције (слика 4.40) за сваку фазу предложеног алгоритма визуелног управљања. Разлика у односу на прва два експеримента односи се на трећу фазу у оквиру које крива конвергенције има непожељан тренд. Овакав тренд додатно успорава извршавање треће фазе, а самим тим и целокупни алгоритам визуелног управљања. Разлог настанка овакве криве конвергенције односи се на потребу за корекцијом оријентације мобилног робота током треће фазе. Због инхерентних разлика између слика генерисаних у реалном и симулационом окружењу, процес сегментације, као и регистрације, није изведен за тачношћу као и у прва два експеримента па самим тим и роботски систем не показује исти ниво тачности. Остале фазе визуелног управљања су слично извршене као и код прва два експеримента.



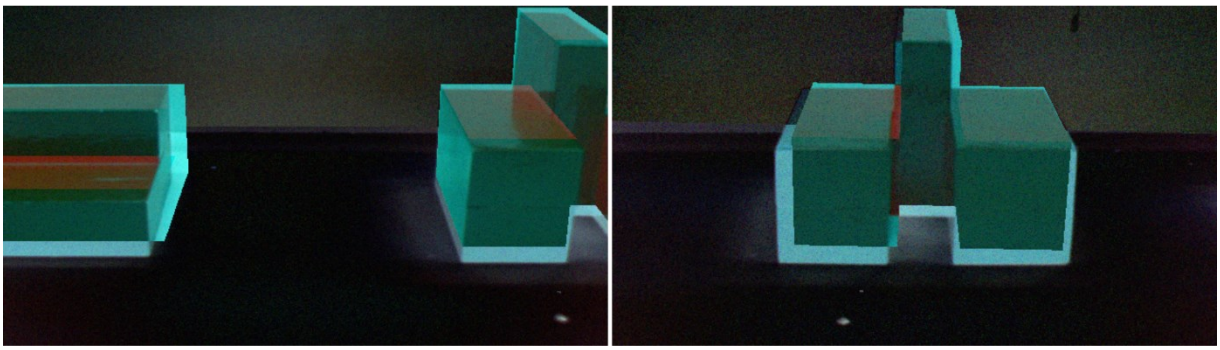
Слика 4.40. Криве конвергенције грешака и угаоних брзина свих фаза стерео визуелног алгоритма генерисане током трећег експеримента у реалном окружењу.

Ради оцене тачности положаја мобилног робота, приказане су финалне и жељене слике, као и њихове семантичке мапе на слици 4.41. Као што се може приметити, разлика између жељених и финалних слика је минимална иако процес семантичке сегментације на сликама генерисаним у симулацији има мањи ниво тачности. Такође, мора се напоменути да постоје разлике између виртуелних и реалних камера.



Слика 4.41. Жељене слике генерисане у симулацији и њихове семантичке мапе (прва врста) и слике у финалном положају и њихове семантичке мапе (друга врста).

На крају, представљена је слика апсолутне разлике између финалних и жељених слика. Због разлике између слика генерисаних у виртуелном и реалном окружењу, тачност на слици 4.42 је мања него у прва два експеримента (слике 4.30 и 4.37). Међутим, грешка финалног положаја указује искључиво на грешке приказане светло плавим пикселима, док је све остало настало као резултат разлика слика генерисаних у симулацији. Може се уочити већа разлика по вертикалној оси слике него у претходним експериментима, што је и очекивано на основу криве конвергенције за трећу фазу.



Слика 4.42. Слике апсолутне разлике интензитета пиксела између жељених и крајњих слика у трећем експерименту.

5. Терминирање мобилних робота у оквиру интелигентних технолошких система

Остваривање поузданог, ефикасног и оптимално терминираног система за унутрашњи транспорт материјала представља један од кључних праваца развоја у оквиру интелигентних технолошких система базираних на парадигми Индустрије 4.0. „*Терминирање представља процес планирања машинске обраде, као и „придруживање“ технолошких операција за сваки од обрадака одговарајућој машини алатки, уз истовремено временско распоређивање“* [128]. Оптималан план терминирања у оквиру интелигентног технолошког система остварује се процесом оптимизације. Оптимизација се може дефинисати као процес проналажења променљивих (нпр. машина алатки или алтернативних технолошких процеса) које оптимално задовољавају неки критеријум перформанси (нпр. минимално време обраде) односно функцију циља. Када се оптимизује само један критеријум перформанси интелигентног технолошког система, анализирани проблем је дефинисан као једнокритеријумска оптимизација. Уколико је потребно оптимизовати два или више критеријума перформанси (нпр. захтева се минимална цена производа и минимално време обраде), проблем је у том случају дефинисан као вишекритеријумска оптимизација. Неке од функција циља које се користе су: (i) минимизација укупног времена обраде свих делова чије се терминирање врши (енгл. *Makespan*), (ii) максимизација уравнотеженог искоришћења машина алатки (енгл. *Balanced level of machine utilization*), (iii) минимизација средњег времена транспортних токова материјала (енгл. *Mean flow time*) и (iv) минимално време транспорта (енгл. *Transpiration time*). Такође, истовременом оптимизацијом више функција циља (применом вишекритеријумске оптимизације) може се постићи оптималан однос између две функције које у технолошком систему имају исти приоритет.

Резултат терминирања представља план терминирања у оквиру кога се налази редослед, време почетка и завршетка технолошких операција свих делова, као и информација о додељеним машина алаткама, алатима и оријентацијама алата за сваку технолошку операцију. Поред тога, у оквиру плана терминирања интегрисано је време чекања, почетка и завршетка транспорта делова између машина, које извршава један мобилни робот. План терминирања може графички бити приказан Грантовим дијаграмом где су на ординати различите машине алатке, а на апсциси време. Међутим, код вишекритеријумске оптимизације, не постоји једно оптимално решење и један оптималан план терминирања, већ се добија Парето фронт који представља скуп оптималних решења са одређеним уделом функција циља [129].

Терминирање представља *NP-hard* дискретан оптимизационо-комбинаторни проблем (енгл. *non-deterministic polynomial optimization problem*) са простором решења великих димензија, а уколико се у оквиру плана терминирања уврсте додатна ограничења везана за транспорт делова једним мобилним роботом, комплексност самог процеса се знатно повећава. Из тог разлога, оптимизација у оквиру ове докторске дисертације врши се различитим метахеуристичким алгоритмима. Карактеристике метахеуристичких алгоритама су једноставност при имплементацији, могућност оптимизације без аналитичког дефинисања извода функције циља и ефикасно претраживање простора решења [130]. Алгоритам који ће бити представљен за решавање проблема терминирања једног мобилног робота је алгоритам инспирисан интелигенцијом чопора вукова (енгл. *Grey Wolf Optimizer* – GWO [131]).

Проблем терминирања технолошког система, у оквиру кога транспортне задатке извршава један мобилни робот, може се дефинисати на следећи начин [132], [11]: за задати скуп делова *Js* (5.1):

$$Js = \{J_1, J_2, ..., J_N\}. \tag{5.1}$$

који се могу обрадити у технолошком систему у оквиру кога се налази скуп *Ms* (5.2) машина алатки:

$$Ms = \{M_1, M_2, ..., M_M\}, \tag{5.2}$$

и скуп алата *Tls* (5.3) који се могу поставити на машине алатке у одговарајућој оријентацији дефинисаној скупом оријентација алата *Tds* (5.4):

$$Tls = \{Tl_1, Tl_2, ..., Tl_L\}, \tag{5.3}$$

$$Tds = \{Td_1, Td_2, ..., Td_D\}. \tag{5.4}$$

потребно је одабрати оптималан алтернативни технолошки процес (према задатом критеријуму перформанси) из скупа алтернативних технолошких процеса Gs_i (5.5) за сваки део:

$$Gs_i = \{G_{i1}, G_{i2}, ..., G_{iG}\}, \tag{5.5}$$

који се дефинише одређеним скупом операција OP_i (5.6):

$$OP_i = \{OP_{i1}, OP_{i2}, ..., OP_{iP}\}. \tag{5.6}$$

Одабиром оптималног алтернативног технолошког процеса за сваки део, истовремено се свакој операцији додељују одређени технолошки ентитети у виду машине алатке, алата и оријентације алата, на основу скупа алтернативних технолошких ресурса за сваку операцију RO_{ij} (5.7):

$$RO_{ij} = \{RP_{ij1}, RP_{ij2}, ..., RP_{ijF}\}. \tag{5.7}$$

5.1. Преглед стања у области терминирања роботских система

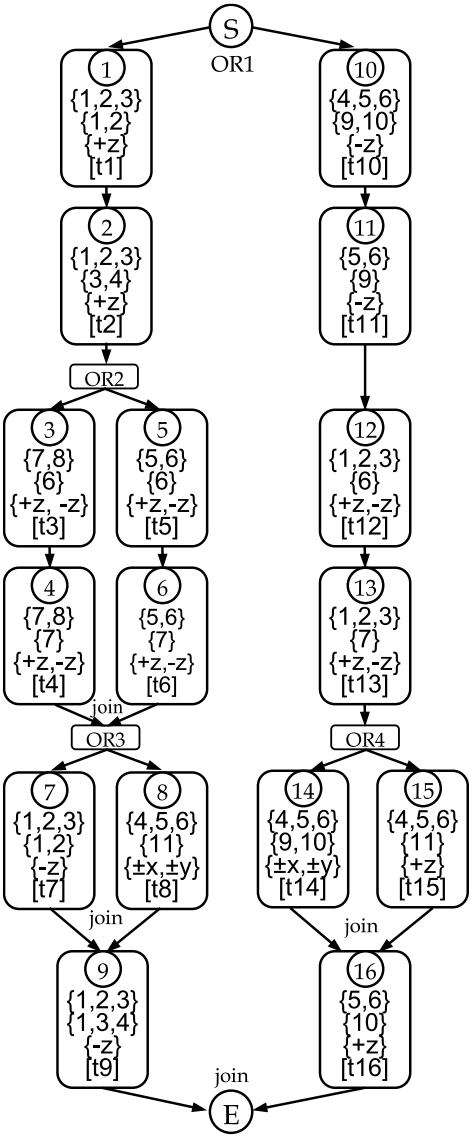
У раду [133] предложена је методологија на бази генетичких алгоритама (енгл. *Genetic Algorithm* – GA) за оптимизацију транспортних токова материјала у технолошком окружењу. Функција циља при оптимизацији је минимално време које је потребно роботском систему да изврши све транспортне задатке. Како се део може наћи у било којим координатама у простору на почетку процеса оптимизације, овај проблем је значајно комплекснији од проблема трговачког путника (енгл. *Traveling salesmen problem*). Извршено је бинарно кодирање проблема, а бинарни стринг је садржао податке о положају дела и путање којом робот треба да се креће. Изведене су две експерименталне верификације, са 30 односно 50 машина на које треба поставити део, у оквиру којих је доказана валидност предложеног алгоритма оптимизације. Аутори рада [134] развили су систем базиран на вештачким неуронским мрежама за терминирање индустријског робота. Задатак индустријског робота био је да опслужује више машина алатки. Предложена функција циља укључивала је кашњење робота и минималну дистанцу кретања роботског система. *Hopfield*-ове неуронске мреже су поређене са *branch-and-bound* алгоритмом и методом која се базира на хеуристици. Метода базирана на вештачким неуронским мрежама остварује резултате знатно ефикасније од друге две методе. У оквиру рада [135] развијена је методологија за терминирање једног мобилног роботског система за задатак транспорта делова. Примењени су алгоритам табу претраге (енгл. *Tabu search*) и алгоритам табу претраге који је базиран на стохастици. Алгоритам стохастичне табу претрага разликује се од традиционалног на основу меморије у оквиру које чува одређени број потенцијалних решења, док се на основу вероватноће селектује један од њих. Алгоритми су поређени на основу функције циља која минимизира дужину путање коју је робот остварио, а након експерименталне верификације алгоритам табу претраге показао је значајно боље

квалитативне и квантитативне резултате. Терминирање технолошког система са једном врстом машина алатки и једним роботским системом за транспорт анализирано је у раду [136]. Примењена је табу претрага како би се решио проблем терминирања дефинисан као генерализација проблема трговачког путника. Оптимизација је извршена на основу функције циља која је подразумевала минимизацију укупног времена транспорта и времена чекања. Алгоритам табу претраге је унапређен применом механизма који тренутно решење може променити решењем које се налази у његовој непосредној близни у простору решења. Експерименталном верификацијом утврђено је да је предложени унапређени табу алгоритам значајно ефикаснији од стандардне табу претраге. Математичко моделирање проблема терминирања једног мобилног робота у производном окружењу предложено је у раду [137]. Терминирање је формулисано као проблем мешовитог целобројног програмирања (енгл. *Mixed-integer programming*). Задатак мобилног робота подразумевао је допуњавање делова у вибрационом додавачу, а функција циља је подразумевала минимизацију пређеног пута мобилног робота. Како је применом мешовитог целобројног програмирања било могуће решити само оптимизационе проблеме са мањим бројем променљивих, аутори су наставили развој и предложили решавање проблема терминирања једног мобилног робота применом генетичких алгоритама [138], [139]. Анализирани проблем је оптимизован на бази функције циља која подразумева минимално време транспорта. Механизам редног укрштања (енгл. *order crossover*) је примењен у оквиру генетичког алгоритма. Експерименталном верификацијом утврђена је тачности и ефикасност развијеног генетичког алгоритма. Моделирање проблема терминирања мобилног роботског система који може да извршава друге задатке у технолошком систему, док чека на извршавање главних транспортних задатака анализирано је у раду [140]. Проблем је дефинисан тако да мобилни робот извршава задатке мањег приоритета све док није потребан ради извршавања приоритетнијих задатака. За решавање анализираног проблема примењени су генетички алгоритми, док је функција циља била минимално време завршетка последње технолошке операције. Вишекритеријумско терминирање аутоматски вођених робоколица у флексибилном технолошком окружењу анализирано је у раду [141]. Примењен је хибридни генетички алгоритам интегрисан са алгоритмом инспирисаним интелигенцијом роја честица (енгл. *Particle Swarm Optimization* – PSO) и фази логиком (енгл. *Fuzzy logic*). Алгоритам базиран на интелигенцији чопора вукова (енгл. *Grey Wolf Optimization Algorithm*) имплементиран у сврху терминирања једног мобилног робота предложен је у раду [142]. Имплементирана функција циља подразумевала је минимизацију времена транспорта. Експерименталном верификацијом утврђено је да GWO постиже боље резултате оптимизације од алгоритмима PSO и GA. Оптимални план терминирања тестиран је применом мобилног робота Khepera II. Математичко моделирање проблема терминирања једног мобилног роботског система у технолошком окружењу дато је у референци [143]. Имплементирана је функција циља која минимизује укупно време потребно да се обраде сви делови. Како би се избегао сценарио у којем робот не може да изврши задатак постављања дела на машину, анализирани су механизми опозиционог и мрежног терминирања. Главна предност предложеног модела огледа се у минималном времену чекања робота у току транспорта и побољшаној ефикасности процеса терминирања. Проблем терминирања два транспортна средства у флексибилном технолошком окружењу представљен је у раду [144]. Примењен је метод мешовитог целобројног програмирања за вишекритеријумску оптимизацију минималне потрошње енергије и минималног времена потребног за обраду свих делова. Оптимизација је извршена применом еволуционог алгоритма који се базира на популацији. Предложени алгоритам поређен је са различитим метахеуристичким оптимизационим алгоритмима, а експерименталном евалуацијом утврђено је да се примењеним алгоритмом значајно умањују губици електричне енергије и смањује време потребно за обраду свих делова. Терминирање мобилних робота у оквиру интелигентног технолошког система са примарним фокусом на енергетску ефикасност представљено је у [145]. Еволуциони алгоритам који се базира на претходно стеченом знању је имплементиран ради минимизације вишекритеријумске

функције циља која подразумева укупну потрошњу електричне енергије и минимално време потребно за обраду свих делова. Имплементирана је нова шема декодирања која повећава ефикасност претраге и омогућава претраживање већег простора решења. Такође, адаптивни механизам мутације и селекције је имплементиран како би се успоставила равнотежа између експлорације и експлоатације. Три експерименталне верификације указују да предложени алгоритам остварује значајно боље резултате у поређењу са четири *state-of-the-art* алгоритма. Проблем терминирања више мобилних робота у оквиру флексибилног технолошког система је анализиран у [146]. Хибридни дискретни *Salp Swam* алгоритам је предложен за решавање проблема мешовитог целобројног програмирања. Оригинални алгоритам унапређен је имплементацијом механизама укрштања и варијације. Такође, имплементирана је посебна хеуристика за иницијализацију популације на почетку процеса оптимизације. Евалуација резултата оптимизације на основу функције циља укупног времена потребног за обраду свих делова показала је ефикасност предложеног алгоритма. За разлику од анализираних радова, у оквиру предметне докторске дисертације развијен је модел за вишекритеријумско терминирање једног мобилног робота који врши све транспортне задатке у интелигентном технолошком систему. У оквиру модела интегрисано је 13 различитих функција, чијом се интеграцијом дефинишу вишекритеријумске функције циља. Оптимизација развијеног модела извршава се помоћу унапређеног GWO алгоритма, са посебно развијеним стратегијама за селекцију лидера. Предложени модел остварује боље перформансе у односу на три *state-of-the-art* метахеуристичка алгоритма.

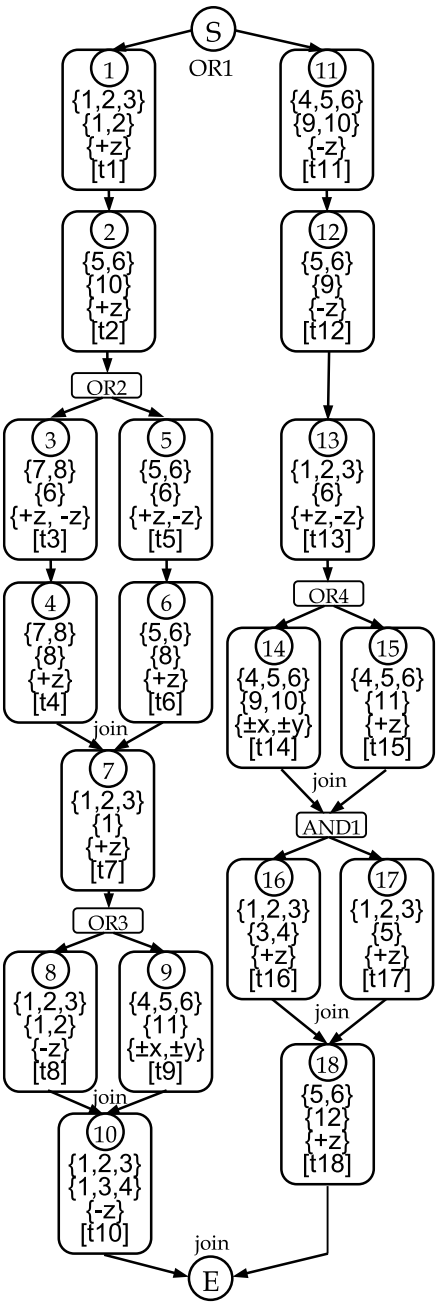
5.2. Кодирање јединки плана терминирања

Као што је приказано у претходном потпоглављу, план терминирања дефинисан је одабиром алтернативног технолошког процеса сваког дела, машине алатке, алата и оријентације алата за сваку операцију и редоследа операција на машинама алаткама. Подаци о три коришћена репрезентативна дела у процесу терминирања приказани су на сликама 5.1, 5.2 и 5.3 [128]. У оквиру AND/OR мрежа алтернативних технолошких процеса (слике 5.1-3 лево), могу се приметити операције које је потребно извршити, као и машине алатке, алати и оријентације алата за те операције. OR конекторима дефинисане су гране између којих се може одабрати једна како би технолошки процес био валидан, док се код AND грана све операције морају извршити, али се редослед операција може мењати. У десној табели на сликама 5.1-3 дата су времена обраде за све алтернативне операције које су дефинисане на основу различите комбинације машина алатки и алата. Детаљна упутства за креирање и коришћење AND/OR мрежа представљена су у литератури [128]. Кодирање плана терминирања потребно је извршити како би се оптимизационим алгоритмима могао представити један план терминирања, а пример кодирања приказан је на слици 5.4. Како се врши терминирање три дела, у оквиру стринга технолошког процеса постоје три гена, где сваки ген представља један део, а цифра у гену означава одабрани алтернативни технолошки процес (слика 5.4). Максималан број операција у оквиру алтернативних технолошких процеса свих делова је седам па ће сам план терминирања имати 21 елемент (три дела по максимално седам операција). План терминирања садржи цифре од нула до три. Цифрама од један до три, дефинисан је редослед операција делова на машинама, док су нуле додате како би сви стрингови имали исти број гена што може настати због различитог број операција одабраних алтернативних технолошких процеса делова. На основу датог плана терминирања (слика 5.4), прво се извршава прва операција дела два, затим прва операција дела један, а након ње друга операција дела два итд. У оквиру стринга машина алатки, црвеном бојом дефинисане су одабране машине следствено за сваку операцију првог дела, наранџастом машине алатке за други део и жутом машине за трећи део, а на идентичан начин дефинисани су и алати, као и њихове оријентације. Како делови један и три имају шест операција за одабрани алтернативни технолошки процес, за седму операцију додате су нуле.



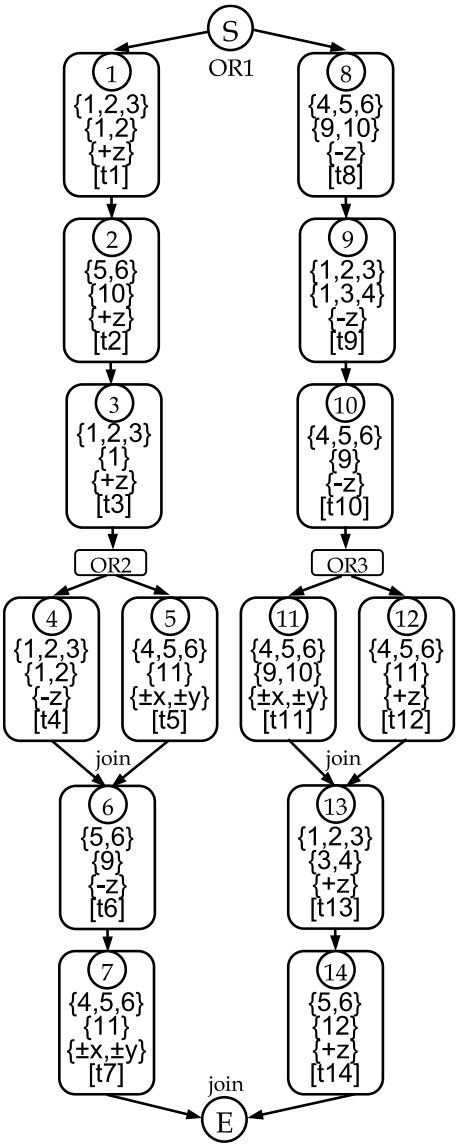
t1 = [2,3 3,3 4,6 6,5 3,8 5,5]
t2 = [1,2 1,0 1,9 1,5 2,0 1,6]
t3 = [5,5 3,8]
t4 = [4,8 8,1]
t5 = [3,3 3,4]
t6 = [10,7 24,1]
t7 = [2,3 3,3 4,6 6,5 3,8 5,5]
t8 = [5,7 4,2 2,8]
t9 = [0,5 0,7 0,4 1,0 1,4 0,8 0,8 1,1 0,7]
t10 = [2,7 2,8 2,9 2,4 2,8 2,5]
t11 = [4,6 7,2]
t12 = [3,3 4,8 3,3]
t13 = [13,4 26,6 30,2]
t14 = [2,7 2,8 2,9 2,4 2,9 2,4]
t15 = [5,7 4,2 2,8]
t16 = [3,1 3,2]

Слика 5.1. Мрежа алтернативних технолошких процеса обраде репрезентативног дела 1 (ti представљају времена обраде) [2], [128].



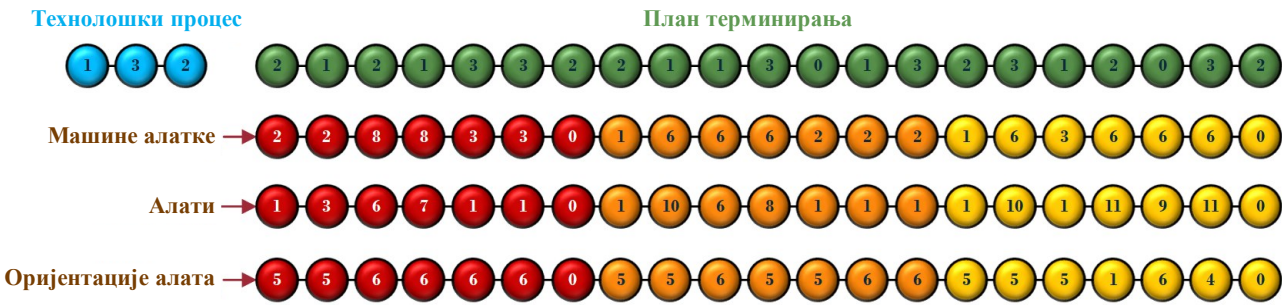
t1 = [2,3 3,3 4,6 6,5 3,8 5,5]
t2 = [3,1 3,2]
t3 = [5,5 3,8]
t4 = [34,3 13,7]
t5 = [3,3 3,4]
t6 = [30,2 13,4]
t7 = [0,7 0,8 0,9]
t8 = [2,3 3,3 4,6 6,5 3,8 5,5]
t9 = [5,7 4,2 2,8]
t10 = [0,5 0,7 0,4 1,0 1,4 0,8 0,8 1,1 0,7]
t11 = [2,7 2,8 2,9 2,4 2,8 2,5]
t12 = [4,6 7,2]
t13 = [3,3 4,8 3,3]
t14 = [2,7 2,8 2,9 2,4 2,9 2,4]
t15 = [5,7 4,2 2,8]
t16 = [1,2 1,0 1,9 1,5 2,0 1,6]
t17 = [1,1 1,5 1,8]
t18 = [0,5 0,6]

Слика 5.2. Мрежа алтернативних технолошких процеса обраде репрезентативног дела 2 (ti представљају времена обраде) [2], [128].



t1 = [2,3 3,3 4,6 6,5 3,8 5,5]
t2 = [3,1 3,2]
t3 = [0,6 0,7 0,8]
t4 = [2,3 3,3 4,6 6,5 3,8 5,5]
t5 = [5,7 4,2 2,8]
t6 = [4,6 7,2]
t7 = [10,8 9,7 7,4]
t8 = [2,7 2,8 2,9 2,4 2,8 2,5]
t9 = [0,5 0,7 0,4 1,0 1,4 0,8 0,8 1,1 0,7]
t10 = [4,3 3,5 6,8]
t11 = [2,7 2,8 2,9 2,4 2,9 2,4]
t12 = [5,7 4,2 2,8]
t13 = [1,2 1,0 1,9 1,5 2,0 1,6]
t14 = [0,4 0,5]

Слика 5.3. Мрежа алтернативних технолошких процеса обраде репрезентативног дела 3 (ti представљају времена обраде) [2], [128].



Слика 5.4. Кодирање плана терминирања.

5.3. Математички модел

Иницијалне претпоставке везане за математичко моделирање процеса терминирања технолошког система где један мобилни робот извршава транспортне задатке су следеће:

1) Све машине алатке и робот су слободни на почетку извршавања плана терминирања;
2) У почетном тренутку извршавања плана терминирања сви делови су постављени на машину алатку на којој се извршава њихова прва операција;
3) Мобилни робот позициониран је поред прве машине на почетку плана терминирања;
4) Уколико постоји транспорт дела пре неке операције, додатно се уводи припремно време машине које укључује време измене алата и време измене оријентације алата;
5) Свака машина алатка може да обрађује само један део у сваком тренутку и мобилни робот може да транспортује само један део у сваком тренутку;
6) Само један алтернативни технолошки процес је одабран за сваки део;
7) Свакој технолошкој операцији додељена је по једна машина алатка, алат и оријентација алата;
8) Наредна операција на машини не може да почне уколико претходна није завршена;
9) Наредна операција дела не може да почне уколико претходна операција дела није завршена.

Индекси коришћени у математичком моделирању су следећи:

- i, i' су индекси који дефинишу делове, $1 \le i', i \le N$,
- l, l' су индекси који дефинишу алтернативне технолошке процесе, $1 \le l', l \le G_i$,
- j, j' су индекси који дефинишу операције везане за алтернативни технолошки процес l дела i, $1 \le j', j \le P_{il}$,
- k, k', k'' су индекси машина алатки, $1 \le k, k', k'' \le M$,
- p је индекс алата за операцију j у оквиру алтернативног технолошког процеса l дела i, $1 \le p \le L_{ijl}$,
- q је индекс оријентације алата за операцију j у оквиру алтернативног технолошког процеса l дела i, $1 \le q \le D_{ijl}$,

где је N укупан број делова, G_i је број алтернативних технолошки процеса дела i, P_{il} је број операција за алтернативни технолошки процес l дела i, M је укупан број машина алатки, L_{ijl} и D_{ijl} су укупни бројеви алата и њихових оријентација за одређену операцију. Индекс j' користи се за означавање операције која претходи операцији j у оквиру плана терминирања (погледати слику 5.4). Такође, i', l' и k' представљају индекс дела, алтернативног технолошког процеса и машине за операцију j', следствено. Индекс k'' служи за представљање машине на којој је претходна операција дела извршена.

Нотације променљивих коришћених у математичком моделирању су следеће:

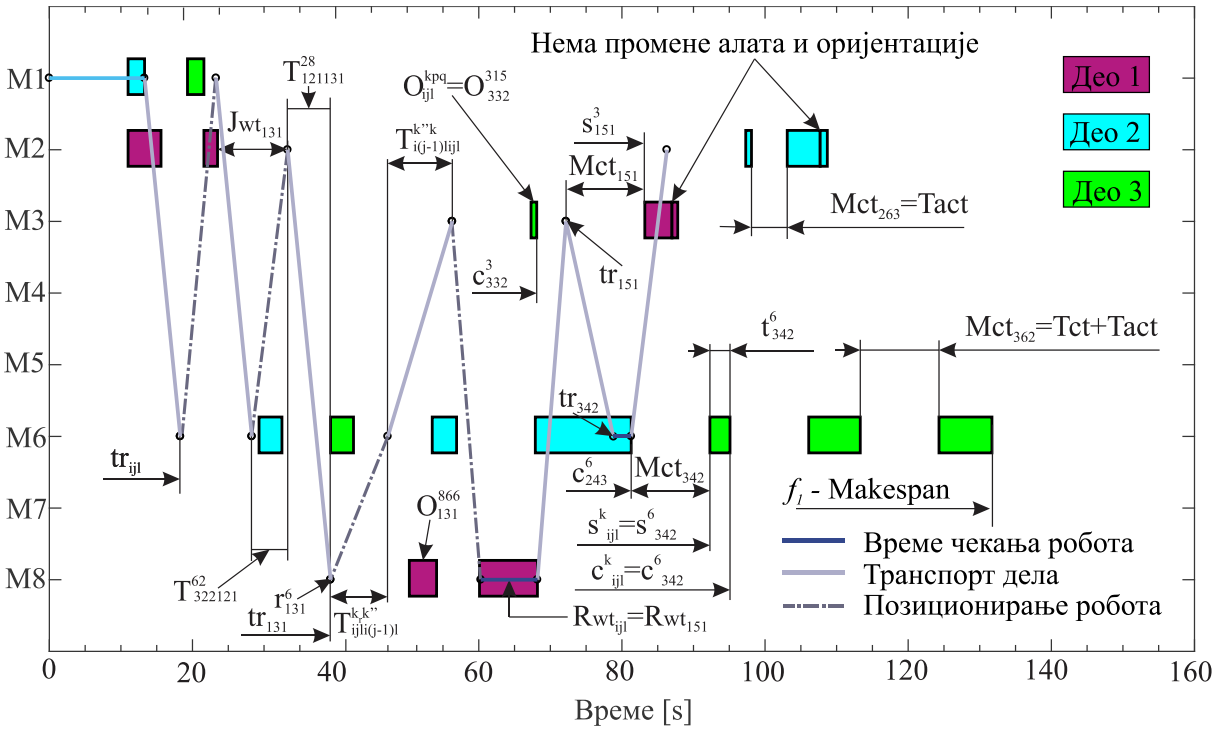
- O_{ijl}^{kpq} : представља j-ту операцију i-тог дела у оквиру l-тог алтернативног технолошког процеса, који се извршава на k-тој машини p-тим алатом у q-тој оријентацији,
- t_{ijl}^{kpq} : главно време обраде j-те операције i-тог дела у оквиру l-тог алтернативног технолошког процеса, који се извршава на k-тој машини p-тим алатом у q-тој оријентацији,
- $r_{ijl}^{k_r}$: машина k_r где се мобилни робот налази пре почетка транспорта за j-ту операцију i-тог дела у оквиру l-тог алтернативног технолошког процеса,
- tr_{ijl} : време завршетка транспортног задатка мобилног робота за j-ту операцију i-тог дела у оквиру l-тог алтернативног технолошког процеса,

- $T^{k''k}_{i(j-1)lijl}$: време транспорта између машине за претходну операцију дела *i* (*k''*) и машине *k* за тренутну операцију дела *i* (транспорт дела),
- $T^{k_r k''}_{ijli(j-1)l}$: време транспорта између машине k_r где се налази мобилни робот пре извршавања транспорта и машине *k''* где се извршава обрада претходне операције дела *i* (време доласка до машине са које треба да се однесе део),
- s^k_{ijl}: време почетка *j*-те операције *i*-тог дела у оквиру *l*-тог алтернативног технолошког процеса,
- c^k_{ijl}: време завршетка *j*-те операције *i*-тог дела у оквиру *l*-тог алтернативног технолошког процеса,
- Rwt_{ijl}: време чекања робота пре транспорта за обраду *j*-те операције *i*-тог дела у оквиру *l*-тог алтернативног технолошког процеса,
- Jwt_{ijl}: време чекања на почетак транспорта за *j*-ту операцију *i*-тог дела у оквиру *l*-тог алтернативног технолошког процеса,
- Mct_{ijl}: припремно време машине за *j*-ту операцију *i*-тог дела у оквиру *l*-тог алтернативног технолошког процеса,
- Tct: време измене алата,
- $Tact$: време измене оријентације алата,
- dd_i: рок за завршетак *i*-тог дела,
- dd_r: рок за завршетак транспорта.

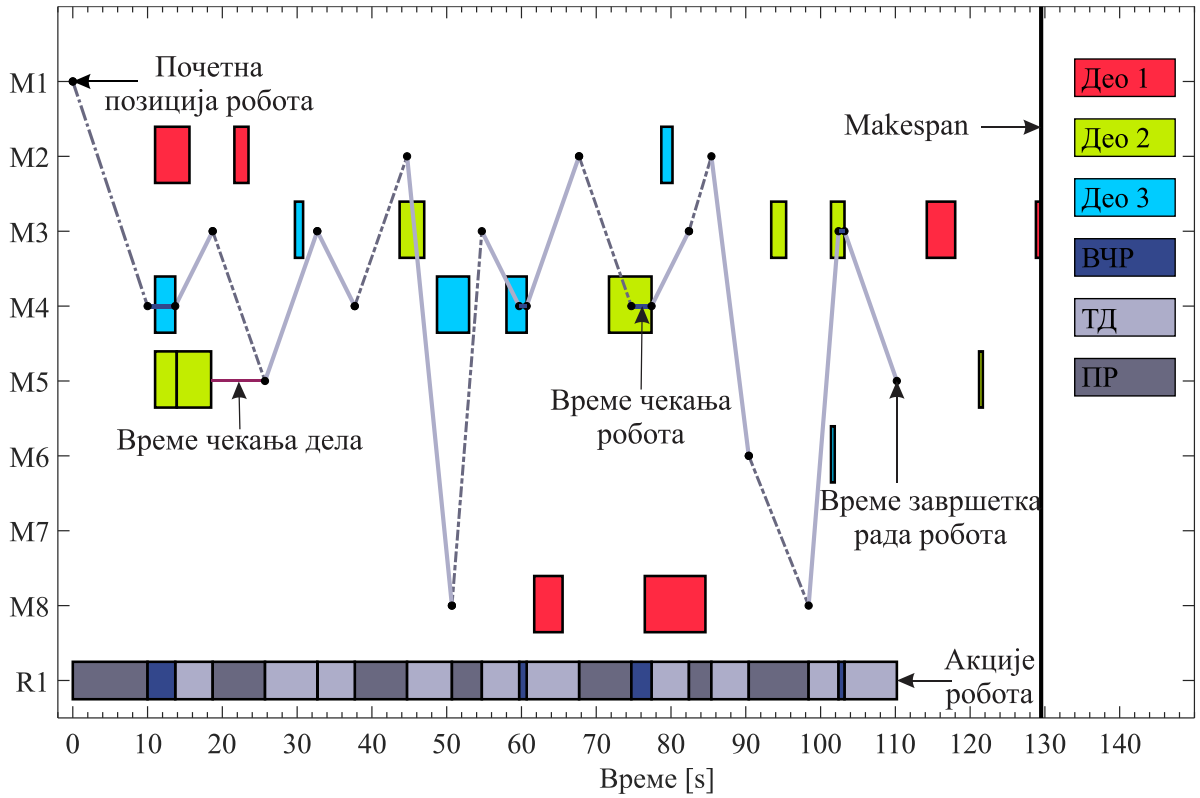
Променљиве за избор алтернативних ресурса:

- $X_{il} = \begin{cases} 1, & \text{уколико је } l\text{-ти алтернативни технолошки процес одабран за } i\text{-ти део} \\ 0, & \text{у супротном} \end{cases}$
- $Z^k_{ijl} = \begin{cases} 1, & \text{уколико је машина } k \text{ одабрана за операцију } O^{kpq}_{ijl} \\ 0, & \text{у супротном} \end{cases}$
- $\Psi^p_{ijl} = \begin{cases} 1, & \text{уколико је алат } p \text{ одабран за операцију } O^{kpq}_{ijl} \\ 0, & \text{у супротном} \end{cases}$
- $\Xi^q_{ijl} = \begin{cases} 1, & \text{уколико је оријентација алата } q \text{ одабрана за операцију } O^{kpq}_{ijl} \\ 0, & \text{у супротном} \end{cases}$
- $\lambda_{ijl} = \begin{cases} 1, & \text{уколико } j\text{-та операција } i\text{-тог дела није прва операција тог дела} \\ 0, & \text{у супротном} \end{cases}$
- $\Theta_{ijl} = \begin{cases} 1, & \text{уколико машина } k \text{ операције } O^{kpq}_{ijl} \text{ није иста као машина } k'' \text{ операције } O^{k''p'q'}_{i(j-1)l} \\ 0, & \text{у супротном} \end{cases}$
- $R_{ijlk_r} = \begin{cases} 1, & \text{уколико машина } k'' \text{ операције } O^{k''pq}_{i(j-1)l} \text{ није иста као} \\ & \text{машина } k_r \text{ где је робот тренутно} \left(r^{k_r}_{ijl}\right) \\ 0, & \text{у супротном} \end{cases}$

Гантов дијаграм за претходно приказани план терминирања (слика 5.4) може се уочити на слици 5.5. На слици 5.6 дат је други пример Гантовог дијаграма где се могу уочити наглашени транспортни задаци које извршава мобилни робот.



Слика 5.5. Гантов дијаграм.



Слика 5.6. Гантов дијаграм са означеним параметрима везаним за мобилни робот, ВЧР – време чекања робота, ТД – транспорт дела, а ПР – позиционирање робота.

Математички модел за одређивање променљивих које је потребно дефинисати како би се израчунале вредности функције циља дат је у наставку. Како постоји само један мобилни робот у интелигентном технолошком систему, мора се одредити поред које машине алатке се мобилни робот налази у сваком тренутку. Тренутак када мобилни робот заврши транспортни задатак за операцију O_{ijl} уврсти се у променљиву tr_{ijl}. Уколико није било потребно да робот

извршава транспорт за операцију O_{ijl}, време завршетка транспорта ће бити исто као и код претходне операције у плану терминирања ($tr_{i'j'l'}$), једначина (5.8):

$$tr_{ijl} = tr_{i'j'l'} + \lambda_{ijl} \cdot \left(Rwt_{ijl} + \Theta_{ijl} \cdot \left(T^{k''k}_{i(j-1)lijl} + R_{ijlk_r} \cdot T^{k_r k''}_{ijli(j-1)l} \right) \right) \tag{5.8}$$

Податак о томе поред које машине алатке се робот тренутно налази дефинисан је променљивом $r^{k_r}_{ijl}$. Користећи тај податак, као и податак везан за машину на којој је извршена претходна операција дела, могу се дефинисати две акције мобилног робота. Уколико се робот налази на машини на којој је извршена претходна операција дела, укупно време транспорта ће укључивати само време транспорта дефинисано у оквиру променљиве $T^{k_r k''}_{ijli(j-1)l}$. Уколико робот није тренутно поред машине на којој се претходна операција дела извршила, неопходно је да робот изврши акцију позиционирања чије је време дефинисано променљивом $T^{k''k}_{i(j-1)lijl}$. У том случају, робот долази до машине где се део налази (где је завршена претходна операција тог дела), како би затим могао да га транспортује на машину за наредну операцију. У оквиру једначине (5.8), на основу R_{ijlkr} и Θ_{ijl}, дефинише се да ли постоји акција позиционирања и акција транспорта. Како би се израчунало време завршетка транспорта робота, потребно је дефинисати још променљиву време чекања робота (Rwt_{ijl}), а уз њу ће бити дефинисана и променљива време чекања дела (Jwt_{ijl}) на основу времена чекања (Wt_{ijl}) датог у једначини (5.9):

$$Wt_{ijl} = \lambda_{ijl} \cdot \Theta_{ijl} \cdot \left(c^{k''}_{i(j-1)l} - \left(tr_{i'j'l'} + R_{ijlk_1} \cdot T^{k_r k''}_{ijli(j-1)l} \right) \right) \tag{5.9}$$

где је са $c^{k''}_{i(j-1)l}$ дефинисано време завршетка претходне операције дела i. Време чекања робота (5.10) је период од тренутка када робот заврши акцију позиционирања, а пре него што почне акција транспорта дела. Чекање настаје услед стизања робота до машине на којој се извршава претходна операција дела пре завршетка саме операције. Период између тренутка када се заврши операција дела на машини алатки и тренутка почетка транспорта ка наредној машини назива се време чекања дела (5.11).

$$Rwt_{ijl} = \begin{cases} Wt_{ijl}, \text{ if } Wt_{ijl} > 0 \\ 0, \quad \text{ if } Wt_{ijl} \le 0 \end{cases} \tag{5.10}$$

$$Jwt_{ijl} = \begin{cases} \left| Wt_{ijl} \right|, \text{if } Wt_{ijl} < 0 \\ 0, \quad \text{if } Wt_{ijl} \ge 0 \end{cases} \tag{5.11}$$

Као што се може закључити из једначина (5.10) и (5.11), уколико постоји време чекања робота, неће постојати време чекања дела и обрнуто. Када се заврши транспорт дела, потребно је ажурирати време завршетка транспортног задатка (5.8), као и позицију мобилног робота (5.12):

$$r^{k_r}_{ijl} = \begin{cases} k_r', \text{ if } \lambda_{ijl} \cdot \Theta_{ijl} = 0 \\ k, \text{ if } \lambda_{ijl} \cdot \Theta_{ijl} = 1 \end{cases} \tag{5.12}$$

Време почетка сваке операције дефинисано је на основу једначине (5.13):

$$\lambda_{ijl} \cdot \Theta_{ijl} \cdot tr_{ijl} - c^{k}_{i'j'l'} \begin{cases} > 0, s^{k}_{ijl} = tr_{ijl} + Mct_{ijl} \\ \le 0, s^{k}_{ijl} = c^{k}_{i'j'l'} + Mct_{ijl} \end{cases} \tag{5.13}$$

где је са $c^{k}_{i'j'l'}$ дефинисано време завршетка претходне операције на машини k. Припремно време машине алатке (Mct_{ijl}), рачуна се на основу једначине (5.14):

$$Mct_{ijl} = \Theta_{ijl} \cdot \left(Tct + Tact\right) + \left(1 - \Theta_{ijl}\right) \cdot \left(Tct \cdot \tau 1_{ijl} + Tact \cdot \tau 2_{ijl}\right) \tag{5.14}$$

На основу параметра Θ_{ijl} дефинисан је услов који гарантује да у случају промене машина на којој се део обрађује, постоји припремно време везано за измену алата (*Tct*) и оријентације алата (*Tact*), док уколико део остаје на истој машини за две узастопне операције, припремно време зависи од променљивих $\tau 1_{ijl}$ и $\tau 2_{ijl}$. Време завршетка операције c_{ijl}^{k} одређује се на основу једначине (5.15):

$$c_{ijl}^{k} = s_{ijl}^{k} + t_{ijl}^{kpq}. \tag{5.15}$$

Затим, потребно је математички дефинисати функције циља. Прво је дефинисана функција циља које се најчешће користи при терминирању технолошких система – минимизација времена потребног да се изврши обрада свих делова чије се терминирање врши (енгл. *Makespan* – 5.16). На основу ове функције, технолошки ентитети се терминирају тако да време операције која се последња заврши буде минимално:

$$f_1 = \min\left(\max\left(c_{ijl}^{k}\right)\right),\ \forall i \in \left[1, N\right],\ l \in \left[1, G_i\right],\ j = P_{il}, k \in \left[1, M\right]. \tag{5.16}$$

Друга функција циља односи се на мобилни роботски систем и назива се време завршетка рада робота (енгл. *Robot finishing time* – 5.17). Применом ове функције, минимизира се време које је роботу потребно да изврши све акције.

$$f_2 = \min\left(\max\left(tr_{ijl}\right)\right)\ \forall i \in \left[1, N\right],\ l \in \left[1, G_i\right],\ j = P_{il}. \tag{5.17}$$

Трећа функција циља такође се директно односи на роботски систем и представља минимално време транспорта (енгл. *Robot transport time* – 5.18). Уколико се роботском систему испразни батерија, а неопходно је завршити започету обраду пре пуњења, могуће је минимизирати време транспорта робота.

$$f_3 = \min\left(\sum_{i=1}^{N}\sum_{j=1}^{P_{il}} \Theta_{ijl} \cdot \left(T_{i(j-1)lijl}^{k''k} + R_{ijlk_1} \cdot T_{ijli(j-1)l}^{k_r k''}\right)\right),\ l \in \left[1, G_i\right],\ k \in \left[1, M\right]. \tag{5.18}$$

Наредна функција циља односи се на искоришћење мобилног робота (енгл. *Robot utilization* – 5.19). Поменута функција дефинише се као однос времена завршетка рада робота и времена када робот врши кретање.

$$f_4 = \min\left(\frac{f_2}{f_3}\right). \tag{5.19}$$

Пета функција циља је минимизација времена које мобилни робот чека (енгл. *Robot waiting time* – 5.20),

$$f_5 = \min\left(\sum_{i=1}^{N}\sum_{j=1}^{P_{il}} Rwt_{ijl}\right),\ l \in \left[1, G_i\right], \tag{5.20}$$

такође, дефинише се и функција циља која се односи на минимизацију времена чекања делова (енгл. *Job waiting time* – 5.21).

$$f_6 = \min\left(\sum_{i=1}^{N}\sum_{j=1}^{P_{il}} Jwt_{ijl}\right),\ l \in \left[1, G_i\right], \tag{5.21}$$

Минимизација интегрисаног времена завршетка свих делова чије се терминирање врши представља седму функцију циља (енгл. *Total flow time* – 5.22), док је осмом функцијом дефинисана минимизација просечног времена завршетка свих делова (енгл. *Mean flow time* – 5.23).

$$f_7 = \min\left(\sum_{i=1}^{N}\left(c_{ijl}^{k}\right)\right),\ j = P_{il}, l \in \left[1, G_i\right], k \in \left[1, M\right], \tag{5.22}$$

$$f_8 = \min\left(\frac{1}{N}\sum_{i=1}^{N}\left(c_{ijl}^{k}\right)\right),\ j = P_{il}, l \in \left[1, G_i\right], k \in \left[1, M\right], \tag{5.23}$$

Затим су дефинисане девета и десета функција циља које се односе на минимизацију укупног времена које је потребно роботу да заврши последњу операцију транспорта за сваки део понаособ (енгл. *Total robot flow time* – 5.24) и просечног времена које је потребно роботу да заврши последњу операцију транспорта за сваки део (енгл. *Mean robot flow time* – 5.25).

$$f_9 = \min\left(\sum_{i=1}^{N}\left(tr_{ijl}\right)\right), j = P_{il}, l \in \left[1, G_i\right], \tag{5.24}$$

$$f_{10} = \min\left(\frac{1}{N}\sum_{i=1}^{N}\left(tr_{ijl}\right)\right), j = P_{il}, l \in \left[1, G_i\right], \tag{5.25}$$

Минимизација укупне цене транспорта применом мобилног робота предложена је у оквиру функције циља f_{11} (енгл. *Total robot cost* – 5.26),

$$f_{11} = \min\left(\sum_{i=1}^{N}\sum_{j=1}^{P_{il}}\Theta_{ijl}\left(T_{i(j-1)lijl}^{k''k}w_1 + R_{ijlk_r}T_{ijli(j-1)l}^{k_r k''}w_2\right) + Rwt_{ijl}w_3\right),\ l \in \left[1, G_i\right],\ k \in \left[1, M\right], \tag{5.26}$$

где је утицај свих компоненти транспорта (позиционирање, транспорт и чекање робота) у укупној цени транспорта дефинисан фактором w_x (x = 1, 2, 3).

Како се у оквиру технолошког окружења морају испоштовати рокови израде делова, наредна функција циља везана је за укупно кашњење (енгл. *Total tardiness* – 5.27), које је дефинисано као разлика времена завршетка последње операције дела и рока за завршетак тог дела:

$$f_{12} = \min\left(\sum_{i=1}^{N}\max\left(0, c_{ijl}^{k} - d_i\right)\right),\ j = P_{il}, l \in \left[1, G_i\right], k \in \left[1, M\right], \tag{5.27}$$

Последња предложена функција циља везана је за минимизацију кашњења робота (енгл. *Total robot tardiness* – 5.28) и израчунава се као рок за завршетак транспортних задатака за неки део и време завршетка рада робота на том делу:

$$f_{13} = \min\left(\max\left(0, \max\left(tr_{ijl}\right) - d_r\right)\right), \forall i \in \left[1, N\right], j = P_{il}, l \in \left[1, G_i\right], \tag{5.28}$$

Како би предложени оптимални план терминирања био остварив током процеса оптимизације, неопходно је поставити одређене услове које сваки план терминирања мора да задовољи. Прво је неопходно обезбедити да је само један алтернативни технолошки процес одабран за сваки део (5.29):

$$\sum_{l=1}^{G_i} X_{il} = 1, \forall i \in \left[1, N\right], \tag{5.29}$$

Затим, за сваку операцију у алтернативном технолошком процесу, мора се одабрати само једна машина (5.30), један алат (5.31) и једна оријентација алата (5.32) из датих скупова:

$$\sum_{k=1}^{M} Z_{ijl}^{k} = 1, \forall i \in [1, N],\ \forall j \in \left[1, P_{ij}\right], \forall l \in [1, G_i], \tag{5.30}$$

$$\sum_{p=1}^{L_{ijl}} \Psi_{ijl}^{p} = 1, \forall i \in [1, N],\ \forall j \in \left[1, P_{ij}\right], \forall l \in [1, G_i], \tag{5.31}$$

$$\sum_{q=1}^{D_{ijl}} \Xi_{ijl}^{q} = 1, \forall i \in [1, N],\ \forall j \in \left[1, P_{ij}\right], \forall l \in [1, G_i], \tag{5.32}$$

Ради краћег записа, све информације везане за технолошке ресурсе интегрисане су у променљиву дефинисану једначином (5.33):

$$\Omega_{ijl}^{kpq} = \Xi_{ijl}^{q} \cdot \Psi_{ijl}^{p} \cdot Z_{ijl}^{k} \cdot X_{il}, \tag{5.33}$$

Како би се обезбедило да две узастопне операције једног дела не могу да се изврше у исто време, дато је следеће ограничење (5.34):

$$\begin{aligned} &\left(c_{ijl}^{k} \cdot \Omega_{ijl}^{kpq}\right) - \left(c_{i(j-1)l}^{k''} \cdot \Omega_{i(j-1)l}^{kpq}\right) + A\left(1 - \Omega_{ijl}^{kpq}\right) \geq Jwt_{ijl} + \left(\Theta_{ijl} \cdot T_{i(j-1)lijl}^{k''k}\right) + Mct_{ijl} + \left(t_{ijl}^{kpq} \cdot \Omega_{ijl}^{kpq}\right), \\ &\forall i \in [1, N], \forall j \in \left[2, P_{ij}\right], \forall l \in [1, G_i], k, k'' \in [1, M] \end{aligned}, \tag{5.34}$$

где је са A представљена велика целобројна вредност.

Ограничење везано за чињеницу да наредна операција на машини k не може да почне уколико претходна операција на тој исто машини није завршена (5.35):

$$\begin{aligned} &\left(s_{ijl}^{k} \cdot \Omega_{ijl}^{kpq}\right) - \left(c_{i'j'l'}^{k} \cdot \Omega_{i'j'l'}^{kpq}\right) + A\left(1 - \Omega_{ijl}^{kpq}\right) \geq 0, \forall k \in [1, M], \\ &\forall i, i' \in [1, N], \forall j, j' \in \left[1, P_{ij}\right], \forall l \in [1, G_i], \end{aligned}. \tag{5.35}$$

Како би се обезбедило да се, уколико се две узастопне операције дела i извршавају на различитим машинама, користи и припремно време машине, предложено је ограничење (5.36):

$$\begin{aligned} &\left(s_{ijl}^{k} \cdot \Omega_{ijl}^{kpq}\right) - \left(c_{i(j-1)l}^{k''} \cdot \Omega_{i(j-1)l}^{kpq}\right) + A\left(1 - \Omega_{ijl}^{kpq}\right) \geq Jwt_{ijl} + \left(\Theta_{ijl} \cdot T_{i(j-1)lijl}^{k''k}\right) + Mct_{ijl}, \\ &\forall i \in [1, N], \forall j \in \left[2, P_{ij}\right], \forall l \in [1, G_i], k, k'' \in [1, M], k \neq k'' \end{aligned}. \tag{5.36}$$

Ограничење које обезбеђују да робот не преузме део док се претходна операција није завршила дато је (5.37):

$$\begin{aligned} &tr_{ijl} + A\left(1 - \Omega_{ijl}^{kpq}\right) \geq \left(c_{i(j-1)l}^{k''} \cdot \Omega_{i(j-1)l}^{kpq}\right) + Jwt_{ijl} + \Theta_{ijl} \cdot T_{i(j-1)lijl}^{k''k}, \\ &\forall i \in [1, N], \forall j \in \left[2, P_{ij}\right], \forall l \in [1, G_i], k, k_{\mathrm{r}}, k'' \in [1, M] \end{aligned}. \tag{5.37}$$

Како не би било могуће да мобилни робот почне извршавање наредног транспортног задатка пре него што је претходни завршен, дато је ограничење (5.38):

$$\begin{aligned} &tr_{ijl} - tr_{i'j'l'} + A\left(1 - \Omega_{ijl}^{kpq}\right) \geq Rwt_{ijl} + \Theta_{ijl} \cdot \left(T_{i(j-1)lijl}^{k''k} + R_{ijlk_r} \cdot T_{ijli(j-1)l}^{k_r k''}\right), \\ &\forall i \in [1, N], \forall j \in \left[2, P_{ij}\right], \forall l \in [1, G_i], k, k_{\mathrm{r}}, k'' \in [1, M] \end{aligned}. \tag{5.38}$$

Ограничење које не дозвољава мобилном роботу да непотребно чека дато је (5.39):

$$\begin{aligned} &Rwt_{ijl} + A\left(1-\Omega_{ijl}^{kpq}\right) \geq c_{i(j-1)l}^{k''} - tr_{i'j'l'} - R_{ijlk_r} \cdot T_{ijli(j-1)l}^{k_r k''}, \\ &\forall i \in [1,N], \forall j \in \left[2,P_{ij}\right], \forall l \in [1,G_i], k,k_{\mathrm{r}},k'' \in [1,M] \end{aligned}. \tag{5.39}$$

5.4. Вишекритеријумска оптимизација планова терминирања

Вишекритеријумска оптимизација подразумева решавање проблема у којем је потребно задовољити различите супростављене критеријуме. Током оптимизације процеса или производа постоје супротстављени критеријуми (нпр. цена и квалитет производа) који морају бити задовољени. Ти критеријуми у области оптимизације биће математички моделирани функцијама циља, док се применом процеса оптимизације тежи њиховим оптималним вредностима. Током процеса вишекритеријумске оптимизације потребно је извршити минимизацију обједињене функције (5.40) циља:

$$F(\mathbf{x}) = \left\{f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_w(\mathbf{x})\right\}, \tag{5.40}$$

која подлеже следећим ограничењима (5.41-43):

$$g_i(\mathbf{x}) \geq 0,\ i = 1,...,c \tag{5.41}$$

$$h_i(\mathbf{x}) = 0,\ i = 1,...,v \tag{5.42}$$

$$L_i \leq x_i \leq U_i,\ i = 1,...,b \tag{5.43}$$

где је са w представљен број функција циља, b је укупан број примењљивих, c и v су укупни бројеви ограничења, док је са g_{i} и h_{i} представљено i-то ограничење. Са L_{i} и U_{i} представљене су доња и горња граница опсега променљивих.

У оквиру једнокритеријумске оптимизације, на основу вредности функције циља једноставно је одредити које решење из скупа је оптимално. Међутим, код вишекритеријумске оптимизације уводи се појам Парето доминантности ($\prec$) [129]. Решење **x** је доминантно у односу на решење **y** уколико су све вредности функција циља решења **x** мања или једнака (у случају када се ради минимизација свих функција циља) од вредности функција циља решења **y**, што се може записати једначином (5.44):

$$\mathbf{x} \prec \mathbf{y} \text{ if } f_i(\mathbf{x}) \leq f_i(\mathbf{y}), \forall i \in (1,...,w) \tag{5.44}$$

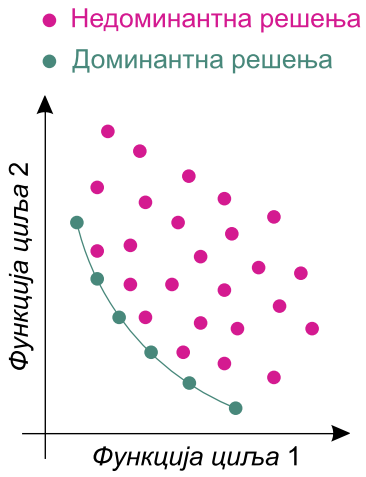
У оквиру скупа свих решења која се добијају на основу процеса оптимизације, свако решење над којим не доминира ни једно друго решење припада подскупу названом Парето оптимална решења (доминанта решења) (5.45):

$$P_s := \left\{\mathbf{x}, \mathbf{y} \in X \mid \nexists F(\mathbf{y}) \prec F(\mathbf{x})\right\} \tag{5.45}$$

Подскуп Парето оптималних решења користи се за формирање Парето фронта, који представља излаз вишекритеријумске оптимизације (5.46):

$$P_f := \left\{F(\mathbf{x}) \mid \mathrm{x} \in P_s\right\} \tag{5.46}$$

Сва решења у Парето фронту (слика 5.7) сматрају се оптималним и које решење ће бити одабрано за имплементацију у реалном систему, зависи од тренутног стања система.



Слика 5.7. Пример Парето фронта са доминантним решењима [147].

При поређењу више алгоритама у области вишекритеријумске оптимизације, сваки алгоритам генерише Парето фронт на крају оптимизације па је потребно дефинисати метрике на основу којих је могуће поредити перформансе алгоритама. Четири метрике које се користе у оквиру предметне докторске дисертације подељене су у две групе, метрике које се односе на конвергентне карактеристике алгоритама и метрике за покривеност простора решења. Дистанца (енгл. *Generational Distance* – GD [148]) и инверзна дистанца (енгл. *Inverted Generational Distance* – IGD [148]) су метрике које се користе за одређивање карактеристика конвергентности решења. Оне се одређују на основу референтних, тј. најбољих решења. Насупрот њима, метрике за одређивање покривености простора решења су одстојање (*Spacing* – SP [149]) и максимално растојање (*Maximum Spread* – MS [149]) и оне се одређују искључиво на основу решења у Парето фронту. Математичка формулација метрике GD дефинисана је једначином (5.47):

$$GD = \frac{1}{n}\sqrt{\left(\sum_{i=1}^{n} d_i^2\right)}. \tag{5.47}$$

где је са d_i дефинисана Еуклидска дистанца i-тог решења у Парето фронту од најближе референте тачке, а n је број решења у Парето фронту. Метрика IGD дефинисана је на основу једначине (5.48):

$$IGD = \frac{1}{m}\sqrt{\left(\sum_{j=1}^{m} \hat{d}_j^2\right)}. \tag{5.48}$$

где је са $\hat{d}_j^2$ дефинисана Еуклидска дистанца од j-те референте тачке до најближег решења у Парето фронту, а са m је представљен број референтних тачака. Метрика SP (5.49) користи се за одређивање покривености простора решења добијеног Парето фронта,

$$SP = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(da_i - \overline{da}\right)^2}, \tag{5.49}$$

$$da_i = \min\left(\sum_{m \in M_f} \left|f_m^i - f_m^u\right|\right), \forall u \in [1,n], u \neq i, \tag{5.50}$$

$$\overline{da} = \frac{1}{n}\sum_{i=1}^{n} da_i, \tag{5.51}$$

где је са da_i (5.50) представљена минимална апсолутна дистанца између i-тог и свих осталих решења у Парето фронту, док је са $\overline{da}$ (5.51) представљена средња вредност параметра da_i за сва решења, M_f је сет одабраних функција циља које се користе за вишекритеријумску оптимизацију, а f_m^i представља вредност m-те функције циља i-тог решења у Парето фронту. Последња метрика MS (5.52) одређује се као сума максималних Еуклидских дистанци између решења у Парето фронту по свакој функцији циља:

$$MS = \sqrt{\sum_{m \in M_f} \left(\max_{i=1}^{n} \left(f_m^i \right) - \min_{i=1}^{n} \left(f_m^i \right) \right)^2} . \tag{5.52}$$

5.5. Алгоритам инспирисан интелигенцијом чопора вукова

Алгоритам инспирисан интелигенцијом чопора вукова (енгл. *Grey Wolf Optimizer* – GWO [131]) је метахеуристички алгоритам чији је основни принцип оптимизације инспирисан начином лова чопора сивих вукова. GWO је алгоритам у оквиру кога се на основу интеракције између јединки у популацији које представљају решења оптимизационог проблема, током одређеног броја итерација, усваја оптимално решење. У оквиру чопора постоје три доминантна вука названа α – алфа, β – бета, δ – делта, који у оквиру оптимизационог алгоритма представљају лидере, тј. три најбоља решења оптимизационог проблема. Кретање свих осталих вукова у потрази за пленом зависиће од ова три лидера. Дистанца у простору решења између три лидера и свих осталих решења дефинисана је једначинама (5.53-55):

$$\mathbf{D}_\alpha^t = \left| \mathbf{C}_1 \cdot \mathbf{X}_\alpha^t - \mathbf{X}^t \right| \tag{5.53}$$

$$\mathbf{D}_\beta^t = \left| \mathbf{C}_2 \cdot \mathbf{X}_\beta^t - \mathbf{X}^t \right| \tag{5.54}$$

$$\mathbf{D}_\delta^t = \left| \mathbf{C}_3 \cdot \mathbf{X}_\delta^t - \mathbf{X}^t \right| \tag{5.55}$$

где су са $\mathbf{X}_\alpha^t, \mathbf{X}_\beta^t$ и $\mathbf{X}_\delta^t$ представљена три најбоља решења (лидера), тј. позиције вукова α, β, и δ у простору решења, док $\mathbf{X}^t$ представља позицију вука у простору решења који се тренутно ажурира. Такође, $\mathbf{D}_\alpha^t$, $\mathbf{D}_\beta^t$ и $\mathbf{D}_\delta^t$ су дистанце тренутног вука од сваког лидера, а $\mathbf{C}_1, \mathbf{C}_2$ и $\mathbf{C}_3$ представљају векторе случајних бројева дефинисане једначином (5.56):

$$\mathbf{C}_i = 2\mathbf{r}_2, \; i = \{1,2,3\}, \tag{5.56}$$

где је са $\mathbf{r}_2$ представљен вектор случајних бројева у опсегу [0, 1]. На основу параметра **A** (једначина 5.57), дефинише се да ли ће у одређеној итерацији бити изведена фаза експлорације или експлоатације:

$$\mathbf{A} = 2\boldsymbol{a} \cdot \mathbf{r}_1 - \boldsymbol{a} , \tag{5.57}$$

где је са $\mathbf{r}_1$ представљен вектор случајних бројева у опсегу [0, 1], а параметар $\boldsymbol{a}$ линеарно се смањује од 2 до 0 са бројем итерација оптимизационог алгоритма. Уколико је параметар **A** већи од 1, извршава се експлорација, док се у супротном извршава експлоатација. У првих 75% итерација оптимизационог алгоритма могуће је, на основу вредности случајног вектора $\mathbf{r}_1$, одабрати било коју фазу, док је у последњих 25% итерација извршава искључиво експлоатација. У оквиру експлорације, вукови претражују простор решења у потрази за добрим решењем, док се у оквиру експлоатације користи позиција три најбоља вука да би се на основу њих остварило боље решење. Нова позиција вука који се тренутно ажурира дефинисана је на основу једначина (5.58-61):

$$\mathbf{X}_1^t = \mathbf{X}_\alpha^t - \mathbf{A}_1 \cdot \mathbf{D}_\alpha^t \tag{5.58}$$

$$\mathbf{X}_2^t = \mathbf{X}_\beta^t - \mathbf{A}_2 \cdot \mathbf{D}_\beta^t \tag{5.59}$$

$$\mathbf{X}_3^t = \mathbf{X}_\delta^t - \mathbf{A}_3 \cdot \mathbf{D}_\delta^t \tag{5.60}$$

$$\mathbf{X}^{t+1} = \frac{\mathbf{X}_1^t + \mathbf{X}_2^t + \mathbf{X}_3^t}{3} \tag{5.61}$$

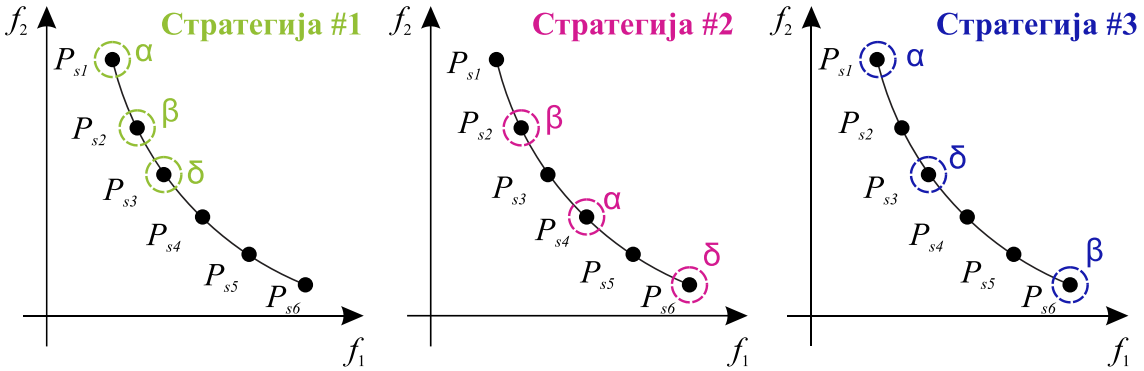
5.6. Примена алгоритма инспирисаног интелигенцијом чопора вукова у вишекритеријумском терминирању једног мобилног робота

Како би се GWO алгоритам применио у вишекритеријумском терминирању једног мобилног робота, потребно је извршити унапређење алгоритма. Првенствено, потребно је одредити процедуру за одређивање три најбоља решења (α, β, и δ вукове). Како су сва решења која су тренутно у Парето фронту оптимална, не могу се рангирати и одабрати три најбоља. У ту сврху, предложене су три стратегије за одабир три тренутно најбоља решења [11].

Прва стратегија (стратегија #1 на слици 5.8) ослања се на алгоритам за сортирање и у оквиру њега бира прва три решења која су укључена у Парето оптимални скуп. Уколико у тренутном Парето фронту нема три решења, узимају се сва која постоје, а остала се бирају из сортиране популације тако да одабрана решења имају минимални број јединки које их доминирају.

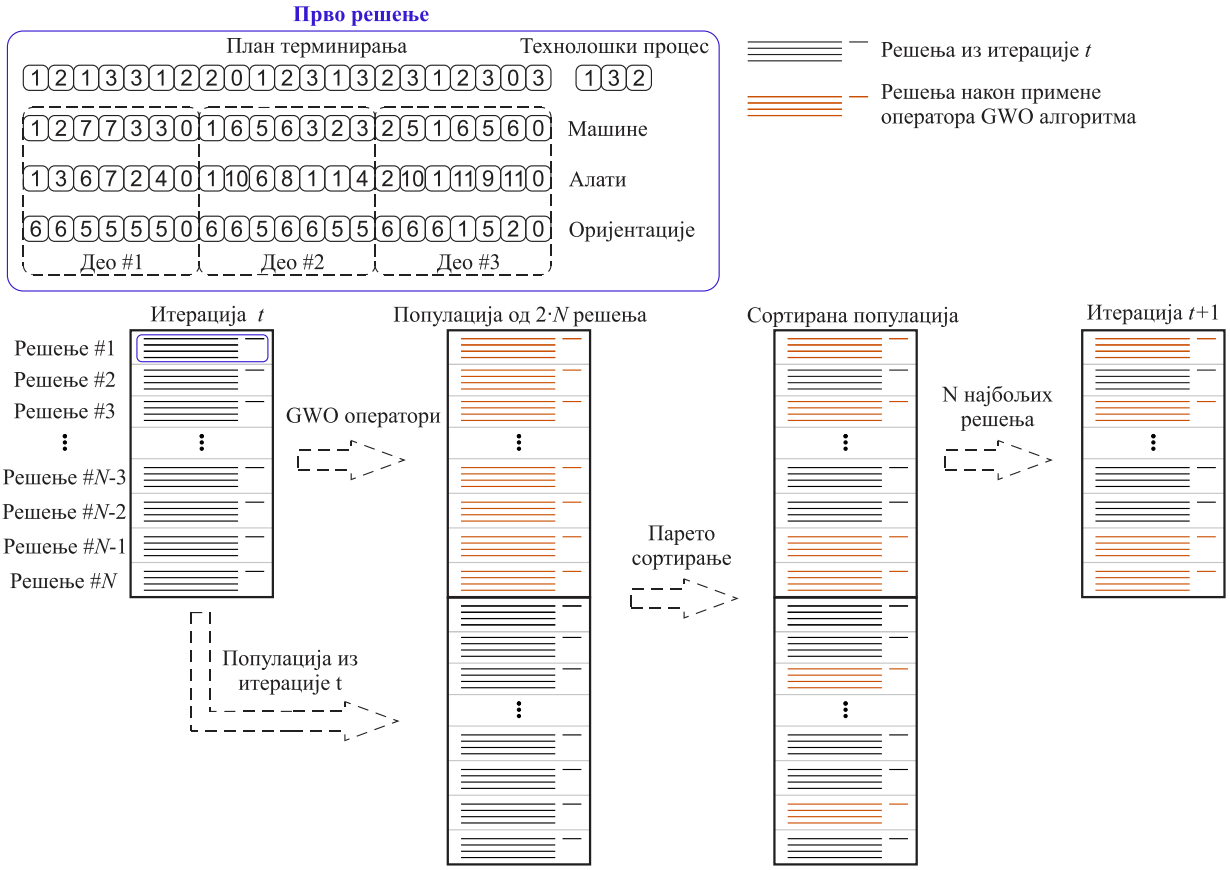
У оквиру друге стратегије (стратегија #2 на слици 5.8) примењена је стохастика, на основу које се случајно бирају три решења из Парето фронта. Сва решења имају једнаку вероватноћу да буду одабрана.

Трећа стратегија (стратегија #3 на слици 5.8) подразумева да се бирају два решења са максималним вредностима индивидуалних функција циља, док се треће решење бира случајно из остатка Парето фронта. Током селекције трећег решења, сва преостала решења имају исту вероватноћу да буду одабрана.



Слика 5.8. Стратегије за одабир лидера код GWO алгоритма.

Процес вишекритеријумске оптимизације за случајно одабрану популацију решења, модификовану на основу GWO алгоритма, дат је на слици 5.9 [11]. Као што се може приметити, иницијална популација је удвостручена у току процеса оптимизације, тако што се првих N решења модификује на основу GWO алгоритма, а других N решења проследи непромењено из иницијалне популације. Затим се извршава Парето сортирање како би се одредило најбољих N решења која ће бити усвојена као популација за наредну итерацију.



Слика 5.9. Процес модификације решења током једне генерације GWO алгоритма.

Псеудокод GWO алгоритма за вишекритеријумско терминирање једног мобилног робота дат је у табели 5.1.

Табела 5.1. Псеудокод GWO алгоритма за вишекритеријумско терминирање.	
1.	**Улаз**: Подаци о проблему Подешавање параметара алгоритма
2.	Иницијализација почетне популације
3.	Израчунавање функција циља за свако решење и Парето сортирање популације
4.	**for #1** за сваку итерацију
5.	Селекција α, β, и δ вукова на основу одабране стратегије Израчунавање параметра ***a***
6.	**for #2** за свако решење у популацији
7.	Генерисати случајне векторе $\mathbf{r}_1$, $\mathbf{r}_2$, за све елементе стрингова за машине, алате и оријентације алата Израчунавање вредности **A** и **C**
8.	**for #3** за сваки елемент у стрингу
9.	Модификација елемената стринга за машине на основу α, β, и δ вукова
10.	Модификација елемената стринга за алате на основу α, β, и δ вукова
11.	Модификација елемената стринга за оријентације алата на основу α, β, и δ вука
12.	**endfor #3**
13.	**endfor #2**
14.	Провера да ли су сва решења у дозвољеном опсегу
15.	Спајање претходне популације и популације модификоване на основу корака 9, 10 и 11. Израчунавање функције циља свих решења и Парето сортирање укупне популације. Издвајање *N* најбољих решења.
16.	**endfor #1**
17.	Чување резултата (Парето фронт)

У оквиру првог корака псеудокода из табеле 5.1, подаци о проблему подразумевају: скуп делова, скуп алтернативних технолошких процеса делова, скуп машина алатки, алата, могућих оријентација алата, времена обраде за сваку операцију и времена транспорта између машина алатки, док подешавање параметара алгоритма подразумева: дефинисање функција циља које ће учествовати у вишекритеријумској оптимизацији, број понављања, величина популације, број итерација, стратегија одабира лидера GWO алгоритма, опсег за параметре **C** и ***a***. Почетна популација у другом кораку алгоритма иницијализује се случајним одабиром решења из скупа могућих, а детаљна процедура може се пронаћи у литератури [132]. Затим је потребно за свако решење израчунати све појединачне функције циља и на основу њих извршити Парето сортирање решења. Након дефинисања сортиране иницијалне популације, почиње процес оптимизације који се извршава претходно дефинисани број итерација. У свакој итерацији потребно је усвојити три најбоља решења и дефинисати их као α, β, и δ вукове. Сва решења се оптимизују на основу вредности ова три најбоља решења и самог решења. На крају једне итерације, потребно је проверити да нису за неко решење усвојене нерегуларне вредности и ако јесу модификовати их. Затим се пореде решења добијена у тренутној и претходној итерацији и најбољих N се усвајају као нова популација. На крају процеса оптимизације, чувају се сва решења која припадају Парето фронту у последњој итерацији.

5.7. Експериментална верификација оптимизације планова терминирања

Како различити параметри у оквиру метахеуристичких алгоритама могу значајно да утичу на квалитет остварених решења, први део овог потпоглавља посвећен је дефинисању и подешавању параметара. Такође, поред подешавања самих параметара, биће извршена идентична процедура на основу које се бира једна од три предложене стратегије за одабир оптималних решења код GWO алгоритма. Извршени су иницијални експерименти за одређивање оптималних вредности параметара ***a*** и ***C***. Ова два параметра директно утичу на удаљеност која одређује колико ће се положаји осталих вукова померити у простору параметара у односу на три лидера. Скуп вредности који је коришћен при промени вредности параметара ***a*** и ***C*** су $\{2, 3, 4\}$. Уколико параметар ***a*** има вредност 4, то утиче на опсег линеарног опадања вредности **A** од 4 до 0 са итерацијама, чиме се повећава утицај експлорације у односу на експлоатацију, док се променом опсега параметра ***C*** повећава удаљеност за коју ће се вукови померати ка лидерима. Извршен је иницијални експеримент на три репрезентативна дела са пет типова флексибилности, како би се одредили оптимални параметри GWO алгоритма. Оптималне добијене вредности су ***a*** = 2 и ***C*** = 2, а друга стратегија (стратегија #2) је усвојена као најбоља. Подешени су број итерација алгоритма и величина популације на 100. Тежински коефицијенти у функцији циља f_{11} су $w_1 = 0{,}5$; $w_2 = 0{,}4$ и $w_3 = 0{,}1$. Рок за израду делова 1, 2 и 3 подешен је на 50, 70 и 100 респективно (за f_{12}), рок за завршетак рада робота је 70 (за f_{13}). У табели 5.2 дате су комбинације функција циља које су коришћене за вишекритеријумску оптимизацију.

Табела 5.2. Комбинације функција циља за вишекритеријумску оптимизацију.		
Ознака	Функције циља	Називи функција циља
CF #1	f_1, f_3	*Makespan*, Време транспорта
CF #2	f_6, f_7	Време чекања делова, Укупно време завршетка обраде свих делова
CF #3	f_9, f_{11}	Укупно време завршетка транспорта за све делове, Цена транспорта
CF #4	f_3, f_{13}	Време транспорта, Време кашњења робота
CF #5	f_2, f_6	Време завршетка рада робота, Време чекања делова
CF #6	f_2, f_5	Време завршетка рада робота, Време чекања робота
CF #7	f_4, f_{10}	Искоришћење робота, Средње време завршетка транспорта за све делове
CF #8	f_5, f_8	Време чекања робота, Средње време завршетка обраде свих делова
CF #9	f_4, f_{12}	Искоришћење робота, Укупно кашњење делова

5.7.1. Први експеримент терминирања једног мобилног робота

У првом експерименту, извршено је терминирање технолошког система у оквиру кога се врши обрада три репрезентативна дела, а транспортне задатке извршава један мобилни робот. Сви подаци везани за делове, времена обраде свих операција, времена транспорта између машина алатки могу се пронаћи у литератури [150]. Предложени GWO алгоритам за вишекритеријумску оптимизацију поређен је са генетичким алгоритмом (енгл. *Genetic Algorithm* – GA), аритметичким алгоритмом (енгл. *Arithmetic Optimization Algorithm* – AOA) [151] и алгоритмом инспирисаним интелигенцијом роја честица (енгл. *Particle Swarm Optimization* – PSO) [152]. Анализирани алгоритми поређени су на основу свих девет комбинација функција циља (табела 5.2), а извршено је 30 понављања за сваки алгоритам. Алгоритми су поређени на основу четири предложене метрике, а резултати су дати у табелама 5.3-5.6. Такође, поред најбољих решења, анализирана је и средња вредност за сва понављања, најлошије понављање, као и стандардна девијација добијене вредности метрике. Такође, уколико неки алгоритам оствари оптимални Парето фронт, вредности његових метрика ће бити нула. Као што се може уочити из табеле 5.3, GWO алгоритам остварио је шест од девет најбољих решења у првом експерименту за метрику GD. Такође, GWO је остварио пет од девет најбољих просечних вредности. На основу метрике GD, може се закључити да се алгоритам GWO показао као најбољи. Код IGD метрике (табела 5.4), GA је остварио четири, PSO три, а GWO два најбоља резултата. За разлику од прве метрике, где се GWO показао као убедљиво најбољи алгоритам, код IGD метрике то није случај. Међутим, како се обе метрике GD и IGD односе на конвергентне карактеристике алгоритама, резултати се могу анализирати обједињено, на основу чега се може закључити да GWO представља алгоритам који има најбоље конвергентне карактеристике.

Табела 5.3. Резултати првог експеримента за GD метрику.												
Резултати	**CF #1**				**CF #2**				**CF #3**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,100	0,234	**0,054**	0,120	4,545	11,989	5,986	**4,239**	0,283	0,564	**0,196**	0,429
Медијана	0,093	0,227	0,038	0,117	4,251	12,459	5,389	2,990	0,232	0,580	0,199	0,446
Станд. дев.	0,050	0,043	0,060	0,106	2,063	4,453	4,003	4,173	0,152	0,158	0,068	0,142
Најлошији	0,183	0,310	0,272	0,445	9,389	18,725	17,431	14,447	0,571	0,917	0,313	0,663
Најбољи	**0**	0,175	0,001	0,006	2,033	4,578	0,561	**0,262**	0,053	0,190	**0**	0,090
	CF #4				**CF #5**				**CF #6**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,339	0,596	**0,239**	0,391	0,422	0,921	**0,308**	0,380	0,179	0,328	0,245	**0,193**
Медијана	0,293	0,586	0,231	0,407	0,383	0,929	0,314	0,378	0,184	0,324	0,229	0,179
Станд. дев.	0,143	0,175	0,090	0,150	0,172	0,224	0,147	0,214	0,061	0,057	0,120	0,062
Најлошији	0,722	1,044	0,509	0,554	0,836	1,311	0,650	0,742	0,265	0,476	0,570	0,286
Најбољи	0,169	0,306	0,042	**0,012**	0,170	0,436	0,070	**0,003**	0,046	0,234	**0,039**	0,088
	CF #7				**CF #8**				**CF #9**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,009	0,022	0,024	**0,006**	0,190	0,293	0,252	**0,131**	**0,006**	0,013	0,019	**0,006**
Медијана	0,009	0,022	0,024	0,006	0,178	0,290	0,240	0,106	0,006	0,012	0,017	0,006
Станд. дев.	0,003	0,008	0,008	0,002	0,073	0,069	0,089	0,066	0,001	0,006	0,010	0,002
Најлошији	0,016	0,040	0,043	0,012	0,396	0,442	0,444	0,344	0,011	0,028	0,048	0,010
Најбољи	0,006	0,009	0,012	**0,003**	0,099	0,185	0,116	**0,025**	0,004	0,005	0,006	**0,003**

Табела 5.4. Резултати првог експеримента за IGD метрику.												
Резултати	**CF #1**				**CF #2**				**CF #3**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,125	0,239	**0,110**	0,155	0,509	1,297	**0,496**	1,239	0,266	0,515	**0,183**	0,412
Медијана	0,124	0,236	0,105	0,136	0,478	0,613	0,442	0,717	0,224	0,533	0,164	0,419
Станд. дев.	0,043	0,034	0,050	0,106	0,271	1,614	0,391	1,212	0,132	0,143	0,063	0,136
Најлошији	0,210	0,290	0,307	0,516	1,565	7,857	2,355	5,290	0,503	0,852	0,311	0,657
Најбољи	0,058	0,163	**0,032**	0,044	0,234	0,415	**0,155**	0,317	0,056	0,192	**0,005**	0,088
	CF #4				**CF #5**				**CF #6**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,283	0,597	**0,250**	0,413	0,314	0,702	**0,243**	0,420	0,192	0,312	**0,190**	0,238
Медијана	0,259	0,584	0,239	0,454	0,349	0,724	0,229	0,399	0,209	0,323	0,183	0,254
Станд. дев.	0,101	0,132	0,081	0,147	0,105	0,126	0,107	0,231	0,060	0,065	0,072	0,069
Најлошији	0,623	0,850	0,562	0,589	0,498	0,932	0,499	0,900	0,286	0,409	0,380	0,351
Најбољи	0,141	0,339	0,098	**0,046**	0,138	0,478	0,077	**0,036**	**0,072**	0,195	0,076	0,108
	CF #7				**CF #8**				**CF #9**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	**0,015**	0,035	0,047	0,025	0,259	0,321	**0,234**	0,334	**0,018**	0,050	0,070	0,038
Медијана	0,015	0,035	0,046	0,024	0,259	0,300	0,226	0,350	0,018	0,049	0,065	0,034
Станд. дев.	0,003	0,008	0,011	0,006	0,085	0,071	0,067	0,098	0,004	0,015	0,021	0,010
Најлошији	0,020	0,047	0,074	0,039	0,401	0,476	0,436	0,448	0,028	0,099	0,114	0,052
Најбољи	**0,009**	0,021	0,021	0,018	**0,118**	0,219	0,120	0,125	**0,013**	0,029	0,033	0,024

Насупрот прве две метрике које су коришћене за мерење конвергентности, друге две метрике користе се за анализу покривеност простора добијених решења у Парето фронту. Тачније, овим метрикама генерише се информација о различитости добијених решења, а тиме и покривености простора решења Парето фронтом. Резултати анализираних алгоритама за метрике SP и MS дати су у табелама 5.5 и 5.6, респективно. На основу резултата добијених метриком SP, алгоритам GWO остварује значајно боље резултате од осталих алгоритама и то

се огледа у добијених осам од девет најбољих резултата, као и осам од девет најбољих средњих резултата. Уколико је у табелама 5.5 и 5.6 неки резултат означен са N/A, то значи да је у финалном Парето фронту остварено само једно решење, на основу чега није могуће одредити SP метрику. Такође, уколико неки алгоритам има вредност 0 за SP метрику, то значи да је минимална вредност растојања између решења једнака.

Табела 5.5. Резултати првог експеримента за SP метрику.

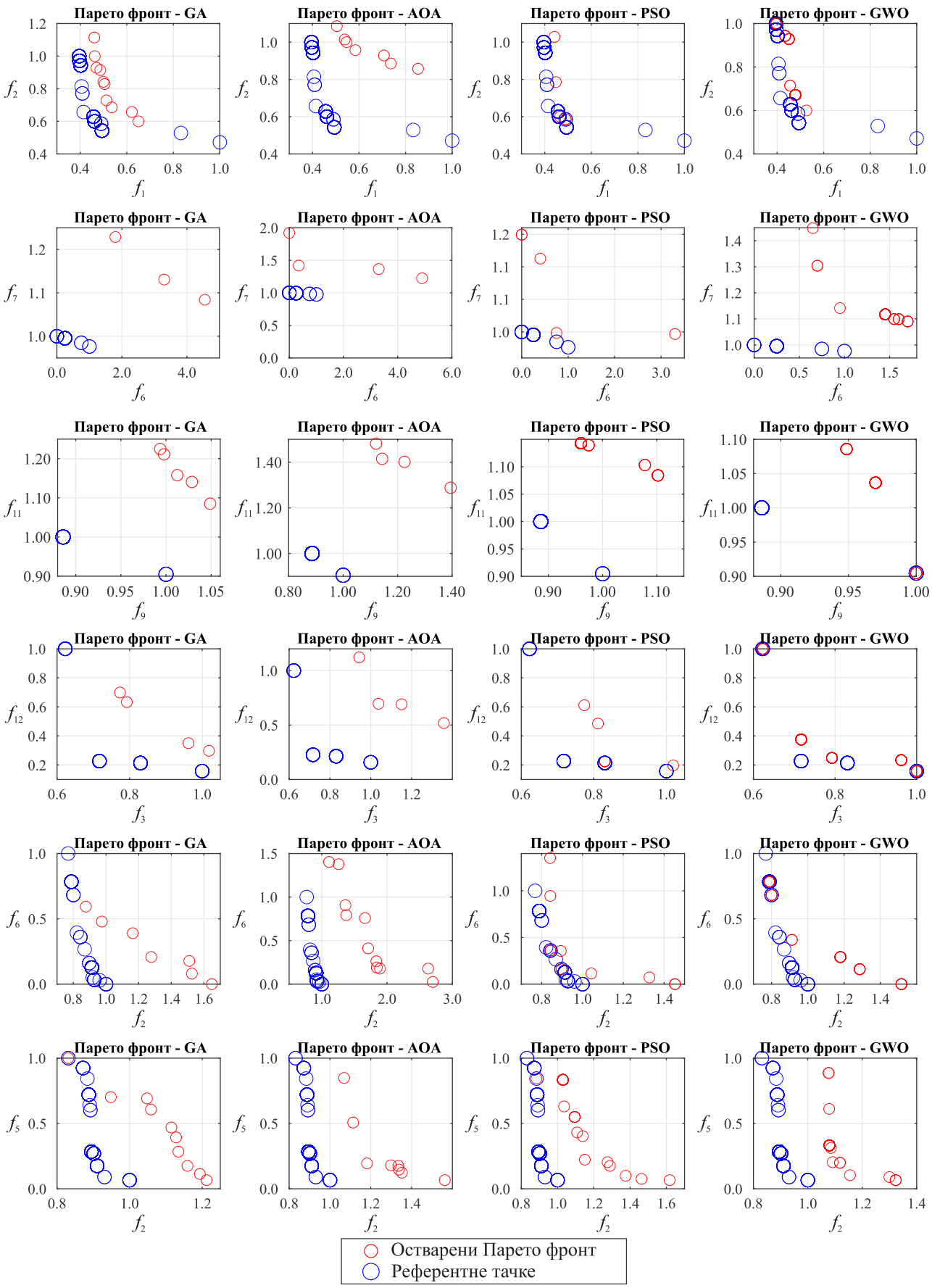
Резултати	**CF #1**				**CF #2**				**CF #3**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	N/A	0,064	0,029	**0,012**	2,046	3,072	1,297	**0,654**	N/A	N/A	**0,003**	0,008
Медијана	N/A	0,055	0,022	0,011	1,268	3,133	1,088	0,314	N/A	N/A	0	0
Станд. дев.	N/A	0,038	0,03	0,008	1,595	1,959	1,112	0,855	N/A	N/A	0,007	0,011
Најлошији	0,131	0,137	0,119	0,025	5,996	6,664	5,362	3,248	0,152	0,135	0,026	0,029
Најбољи	0,002	**0**	**0**	**0**	0,174	0,509	0,135	**0**	**0**	**0**	**0**	**0**
	CF #4				**CF #5**				**CF #6**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,135	N/A	0,035	**0,020**	0,182	0,218	0,136	**0,052**	0,120	0,102	0,129	**0,021**
Медијана	0,074	N/A	0,017	0,019	0,157	0,160	0,117	0,050	0,106	0,090	0,105	0,018
Станд. дев.	0,126	N/A	0,054	0,020	0,136	0,203	0,083	0,051	0,077	0,054	0,099	0,018
Најлошији	0,431	0,736	0,228	0,093	0,554	1,136	0,379	0,275	0,373	0,252	0,456	0,066
Најбољи	**0**	**0**	**0**	**0**	**0**	0,049	0,015	**0**	**0**	0,034	0,012	**0**
	CF #7				**CF #8**				**CF #9**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,013	0,027	0,027	**0,010**	0,164	0,113	0,071	**0,014**	0,014	0,020	0,022	**0,012**
Медијана	0,010	0,021	0,021	0,009	0,084	0,076	0,058	0,011	0,012	0,018	0,019	0,010
Станд. дев.	0,009	0,017	0,020	0,004	0,197	0,086	0,043	0,014	0,008	0,008	0,015	0,005
Најлошији	0,041	0,073	0,107	0,016	0,791	0,331	0,176	0,052	0,034	0,046	0,086	0,029
Најбољи	**0,004**	0,007	0,007	0,005	0,013	0,025	0,018	**0**	0,006	0,008	0,005	**0,004**

За метрику MS може се закључити да је алгоритам GWO остварио пет од девет најбољих резултата, као и четири од девет средњих најбољих резултата. Целокупном анализом метрика за одређивање покривености простора решења може се закључити да је алгоритам GWO остварио најбоље резултате. Разлог за значајно боља остварена решења алгоритма GWO, приписује се добро одабараној стратегији за селекцију лидера, чиме се обезбеђује равномерно претраживање простора решења.

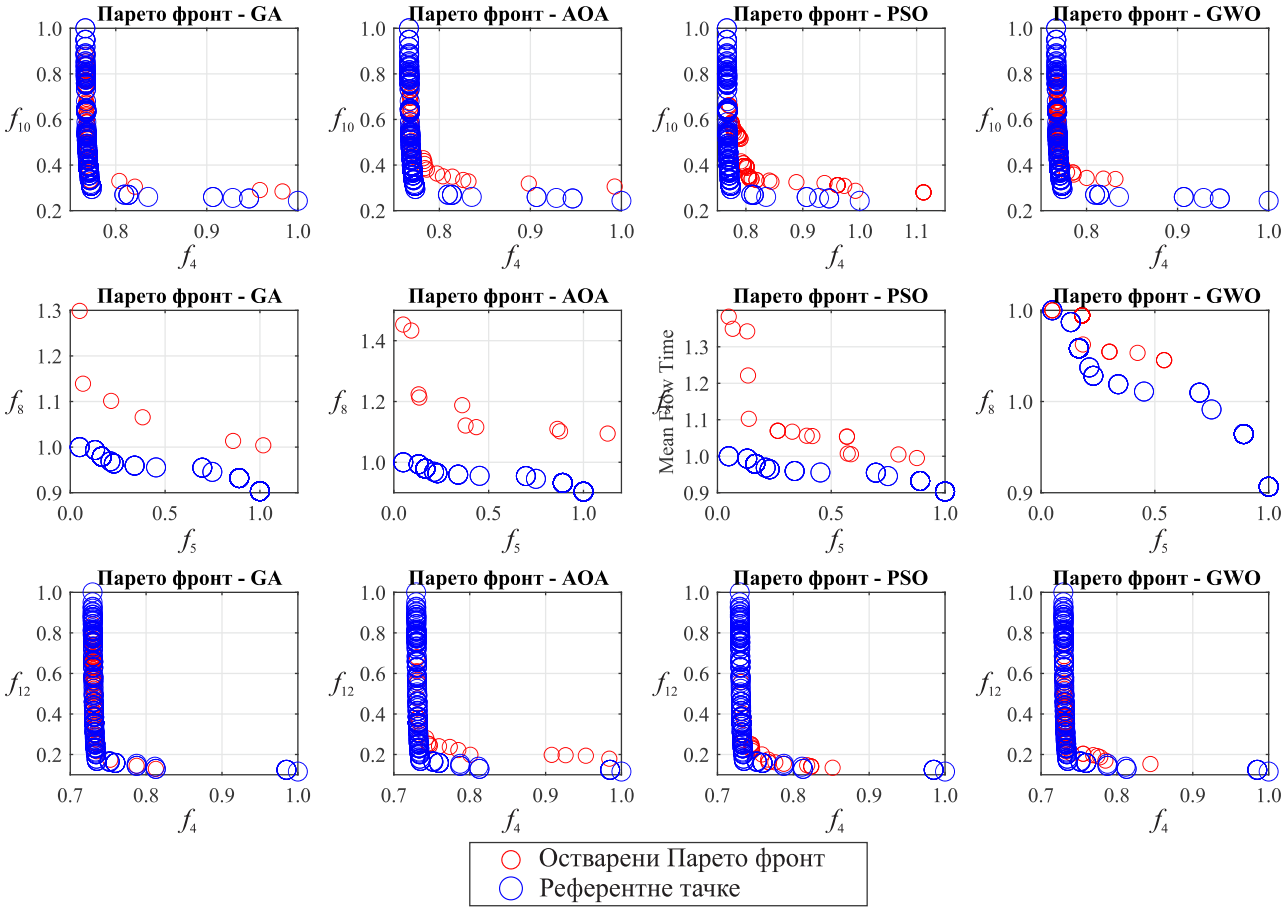
Табела 5.6. Резултати првог експеримента за MS метрику.												
Резултати	**CF #1**				**CF #2**				**CF #3**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,419	0,571	**0,289**	0,516	**14,833**	29,099	16,789	15,426	0,172	0,235	**0,160**	0,193
Медијана	0,474	0,521	0,245	0,514	15,702	30,384	16,454	18,329	0,161	0,271	0,124	0,210
Станд. дев.	0,208	0,162	0,126	0,156	6,439	10,165	9,009	10,537	0,125	0,155	0,080	0,122
Најлошији	0,724	0,949	0,715	0,859	26,659	44,801	37,451	33,260	0,556	0,534	0,323	0,353
Најбољи	**0**	0,383	0,129	0,278	2,754	8,702	2,950	**0,901**	**0**	**0**	**0**	**0**
	CF #4				**CF #5**				**CF #6**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,941	0,911	**0,656**	1,056	1,653	2,593	1,635	**1,303**	0,863	0,846	1,222	**0,507**
Медијана	0,806	0,789	0,538	1,015	1,728	2,465	1,541	1,341	0,882	0,765	1,124	0,467
Станд. дев.	0,553	0,460	0,338	0,265	0,609	0,809	0,480	0,595	0,368	0,300	0,377	0,300
Најлошији	2,201	2,078	1,848	1,408	2,754	4,073	2,699	2,543	1,717	1,522	1,865	1,261
Најбољи	0,077	**0**	0,166	0,288	0,517	1,104	0,708	**0,372**	0,163	0,426	0,309	**0,076**
	CF #7				**CF #8**				**CF #9**			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
Средњи	0,616	0,509	0,506	**0,459**	0,790	0,811	0,935	**0,338**	0,729	0,521	**0,499**	0,551
Медијана	0,599	0,510	0,497	0,449	0,683	0,674	0,897	0,302	0,719	0,513	0,482	0,558
Станд. дев.	0,078	0,083	0,098	0,067	0,434	0,404	0,300	0,211	0,080	0,063	0,114	0,059
Најлошији	0,799	0,708	0,755	0,624	1,800	1,512	1,649	0,826	0,964	0,634	0,755	0,643
Најбољи	0,505	0,401	**0,326**	0,371	0,281	0,274	0,313	**0,083**	0,601	0,379	**0,338**	0,466

На сликама 5.10 и 5.11, приказани су остварени Парето фронтови свих алгоритама за све комбинације функција циља, као и оптимално остварени (референтни) Парето фронт. Референтни Парето фронт добија се када се обједине сва решења која су сви анализирани алгоритми остварили и издвоје само оптимална решења. Као што се може уочити, фронтови GWO алгоритма су ближи оптималном фронту, док су решења у фронту униформно распоређена у простору решења. Како су различите функције циља коришћене у различитим комбинацијама (CF), број решења у Парето фронту значајно варира са комбинацијом функција циља. Такође, ради адекватног рачунања метрика, решења су линеарно скалирана на основу индивидуално најбољих вредности функција циља.

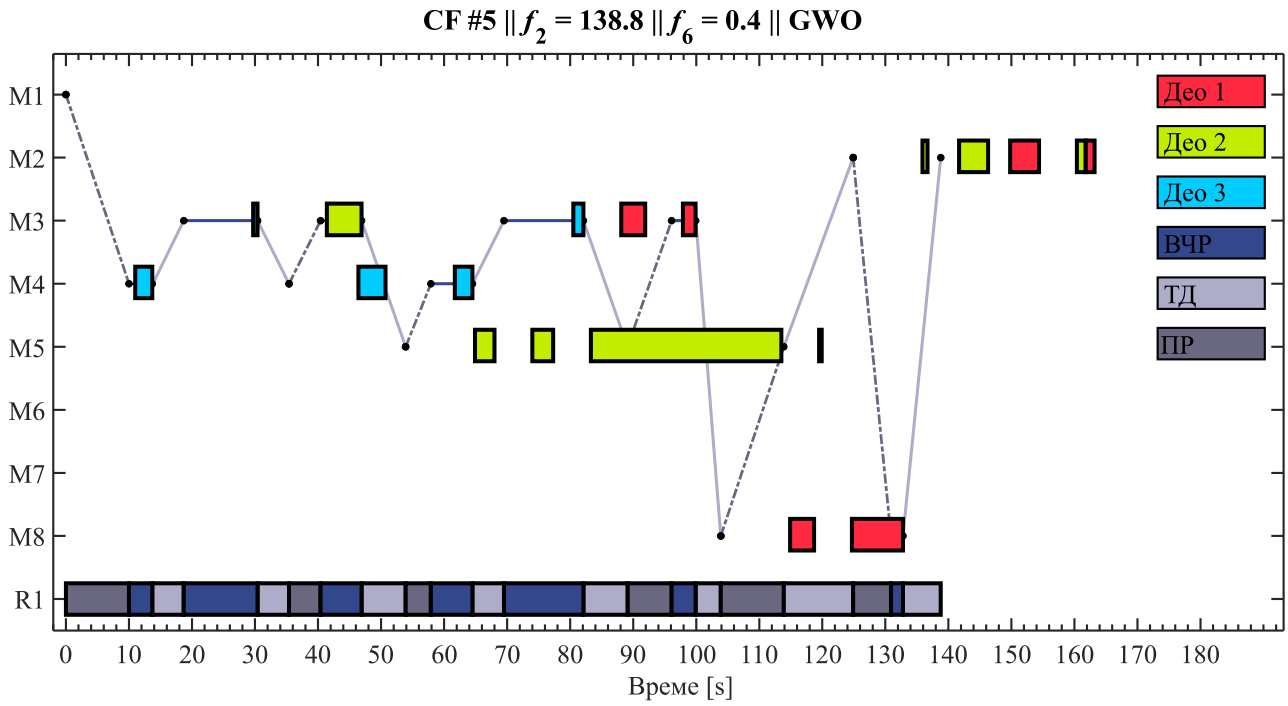
На крају, Гантови дијаграми одабраних решења у Парето фронту, за различите комбинације функција циља, дати су на сликама 5.12 и 5.13. Ови дијаграми представљају репрезентацију финалног решења плана терминирања које је одабрано за пласирање у технолошки систем у коме се налази мобилни робот.

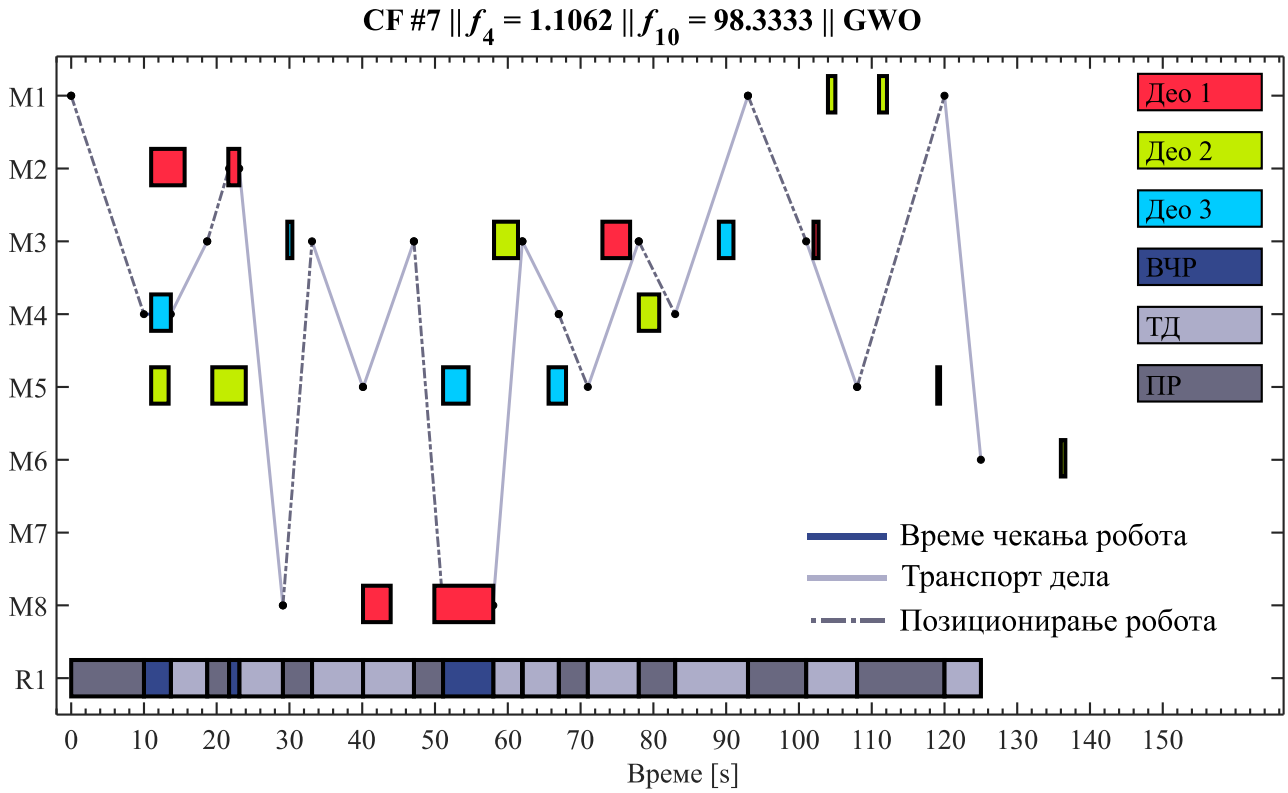


Слика 5.10. Парето фронтови за све алгоритме CF1-6.

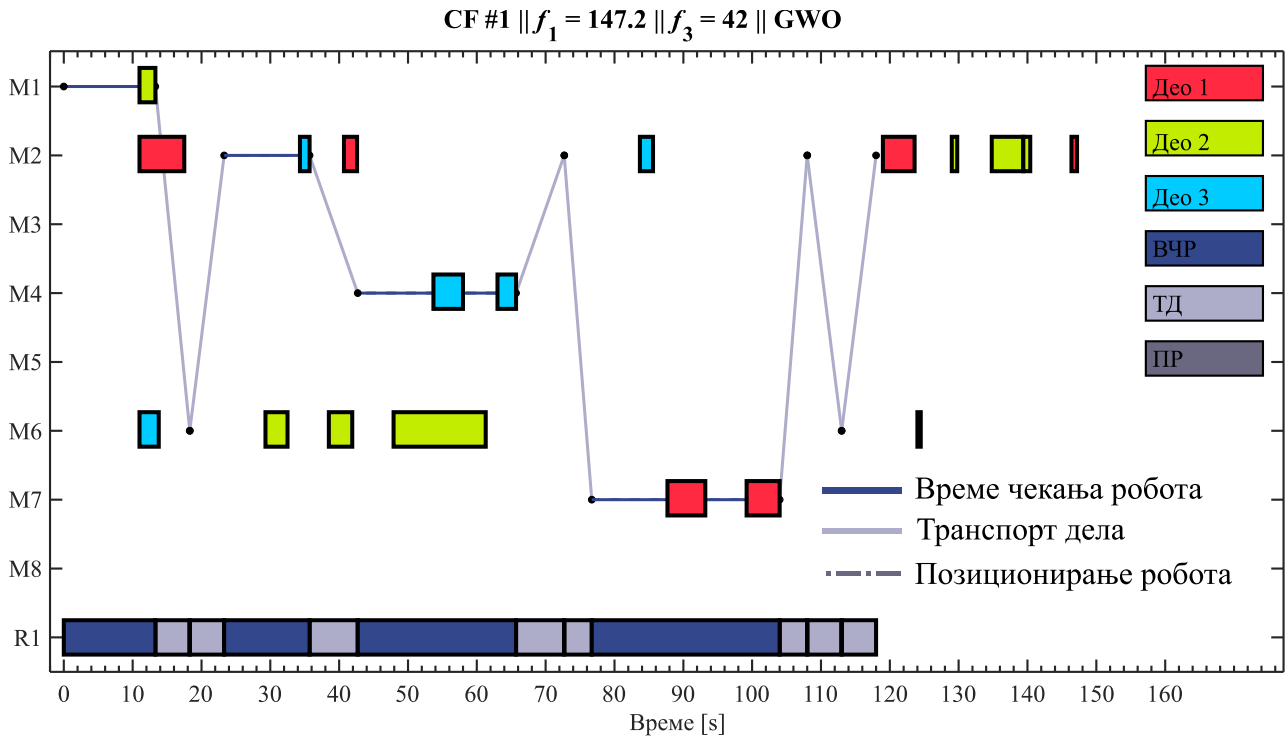


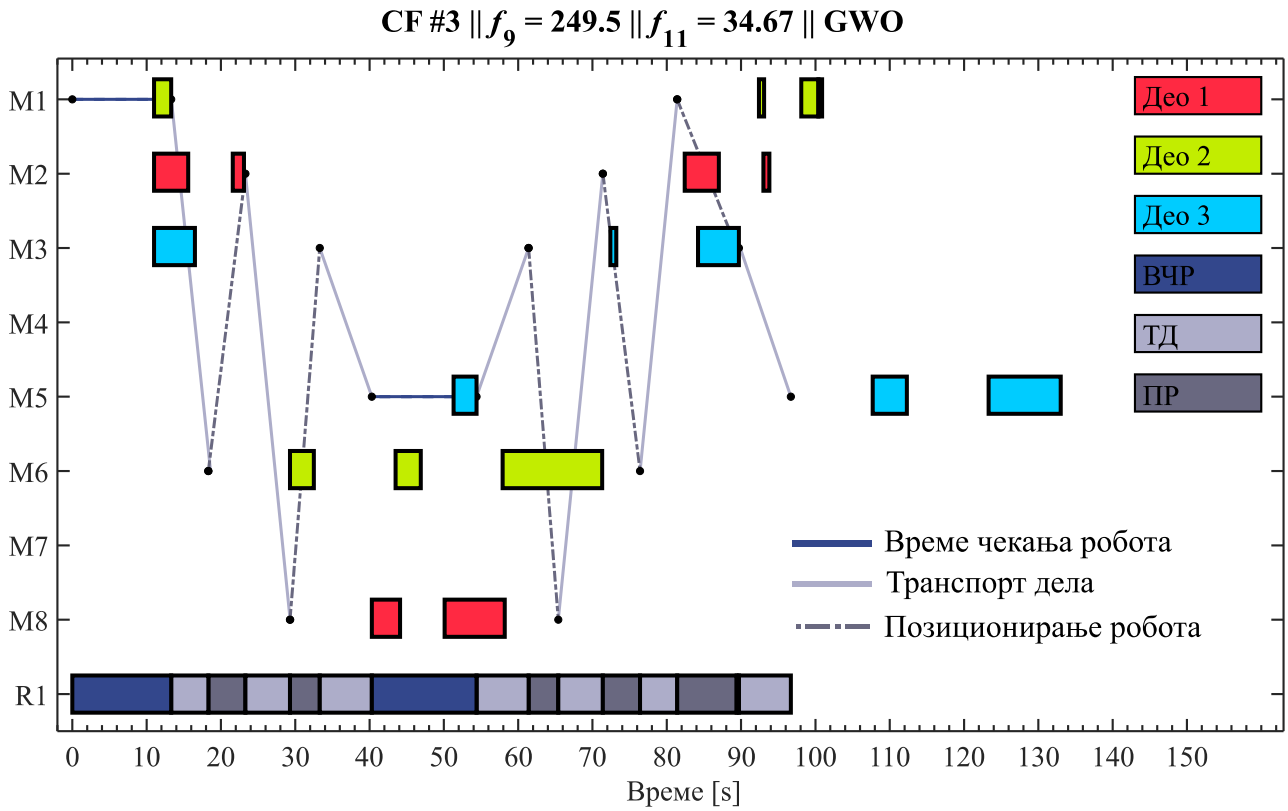
Слика 5.11. Парето фронтови за све алгоритме CF7-9.





Слика 5.12. Прва два Гантова дијаграма за одабрана оптимална решења.





Слика 5.13. Друга два Гантова дијаграма за одабрана оптимална решења.

5.7.2. Други експеримент терминирања једног мобилног робота

Други експеримент подразумева поређење анализираних алгоритама на 24 репрезентативна проблема који су формулисани комбиновањем 14 различитих делова. Комплетна експериментална поставка дата је у табели 5.7 [128], где се за сваки проблем може уочити који делови га чине, укупан број делова, као и укупан број операција. За разлику од првог експеримента где су делови имали пет типова флексибилности, у овом експерименту делови имају четири, односно немају флексибилност оријентације алата. Међутим, експериментална поставка другог експеримента је знатно комплекснија по броју делова, машина алатки, алата и операција, а такође је могуће поредити перформансе оптимизационих алгоритама и у зависности од броја делова.

Табела 5.7. Поставка другог експеримента

Проблем	Број делова	Број операција	Редни број делова за терминирање
1.	6	99	1-2-3-10-11-14
2.	6	106	4-5-6-11-14-15
3.	6	95	7-8-9-14-15-16
4.	6	94	1-4-7-10-14-16
5.	6	93	2-5-8-11-14-16
6.	6	107	3-6-9-11-15-16
7.	6	99	1-4-8-14-15-16
8.	6	100	2-6-7-10-14-16
9.	6	100	3-5-9-11-14-16
10.	9	154	1-2-3-5-6-10-11-14-15
11.	9	142	4-7-8-9-10-11-14-15-16
12.	9	134	1-4-5-7-8-10-11-14-16
13.	9	159	2-3-6-9-10-11-14-15-16
14.	9	143	1-2-4-7-8-11-14-15-16
15.	9	154	3-5-6-9-10-11-14-15-16
16.	11	189	1-2-3-4-5-6-10-11-14-15-16
17.	11	179	4-5-6-7-8-9-10-11-14-15-16
18.	11	181	1-2-4-5-7-9-10-11-14-15-16
19.	11	186	2-3-5-6-8-9-10-11-14-15-16
20.	11	179	1-2-4-6-7-8-10-11-14-15-16
21.	11	185	2-3-5-6-7-9-10-11-14-15-16
22.	11	188	2-3-4-5-6-8-9-10-11-14-16
23.	12	195	1-4-5-6-7-8-9-10-11-14-15-16
24.	14	233	1-2-3-4-5-6-7-8-9-10-11-14-15-16

У оквиру другог експеримента, за поређење анализираних алгоритама селектоване су функције које су примарно везане за параметре мобилног робота. Такође, по две метрике, једна која се односи на конвергентне карактеристике алгоритама, а друга за покривеност простора решења коришћене су за поређење алгоритама. Прво је изведен експеримент за комбинацију функција циља CF #6, где су анализирани алгоритми поређени на основу метрике IGD (табела 5.8) и MS (табела 5.9). Као што се може уочити из табеле 5.8, GWO је остварио 15/24 најбољих и 19/24 средњих најбољих резултата. За метрику MS, GWO и GA су имали приближан број најбољих резултата оба алгоритма остварују више од 20 најбољих решења. Међутим, алгоритам GWO остварио је далеко више (22/24) средњих најбољих резултата, чиме се поново потврђује његова предност у односу на остале анализиране алгоритме.

Табела 5.8. Резултати другог експеримента за CF #6 – IGD метрика.												
Проблем	Најбољи				Средњи				Станд. дев.			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
1.	0,488	1,028	0,283	**0,024**	**1,357**	1,520	2,005	4,704	1,288	0,720	2,380	2,849
2.	0,372	0,932	0,584	**0**	0,739	1,121	0,962	**0,501**	0,280	0,123	0,210	0,220
3.	0,283	0,413	0,313	**0,157**	0,944	0,760	0,658	**0,328**	0,688	0,386	0,316	0,178
4.	0,227	0,477	0,175	**0**	0,854	0,968	0,910	**0,793**	0,263	0,182	0,343	0,352
5.	0,245	0,412	0,265	**0,055**	0,385	0,550	0,459	**0,294**	0,073	0,071	0,108	0,096
6.	0,464	0,976	0,614	**0,235**	0,913	1,368	1,038	**0,601**	0,527	0,419	0,340	0,309
7.	0,148	0,397	0,239	**0,107**	0,295	0,513	0,473	**0,261**	0,081	0,081	0,148	0,084
8.	**0,128**	0,661	0,412	0,237	0,702	0,897	0,738	**0,449**	0,276	0,099	0,198	0,102
9.	0,574	0,888	0,602	**0,215**	0,898	1,197	1,080	**0,721**	0,221	0,290	0,284	0,273
10.	0,326	0,7	0,571	**0**	0,578	0,940	0,838	**0,445**	0,118	0,132	0,187	0,192
11.	**0,144**	0,331	0,282	0,188	**0,248**	0,451	0,428	0,269	0,055	0,059	0,098	0,076
12.	**0,119**	0,495	0,388	0,120	0,298	0,626	0,599	**0,247**	0,102	0,070	0,138	0,076
13.	0,341	0,706	0,360	**0,167**	0,562	0,937	0,729	**0,391**	0,159	0,116	0,203	0,097
14.	**0,205**	0,541	0,401	0,103	0,397	0,714	0,677	**0,304**	0,129	0,138	0,133	0,133
15.	0,580	1,670	0,852	**0,398**	2,954	3,547	3,640	**2,849**	1,670	2,058	2,456	2,394
16.	0,166	0,675	0,510	**0**	**1,168**	2,764	1,596	2,759	1,722	2,568	1,166	3,922
17.	**0,241**	0,684	0,535	0,041	0,528	0,951	1,116	**0,456**	0,145	0,134	0,469	0,234
18.	**0,337**	0,736	0,529	0,249	0,657	1,011	0,919	**0,496**	0,161	0,237	0,235	0,202
19.	**0,227**	0,603	0,548	0,104	0,383	0,816	0,750	**0,349**	0,100	0,093	0,108	0,113
20.	**0,145**	0,584	0,425	0,212	**0,794**	1,004	1,451	1,115	0,640	0,251	0,837	0,741
21.	0,444	0,982	0,595	**0,272**	0,822	1,243	1,166	**0,703**	0,178	0,176	0,261	0,217
22.	**0,121**	0,645	0,467	0,317	0,848	1,184	0,934	**0,631**	0,473	0,308	0,366	0,376
23.	0,169	0,4	0,371	**0,164**	0,408	0,754	0,626	**0,395**	0,137	0,109	0,111	0,162
24.	0,272	0,8	0,784	**0**	**1,752**	2,480	3,156	3,237	1,426	1,589	2,399	2,733

Табела 5.9. Резултати другог експеримента за CF #6 – MS метрика.												
Проблем	Најбољи				Средњи				Станд. дев.			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
1.	**0**	5,073	**0**	**0**	9,641	18,079	10,083	**5,906**	6,553	8,797	8,824	6,364
2.	0,071	0,381	0,208	**0**	0,802	1,794	1,545	**0,674**	0,504	0,881	0,787	0,529
3.	1,333	1,692	0,113	**0,112**	3,660	4,081	4,324	**1,636**	2,032	1,786	2,621	1,077
4.	**0**	**0**	**0**	**0**	1,522	1,991	**1,277**	1,382	1,335	1,519	1,254	0,96
5.	0,137	0,454	0,116	**0**	0,637	1,403	0,994	**0,494**	0,395	0,527	0,482	0,335
6.	**0**	**0**	**0**	**0**	1,852	3,632	2,263	**1,002**	1,377	2,239	1,291	0,935
7.	0,333	0,659	**0**	**0**	1,336	2,479	1,603	**0,873**	0,606	1,245	0,869	0,669
8.	**0**	**0**	**0**	**0**	0,466	1,136	1,104	**0,415**	0,517	0,425	0,675	0,479
9.	**0**	**0**	0,046	**0**	**0,531**	1,007	1,212	0,669	0,463	0,664	0,755	0,605
10.	**0**	**0**	**0**	**0**	0,705	2,163	1,390	**0,544**	0,657	1,121	1,091	0,478
11.	**0**	0,556	0,407	**0**	1,087	2,552	1,891	**0,783**	0,696	1,181	1,201	0,716
12.	**0**	0,311	0,400	0,034	0,539	1,338	1,181	**0,480**	0,366	0,595	0,608	0,402
13.	**0**	0,459	**0**	0,071	0,793	1,830	1,026	**0,710**	0,625	0,795	0,680	0,531
14.	**0**	0,277	**0**	**0**	0,851	2,345	1,379	**0,497**	0,543	1,219	0,966	0,444
15.	**0**	9,003	**0**	**0**	13,288	27,589	14,493	**6,248**	7,960	17,535	15,398	5,724
16.	**0**	**0**	**0**	**0**	8,869	26,872	16,555	**6,302**	6,750	15,478	10,203	10,577
17.	**0**	1,818	0,226	**0**	2,94,	7,525	6,342	**2,031**	2,179	3,285	3,77	1,349
18.	**0**	0,850	0	0,112	1,448	3,026	2,105	**0,935**	1,159	1,307	2,017	0,627
19.	**0**	0,534	0	**0**	0,592	1,898	1,079	**0,406**	0,464	1,035	0,760	0,277
20.	**0**	1,741	1,008	**0**	2,632	10,780	8,945	**1,413**	2,217	6,419	6,127	1,132
21.	**0**	0	0	**0**	0,732	1,283	1,050	**0,560**	0,418	0,692	0,727	0,570
22.	**0**	1,502	0	**0**	3,440	9,881	5,096	**2,172**	2,566	3,723	3,262	1,880
23.	**0**	1,099	0,368	**0**	1,320	3,069	1,783	**0,634**	0,858	1,144	1,120	0,475
24.	**0**	6,055	1,001	**0**	8,468	42,639	26,373	**8,078**	7,244	23,258	19,014	7,604

Статистички резултати везани за CF #7 одређени на основу метрике GD и SP представљени су у табелама 5.10 и 5.11. GWO алгоритам остварио је 12/24 најбоља резултата за GD метрику, чиме је показао значајно боље способности конвергенције у односу на алгоритме PSO и AOA, и сличне перформансе конвергенције у односу на GA алгоритам. Чињеница да су функције циља у оквиру CF #7 обе везане за мобилни робот (време завршетка рада робота и време чекања робота) указује на валидност примене алгоритма GWO за проблем терминирања једног мобилног робота. За метрику SP, GWO је остварио 19/24 најбоља резултата и тиме показао значајно боље карактеристике покривања простора решења од осталих алгоритама. Међутим, GA је остварио боља средња решења. Свеобухватно, може се закључити да је GWO алгоритам са најбољим перформансама за проблем терминирања једног мобилног робота и за комбинацију функција циља CF #7.

Табела 5.10. Резултати другог експеримента за CF #7 – GD метрика.

Проблем	Најбољи				Средњи				Станд. дев.			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
1.	**0**	0,401	0,180	0,135	**0,325**	0,602	0,672	0,591	0,219	0,152	0,457	0,430
2.	0,106	0,773	0,248	**0,002**	1,183	1,411	1,240	**0,479**	0,972	0,330	0,461	0,366
3.	0,067	0,295	0,211	**0,005**	0,324	0,519	0,556	**0,141**	0,228	0,205	0,219	0,131
4.	0,047	0,096	0,075	**0,043**	0,476	0,416	**0,405**	0,477	0,152	0,176	0,315	0,317
5.	0,131	0,523	0,293	**0,014**	0,713	0,795	0,790	**0,329**	0,376	0,231	0,308	0,270
6.	**0,040**	0,959	0,899	0,064	1,248	1,380	1,590	**0,732**	0,750	0,331	0,501	0,428
7.	0,063	0,299	0,107	**0,009**	0,356	0,550	0,630	**0,291**	0,160	0,184	0,247	0,183
8.	**0**	0,266	0,037	**0**	1,374	1,280	1,143	**0,855**	0,737	0,462	0,634	0,319
9.	**0**	0,346	0,309	0,021	1,275	0,757	0,849	**0,423**	0,729	0,328	0,423	0,363
10.	0,027	0,541	0,196	**0**	**0,309**	1,020	0,937	0,542	0,261	0,300	0,512	0,383
11.	**0,009**	0,578	0,432	0,141	0,769	0,839	0,762	**0,349**	0,392	0,168	0,193	0,158
12.	**0**	0,793	0,622	0,385	**0,925**	1,418	1,449	1,074	0,838	0,377	0,362	0,551
13.	**0**	0,823	0,503	0,094	1,030	1,606	1,592	**0,920**	0,856	0,472	0,752	0,468
14.	**0,001**	0,667	0,398	0,016	0,682	1,357	1,148	**0,664**	0,596	0,511	0,501	0,575
15.	**0,005**	0,717	0,268	0,083	0,662	1,489	1,286	**0,660**	0,601	0,513	0,661	0,535
16.	0,072	0,332	0,115	**0**	**0,414**	0,887	0,790	0,557	0,441	0,354	0,424	0,427
17.	0,030	1,103	0,649	**0,003**	0,902	2,092	1,505	**0,702**	0,789	0,860	0,494	0,681
18.	**0,001**	0,593	0,487	0,103	**0,700**	1,295	1,105	0,707	0,501	0,508	0,663	0,339
19.	0,082	0,761	0,077	**0**	0,908	1,354	1,159	**0,513**	0,749	0,311	0,548	0,539
20.	**0,015**	0,578	0,216	0,019	**0,432**	1,133	0,965	0,703	0,358	0,453	0,600	0,816
21.	0,060	0,793	0,280	**0**	1,019	1,355	1,133	**0,520**	0,956	0,657	0,460	0,411
22.	0,065	0,691	0,572	**0,001**	**0,673**	1,485	1,532	0,806	0,564	0,342	0,598	0,646
23.	**0,001**	0,584	0,454	0,018	0,881	1,276	1,039	**0,845**	0,904	0,341	0,478	0,704
24.	**0**	0,556	0,275	0,141	0,648	1,113	0,790	**0,589**	0,531	0,441	0,426	0,325

За комбинацију функција циља CF #8, алгоритам GWO показао је до сада најбоље резултате за IGD метрику (табела 5.12) тиме што је остварио 18/24 најбољих и 18/24 средњих најбољих резултата. Такође, сличан закључак може се извести и за MS метрику (табела 5.13) јер је остварио 21/24 најбоља и 22/24 средња најбоља резултата.

Табела 5.11. Резултати другог експеримента за CF #7 – SP метрика.												
Проблем	Најбољи				Средњи				Станд. дев.			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
1.	**0**	0,025	0,008	**0**	0,038	0,129	0,101	**0,033**	0,037	0,108	0,087	0,033
2.	0,024	0,037	0,010	**0,002**	0,071	0,162	0,281	**0,042**	0,041	0,097	0,242	0,031
3.	0,009	0,042	0,032	**0,006**	0,043	0,129	0,088	**0,027**	0,053	0,071	0,054	0,017
4.	**0**	0,022	0,001	0,001	**0,015**	0,081	0,074	0,033	0,016	0,050	0,088	0,038
5.	0,016	0,046	0,034	**0,005**	0,055	0,165	0,166	**0,030**	0,027	0,093	0,182	0,030
6.	**0**	0,059	0,073	0,017	**0,099**	0,279	0,214	0,105	0,068	0,179	0,123	0,125
7.	0,003	0	0	**0**	**0,023**	0,105	0,078	0,025	0,021	0,100	0,047	0,017
8.	**0**	0	0	**0**	**0,047**	0,149	N/A	0,048	0,024	0,135	N/A	0,044
9.	0,017	0,004	0,029	**0**	N/A	0,149	0,134	**0,060**	N/A	0,079	0,061	0,081
10.	**0,001**	0,027	0,042	0,003	**0,023**	0,198	0,148	0,071	0,023	0,150	0,112	0,092
11.	0,015	0,047	0,031	**0,007**	**0,042**	0,120	0,090	0,063	0,023	0,077	0,066	0,055
12.	0,006	0,029	0,024	**0,005**	**0,039**	0,216	0,218	0,121	0,025	0,220	0,306	0,293
13.	**0**	0,039	0,036	**0**	**0,103**	0,406	0,302	0,163	0,178	0,328	0,250	0,171
14.	0,003	0,065	0,032	**0**	**0,052**	0,254	0,306	0,082	0,059	0,182	0,640	0,191
15.	**0,002**	0,043	0,037	0,004	**0,059**	0,339	0,172	0,172	0,083	0,305	0,126	0,252
16.	**0**	0,023	**0**	**0**	**0,028**	0,144	0,138	0,152	0,048	0,151	0,134	0,298
17.	0,003	**0**	0,035	**0**	0,188	0,629	0,322	**0,132**	0,314	0,521	0,360	0,159
18.	0,005	0,034	0,016	**0,003**	0,072	0,355	0,251	**0,067**	0,220	0,332	0,443	0,110
19.	0,004	0,092	0,020	**0**	0,126	0,368	0,283	**0,106**	0,161	0,266	0,302	0,229
20.	0,004	0,016	0,034	**0,001**	**0,039**	0,302	0,243	0,053	0,047	0,448	0,340	0,078
21.	**0**	0,063	**0**	**0**	**0,077**	0,285	0,275	0,144	0,081	0,222	0,302	0,239
22.	0,006	0,030	0,042	**0,003**	**0,085**	0,380	0,255	0,190	0,110	0,329	0,186	0,198
23.	0,004	0,026	0,024	**0,004**	**0,040**	0,368	0,163	0,126	0,048	0,461	0,164	0,183
24.	0,008	0,024	**0**	0,004	**0,091**	0,289	0,353	0,233	0,136	0,287	0,690	0,581

Табела 5.12. Резултати другог експеримента за CF #8 – IGD метрика												
Проб.	Најбољи				Средњи				Станд. дев.			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
1.	0,213	0,419	0,318	**0,093**	0,389	0,538	0,468	**0,284**	0,095	0,068	0,098	0,138
2.	0,313	0,759	0,524	**0**	0,642	0,970	0,761	**0,372**	0,168	0,131	0,154	0,138
3.	0,116	0,335	0,206	**0,091**	0,243	0,417	0,336	**0,174**	0,091	0,059	0,080	0,051
4.	0,129	0,377	0,208	**0,013**	0,464	0,540	0,457	**0,435**	0,219	0,152	0,149	0,203
5.	0,106	0,272	0,225	**0,041**	0,218	0,394	0,364	**0,164**	0,075	0,055	0,067	0,084
6.	0,243	0,572	0,471	**0,069**	0,462	0,931	0,733	**0,308**	0,226	0,258	0,175	0,183
7.	0,192	0,384	0,288	**0,051**	0,305	0,500	0,424	**0,201**	0,079	0,065	0,072	0,070
8.	0,408	0,734	**0,034**	0,194	0,910	0,996	0,685	**0,550**	0,463	0,120	0,270	0,198
9.	0,303	0,596	0,512	**0,057**	0,555	0,832	0,747	**0,361**	0,116	0,133	0,138	0,143
10.	0,251	0,718	0,557	**0**	0,572	0,960	0,813	**0,370**	0,213	0,128	0,162	0,164
11.	0,105	0,233	0,182	**0,062**	**0,164**	0,324	0,293	0,176	0,048	0,080	0,079	0,057
12.	0,187	0,556	0,304	**0**	0,336	0,650	0,563	**0,308**	0,080	0,068	0,107	0,107
13.	0,221	0,499	0,316	**0,056**	0,323	0,615	0,471	**0,223**	0,095	0,058	0,095	0,067
14.	**0,082**	0,359	0,316	0,122	0,272	0,529	0,439	**0,220**	0,103	0,066	0,083	0,088
15.	0,285	0,956	0,735	**0**	2,616	2,441	**2,276**	2,771	2,168	1,969	1,590	3,020
16.	0,136	0,671	0,371	**0,115**	**0,558**	1,064	0,793	0,576	0,241	0,211	0,168	0,339
17.	0,210	0,618	0,339	**0,087**	0,391	0,841	0,628	**0,280**	0,103	0,095	0,140	0,115
18.	**0**	0,539	0,472	0,150	0,528	0,795	0,683	**0,361**	0,210	0,140	0,193	0,126
19.	0,134	0,607	0,473	**0,077**	**0,318**	0,739	0,629	0,329	0,115	0,070	0,111	0,125
20.	**0,092**	0,434	0,245	0,101	0,341	0,644	0,547	**0,293**	0,133	0,119	0,143	0,121
21.	0,273	0,682	0,567	**0,134**	**0,506**	1,013	0,860	0,550	0,143	0,167	0,329	0,357
22.	0,166	0,541	0,391	**0,073**	0,394	0,768	0,630	**0,273**	0,164	0,132	0,170	0,115
23.	0,171	0,571	0,292	**0,143**	**0,380**	0,799	0,610	0,426	0,158	0,191	0,243	0,269
24.	0,125	0,674	0,515	**0**	0,546	1,017	0,868	**0,518**	0,267	0,315	0,329	0,210

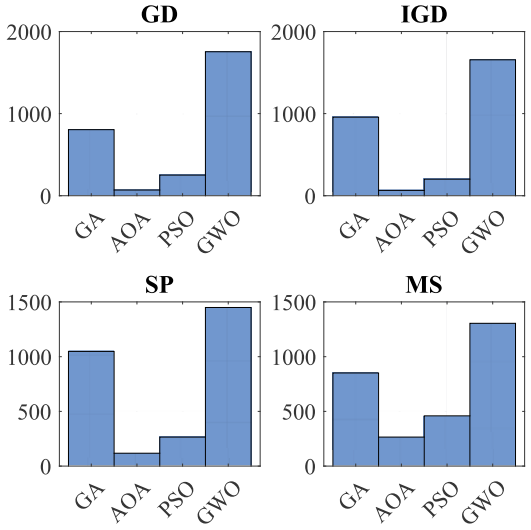
Табела 5.13. Резултати другог експеримента за CF #8 – MS метрика.												
Проблем	Најбољи				Средњи				Станд. дев.			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
1.	**0**	**0**	**0**	**0**	0,607	1,360	0,814	**0,484**	0,394	0,763	0,625	0,330
2.	**0**	0,906	**0**	0,167	1,577	3,216	1,738	**1,103**	1,069	1,584	0,979	0,586
3.	0,431	0,820	0,886	**0,325**	1,564	2,539	1,971	**1,382**	0,704	0,986	0,925	0,729
4.	0,183	0,324	**0**	**0**	2,144	3,397	1,850	**1,392**	1,250	1,784	1,176	0,855
5.	**0,184**	0,806	0,348	0,331	1,057	1,313	1,330	**0,852**	0,381	0,322	0,426	0,344
6.	0,223	**0**	0,266	**0**	1,201	1,869	1,387	**0,713**	0,606	0,808	0,572	0,562
7.	0,450	0,469	0,503	**0,113**	1,612	1,837	1,583	**0,762**	0,687	0,734	0,662	0,475
8.	0,603	0,523	0,557	**0**	3,570	6,423	3,991	**1,970**	1,779	2,731	3,225	1,911
9.	0,240	0,507	**0**	**0**	**1,135**	3,000	2,615	1,682	0,702	1,410	1,679	0,769
10.	**0**	**0**	**0**	**0**	2,546	5,943	3,520	**1,710**	2,387	3,933	2,973	1,662
11.	0	0,464	0,374	**0**	0,681	1,153	1,201	**0,543**	0,421	0,596	0,559	0,379
12.	0,090	0,956	0,346	**0,084**	1,134	2,187	1,675	**0,970**	0,663	0,859	1,077	0,612
13.	**0**	**0**	**0**	**0**	0,438	1,019	0,517	**0,403**	0,386	0,603	0,332	0,343
14.	0,228	0,745	0,390	**0,127**	1,588	1,840	2,084	**0,983**	0,662	0,715	1,498	0,664
15.	4,005	14,009	3,005	**0**	24,937	57,004	29,906	**16,078**	15,912	34,135	18,983	13,911
16.	**0**	1,014	**0**	**0**	**3,204**	9,500	5,215	3,246	2,423	5,384	3,139	2,852
17.	0,162	0,534	0,210	**0**	1,131	1,882	1,939	**0,591**	0,764	0,867	1,227	0,425
18.	**0**	0,375	0,140	**0**	0,709	1,654	1,160	**0,596**	0,579	0,745	0,812	0,433
19.	0,213	0,708	0,241	**0**	0,924	1,807	1,143	**0,610**	0,523	0,913	0,641	0,387
20.	0,144	1,587	0,508	**0**	2,403	4,979	2,623	**1,328**	1,927	2,774	1,485	1,020
21.	0,252	1,051	0,625	**0,132**	3,117	6,532	5,054	**2,030**	2,324	3,801	2,997	1,725
22.	0,171	0,728	0,339	**0**	2,383	6,151	4,743	**1,179**	1,645	3,526	3,009	1,274
23.	0,307	1,046	1,301	**0**	2,713	7,517	5,609	**1,513**	1,953	4,525	2,899	1,324
24.	0,560	**0**	0,895	0,225	3,186	5,905	5,120	**2,389**	2,001	3,355	3,581	3,007

За последњу анализирану комбинацију функција циља (CF #9 – табела 5.14), GWO је остварио сва 24 најбоља резултата и 22/24 средњих најбољих резултата, чиме је потврђена оптималност GWO алгоритма у погледу карактеристика конвергенције. Способност за покривање простора решења потврђена је на основу резултата SP метрике датих у табели 5.15, где је приказано да је GWO остварио 16/24 најбољих решења, док је GA остварио највише средњих најбољих решења. На основу наведених квантитативних резултата, може се закључити да је алгоритам GWO оптималан за анализирани проблем.

Табела 5.14. Резултати другог експеримента за CF #9 – GD метрика.												
Проб.	Најбољи				Средњи				Станд. дев.			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
1.	0,968	2,627	1,352	**0**	3,013	3,775	3,978	**2,618**	1,936	0,863	2,188	2,050
2.	1,549	5,364	2,327	**0,014**	8,221	8,106	7,128	**3,338**	3,722	2,063	3,082	3,720
3.	0,034	0,063	0,106	**0,009**	0,341	0,371	0,324	**0,090**	0,232	0,182	0,175	0,109
4.	0,515	0,603	0,156	**0,077**	1,802	**1,376**	1,400	1,520	1,607	0,463	0,968	1,461
5.	0,038	0,412	0,407	**0,011**	0,731	0,596	0,762	**0,302**	0,444	0,146	0,256	0,206
6.	3,571	50,641	26,470	**0**	58,209	71,324	61,739	**33,893**	31,105	18,591	17,210	21,333
7.	0,029	0,116	0,070	**0,028**	0,418	0,466	0,556	**0,283**	0,338	0,143	0,268	0,252
8.	0,183	2,067	0,480	**0,01**	5,216	4,604	3,973	**2,802**	3,768	1,596	1,647	2,213
9.	78,000	112,001	46,000	**0**	383,110	257,146	279,390	**104,789**	186,399	99,212	154,630	120,448
10.	0,557	1,765	0,980	**0,002**	**1,575**	2,347	2,173	1,734	1,095	0,361	0,867	1,279
11.	0,009	0,057	0,067	**0,006**	0,159	0,171	0,152	**0,053**	0,187	0,107	0,078	0,063
12.	0,174	0,612	0,581	**0**	0,962	1,267	1,464	**0,679**	0,500	0,344	0,529	0,527
13.	1,345	4,707	1,650	**0**	4,753	6,120	5,003	**3,953**	2,206	1,211	1,192	1,928
14.	0,041	0,418	0,206	**0,02**	0,702	0,856	0,761	**0,495**	0,417	0,246	0,326	0,314
15.	2,202	2,876	2,003	**0**	5,350	4,987	4,455	**2,720**	2,091	1,183	1,650	2,291
16.	0,228	1,179	0,704	**0**	1,237	1,784	1,616	**1,209**	0,681	0,370	0,684	0,817
17.	0,516	1,346	1,053	**0**	1,767	1,815	1,855	**1,031**	0,703	0,356	0,425	0,585
18.	0,169	1,004	0,581	**0,052**	1,558	1,584	1,675	**1,287**	1,076	0,547	0,935	0,584
19.	0,084	1,071	0,684	**0**	1,013	1,574	1,593	**0,820**	0,508	0,301	0,542	0,538
20.	0,022	0,529	0,125	**0,01**	0,848	0,935	0,793	**0,505**	0,481	0,247	0,264	0,397
21.	0,500	1,827	1,344	**0**	2,539	2,856	3,051	**1,647**	1,543	0,667	0,863	1,196
22.	0,117	0,812	0,480	**0,001**	1,205	1,410	1,299	**0,832**	0,488	0,271	0,439	0,431
23.	0,045	0,644	0,382	**0,007**	0,704	1,069	0,861	**0,638**	0,536	0,174	0,270	0,414
24.	0,072	0,698	0,466	**0,002**	0,973	1,307	1,180	**0,714**	0,550	0,282	0,332	0,414

Табела 5.15. Резултати другог експеримента за CF #9 – SP метрика.												
Проб.	Најбољи				Средњи				Станд. дев.			
	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO	GA	AOA	PSO	GWO
1.	**0**	0,049	**0**	**0**	**0,077**	N/A	N/A	0,154	0,052	N/A	N/A	0,220
2.	0,039	0,069	**0**	0,005	N/A	0,687	0,500	**0,133**	N/A	0,380	0,393	0,119
3.	0,006	**0**	0,025	0,004	**0,026**	0,165	0,124	0,034	0,016	0,159	0,089	0,024
4.	0,006	0,041	**0**	**0**	**0,057**	N/A	0,119	0,068	0,045	N/A	0,100	0,106
5.	0,017	0,049	0,044	**0,012**	**0,051**	0,142	0,146	0,063	0,034	0,071	0,116	0,066
6.	0,398	1,258	0,856	**0**	1,431	6,456	4,776	**1,271**	0,775	5,809	4,797	1,826
7.	0,007	**0**	0,016	0,007	**0,028**	0,124	0,115	0,031	0,019	0,098	0,093	0,019
8.	**0**	0,100	0,132	**0**	**0,196**	0,558	0,584	0,203	0,201	0,374	0,644	0,421
9.	2,408	2,659	0	**0**	N/A	N/A	N/A	2,813	N/A	N/A	N/A	3,766
10.	**0**	**0**	**0**	**0**	**0,030**	0,119	N/A	0,059	0,023	0,171	N/A	0,062
11.	0,009	0,029	0,021	**0,005**	**0,020**	0,084	0,048	0,023	0,008	0,052	0,019	0,020
12.	0,009	0,014	0,032	**0,004**	**0,030**	0,128	0,109	0,050	0,014	0,085	0,082	0,054
13.	**0**	0,113	**0**	**0**	**0,094**	0,448	0,420	0,232	0,071	0,321	0,472	0,194
14.	**0,007**	0,033	0,019	0,011	0,049	0,099	0,093	**0,035**	0,174	0,063	0,078	0,034
15.	0,017	**0**	**0**	**0**	**0,099**	N/A	0,294	0,101	0,055	N/A	0,301	0,107
16.	0,008	**0**	**0**	**0**	**0,030**	0,124	N/A	N/A	0,016	0,201	N/A	N/A
17.	0,015	0,033	0,029	0,002	**0,050**	0,134	0,107	0,052	0,026	0,074	0,052	0,053
18.	0,014	**0**	**0**	**0**	N/A	0,125	0,112	**0,041**	N/A	0,109	0,128	0,029
19.	0,012	0,048	**0**	**0**	**0,035**	0,138	0,107	0,047	0,027	0,079	0,068	0,061
20.	**0**	0,018	0,024	0,003	**0,022**	0,107	N/A	N/A	0,019	0,104	N/A	N/A
21.	0,014	**0**	**0**	**0**	**0,044**	0,206	0,165	N/A	0,023	0,234	0,088	N/A
22.	0,006	0,041	0,013	**0,004**	**0,021**	0,110	0,104	0,038	0,010	0,047	0,105	0,027
23.	0,007	0,021	**0**	0,003	**0,022**	0,089	0,083	0,060	0,011	0,067	0,060	0,104
24.	**0**	0,015	0,007	0,002	N/A	0,100	0,123	**0,029**	N/A	0,057	0,103	0,045

Како су у оквиру другог експеримента коришћене четири експерименталне студије са значајним бројем података, резултат целокупног експеримента може бити приказан хистограмом на слици 5.14. За 24 проблема, четири комбинације функција циља и 30 понављања за сваки алгоритам, извршено је укупно 2.880 оптимизационих процедура за сваки од алгоритама. Приказ резултата целокупног експеримента може се извршити кроз поређење сваког од 2.880 понављања. Број пута колико је сваки од алгоритама остварио најбољу вредност метрике у оквиру 2.880 понављања приказан је хистограмом на слици 5.14. Као што се може приметити, GWO алгоритам је остварио највише пута оптималну вредност добијену на основу све четири метрике у поређењу са осталим алгоритмима за вишекритеријумску оптимизацију.



Слика 5.14. Хистограм приказаних решења за други експеримент.

6. Закључак

У ери глобалних ланаца снабдевања, конкурентност модерних технолошких система све више зависи од могућности остваривања малих производних серија и великог диверзитета производа, чиме се намеће потреба за високим нивоом флексибилности и могућност реконфигурабилности ентитета у производним системима на основу тренутних потреба дефинисаних тржиштем. Интеграцијом техника вештачке интелигенције са флексибилним технолошким системима остварује се највиша класа, односно интелигентни технолошки системи, који имају могућност да испуне комплексне захтеве глобалног тржишта, а у оквиру којих се може остварити ефикасна физичка и функционална реконфигурабилност технолошких ентитета. Један од начина остваривања физичке реконфигурабилности интелигентног технолошког система је реконфигурација транспортних токова материјала применом мобилних роботских система. Њиховом применом, транспортни токови материјала постају ефикаснији, флексибилнији и оптимално терминирани. Предности које мобилни роботи имају у односу на традиционалне видове транспорта, огледају се високим нивоом флексибилности, ефикасности и брзине транспорта, а што се остварује на основу примене различитих сензора за опажање тренутног стања технолошког окружења, као и напредних алгоритама управљања на бази интеграције сензорских података и техника вештачке интелигенције, који омогућавају прилагођавање тренутном стању технолошког система. Технике вештачке интелигенције омогућавају интерпретацију података добијених од сензора у реалном времену и резоновање о даљим акцијама које је потребно да мобилни робот изврши, како би се технолошки процеси у оквиру производног система несметано извршавали. Имајући то у виду, први део предметне докторске дисертација посвећен је развоју система перцепције мобилног робота.

Систем перцепције мобилног робота RAICO (*Robot with Artificial Intelligence based COgnition*), који се базира на стерео визуелном машинском гледању, развијен је као основни систем за аквизицију и обраду података добијених у интелигентном технолошком окружењу. Хардверски подсистем заснива се на две паралелно постављене индустријске *Basler Dart daA1600-60uc* камере са компатабилним сочивима *Evetar M118B0418W*. Интеграција предложених камера и сочива омогућава роботском систему видно поље од 101/76°, уз минималне дисторзије слике настале због ефекта „рибљег ока“. У сврху обраде података добијених од стерео визуелног система, примењене су технике дубоког машинског учења, тј. конволуционе неуронске мреже (енгл. *Convolutional Neural Networks* – CNN). Задатак за који су примењени CNN модели је семантичка сегментација, која подразумева дефинисање класе сваког пиксела на слици. Анализирани су различити CNN модели са аспеката тачности, брзине процесирања и меморијских захтева. Архитектура на бази *ResNet18* модела је имплементирана са слојевима вештачке неуронске мреже који су подељени у два нивоа од осам блокова, а који су подразумевали, између осталих и слојеве конволуције, линеарне позитивне активације и нормализације тренутне групе улаза. Оптимални модел имплементиран је на *Nvidia Jetson Nano* рачунарску платформу интегрисану у оквиру мобилног робота RAICO и остварена је тачност од 83% и брзина процесирања од 42 FPS-а. Семантичке мапе (излаз из CNN модела) мобилном роботском систему омогућавају препознавање и локализацију различитих технолошких објеката у сцени. Затим, како би се остварила функционална зависност позиције препознатих објеката на сликама генерисаним у различитим положајима у оквиру технолошког окружења, било је потребно имплементирати алгоритам за процес регистрације слика. Градијентни поступак, алгоритам симулираног жарења и еволуциони алгоритам су имплементирани ради извршавања процеса регистрације, док је тачност процеса регистрације дефинисан на основу три различите функције циља, а након тога је анализиран и ефекат интеграције алгоритама оптимизације са одређеним функцијама циља. На основу две експерименталне верификације извршене на основу слика генерисаних у лабораторијском

моделу технолошког окружења мобилним роботским системом RAICO, као најбољи показао се еволуциони алгоритам интегрисан са функцијом циља заједничких информација.

Ради извршавања задатака транспорта и манипулације делова, подсклопова и склопова у оквиру интелигентног технолошког система, мобилни роботски системи морају имати висок ниво тачности остваривања положаја. Традиционални алгоритми локализације базирани на подацима прикупљеним са енкодера, попут модела кретања на основу пређеног пута, због свог инхерентног недостатка који се односи на некоришћење податка о окружењу, не могу обезбедити адекватан ниво тачности. Алгоритми којима се може обезбедити висока тачност оствареног положаја мобилних робота су визуелни системи управљања. Коришћењем визуелних података о тренутном стању окружења, може се минимизирати негативни утицај свих промена генерисаних динамичким и недетерминистичким технолошким окружењем, као и поремећајних фактора. Код система визуелног управљања, брзине које робот остварује генеришу се искључиво на основу визуелних информација. Применом камера које су повезане са роботским системом, генерише се слика циља, тј. слика у жељеном положају. Затим се робот из тренутно одабраног положаја, на основу разлике, тј. грешке између тренутне и жељене слике, може аутоматски довести у жељени положај. Различите методе визуелног управљања дефинишу на другачији начин грешку између тренутне и жељене слике. Основна подела метода визуелног управљања подразумева: 1) Управљање на бази грешака у параметрима слике (енгл. *Image Based Visual Servoing – IBVS*), 2) Управљање на бази издвојених карактеристика са слике неопходних за естимацију положаја изабраног објекта (енгл. *Position-based visual servoing – PBVS*), 3) Хибридне системе управљања (енгл. *Hybrid Visual Servoing – HVS),* и 4) Системе директног визуелног управљања (енгл. *Direct visual servoing – DVS*). IBVS представља 2D визуелно управљање јер се грешка дефинише у параметрима слике, а карактеристични објекти за управљање представљају карактеристичне геометријске примитиве са слика. Насупрот њему, уколико се дефинише 3D положај неког објекта са слике и на основу разлике положаја тог објекта на циљној и тренутној слици врши управљање реч је о PBVS-у. Хибридни системи визуелног управљања подразумевају интеграцију IBVS-а и PBVS-а, или неког другог алгоритма са очекивањем да ће хибридни алгоритам имати боља својства од индивидуалних. На крају, најновија категорија метода визуелног управљања су DVS алгоритми, где се управљање врши на основу свих пиксела на слици или неке репрезентације свих пиксела са слике. Како DVS методе користе више података при управљању, тачност ових метода је знатно виша од претходно наведених. Из тог разлога у оквиру предметне докторске дисертације, развијен је систем стерео визуелног управљања који користи семантичке информације са слика генерисаних у оквиру технолошког окружења ради остваривања тачног положаја нехолономног мобилног роботског система.

На основу семантичких мапа, добијених применом система перцепције (односно CNN модела), генерисаних у тренутном и циљном положају извршава се процес регистрације. Резултат процеса регистрације су матрице трансформације којима се сви пиксели тренутне и циљне слике могу довести у функционалну зависност. Затим се применом Јакобијан матрице слике и матрица трансформације могу генерисати брзине у координатном систему леве и десне камере, а онда и трансформисати у координатни систем мобилног робота како би се њиховим остваривањем достигао жељени положај. Коришћени мобилни роботски систем RAICO креће се помоћу точкова, што проузрокује постојање нехолономних ограничења у његовом кретању. Како би се савладала ова ограничења и остварио жељени положај, предложен је алгоритам управљања који користи визуелне информације и заснива се на четири фазе. Такође, додатни услов који је неопходно испунити је могућност оцене удаљености пиксела на слици. Извршена су два експеримента у 3D симулационом окружењу *CoppeliaSim*, као и три експеримента са реалним мобилним роботским системом RAICO. На основу извршених експеримената, може се закључити да предложени систем управљања остварује високу тачност позиционирања мобилног робота. Такође, посебно се наглашава робустност алгоритма на (i) грешке при семантичкој сегментацији слике, (ii) знатне промене осветљења, као и (iii) заклањање дела

сцене новим објектима. На крају, како је често наглашавани недостатак визуелног управљања потреба за генерисањем нових циљних слика, један од експеримената посвећен је анализи могућности коришћења виртуелних слика циља. Тиме се нове слике циља генеришу у симулацији и директно прослеђују мобилном роботском систему, чиме се флексибилност система управљања знатно повећава. На основу трећег експеримента у реалном окружењу, утврђено је да је могуће, уз смањену тачност оствареног положаја, користити слике циља генерисане у симулацији за предложени алгоритам стерео визуелног управљања.

Фокус истраживања у оквиру предметне докторске дисертације такође је и на развоју алгоритама базираних на техникама вештачке интелигенције који омогућавају оптималну примену мобилних роботских система као транспортних средстава у технолошком окружењу. Првенствено, развијен је систем за интегрисано терминирање технолошких система и једног мобилног робота који извршава опслуживање машина алатки. Развијен је математички модел за оптимизацију планова терминирања, уз представљање 13 функција циља. Предложене функције циља омогућавају оптимизацију рада различитих ентитета (попут машина алатки, робота) или параметара у технолошком систему. Развијена је група функција циља које су везане за параметре кашњења, времена рада или искоришћења роботског система, док је друга група посвећена искоришћењу технолошких ентитета, времену рада машина алатки, кашњењу делова итд. Интеграцијом представљених функција циља генерисане су функције циља за вишекритеријумску оптимизацију. Након процеса оптимизације, решења су представљена применом Парето фронта. Четири метрике вишекритеријумске оптимизације коришћене су за поређење резултата биолошки инспирисаних алгоритама оптимизације. Након анализе резултата два експеримента, који су укључивали око 4.000 понављања процедура оптимизације, као оптималан показао се алгоритам инспирисан интелигенцијом чопора вукова (енгл. *Grey Wolf Optimizer* – GWO). Једно од значајних побољшања у оквиру предложеног GWO алгоритма односи се на три нове стратегије за одабир лидера. Резултат оптимизације планова терминирања представљен је оптималним Гантовим дијаграмом, у оквиру кога се приказује редослед обраде свих операција на машинама алаткама. Такође, добијене су и транспортне операције које треба да изврши мобилни робот како би се технолошки процеси свих делова несметано одвијали. На основу транспортних операција у оквиру оптималног Гантовог дијаграма, дефинисани су сви транспортни задаци које мобилни робот треба да изврши у интелигентом технолошком систему.

Правци будућих истраживања усмерени су ка даљем унапређењу примене мобилних роботских система и техника вештачке интелигенцију у оквиру интелигентних технолошких система, а све са циљем подизања нивоа аутономности, ефикасности и тачности остваривања положаја мобилних робота, као и флексибилности и продуктивности самих интелигентних технолошких система. Будућа истраживања ће обухватати:

- Развој система интелигентног стерео визуелног управљања који се могу имплементирати у оквиру холономних мобилних робота или редундантних индустријских робота;
- Развој нових система дубоког машинског учења који омогућавају ефикасну семантичку сегментацију значајно већег број технолошких ентитета;
- Развој система за сајбер безбедност визуелног управљања мобилних робота на бази алгоритама дубоког машинског учења;
- Развој нове методологије за кодирање и декодирање решења проблема терминирања која омогућава ефикаснију оптимизацију и претрагу простора решења;
- Развој нових метахеуристичких алгоритама за вишекритеријумску оптимизацију планова терминирања.

7. Литература

[1] J. Smit, S. Kreutzer, C. Moeller, M. Carlberg, "Industry 4.0: Study," in *European Parliament*, 2016.

[2] З. Миљковић, М. Петровић, *Интелигентни технолошки системи – са изводима из роботике и вештачке интелигенције*. Универзитет у Београду – Машински факултет, 2021.

[3] M. Mitić, *Empirijsko upravljanje inteligentnog mobilnog robota bazirano na mašinskom učenju*. doktorska disertacija, Univerzitet u Beogradu - Mašinski fakultet, Beograd, 2014.

[4] N. Ghodsian, K. Benfriha, A. Olabi, V. Gopinath, A. Arnou, "Mobile manipulators in Industry 4.0: A review of developments for industrial applications," *Sensors*, vol. 23, no. 19, p. 8026, 2023.

[5] M. B. Alatise, G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020.

[6] D. L. Ripley, T. Politzer, "Vision disturbance after TBI," *NeuroRehabilitation*, vol. 27, no. 3, p. 215, 2010.

[7] Z. Miljković, D. Aleksendrić, *Veštačke neuronske mreže - zbirka rešenih zadataka sa izvodima iz teorije (II izdanje)*. Univerzitet u Beogradu - Mašinski fakultet, 2018.

[8] Z. Miljković, L. Đokić, M. Petrović, "Object detection and tracking in cooperative multi-robot transportation," in *38th International Conference on Production Engineering (ICPE-S 2021)*, 2021, pp. 137–143.

[9] A. Jokić, M. Petrović, Z. Miljković, "Real-time Mobile Robot Perception based on Deep Learning Detection Model," in *New Technologies, Development and Application V (NT 2022). Lecture Notes in Networks and Systems*, Springer, Cham, 2022, pp. 670–677.

[10] A. Jokić, M. Petrović, Z. Miljković, "Mobile robot decision-making system based on deep machine learning," in *9th International Conference on Electrical, Electronics and Computing Engineering (IcETRAN 2022)*, 2022, pp. 653–656.

[11] M. Petrović, A. Jokić, Z. Miljković, Z. Kulesza, "Multi-objective scheduling of a single mobile robot based on the grey wolf optimization algorithm," *Applied Soft Computing*, vol. 131, p. 109784, 2022,

[12] A. Jokić, M. Petrović, Z. Miljković, B. Babić, *Stereo vizuelni sistem percepcije mobilnog robota baziran na dubokom mašinskom učenju*. Tehničko rešenje, Univerzitet u Beogradu - Mašinski fakultet, 2021.

[13] M. I. Jordan, T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[14] A. Toosi, A. G. Bottino, B. Saboury, E. Siegel, A. Rahmim, "A brief history of AI: how to prevent another winter (a critical review)," *PET clinics*, vol. 16, no. 4, pp. 449–469, 2021.

[15] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Ieee, 2009, pp. 248–255.

[17] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR)*, 2015, pp. 1–14.

[18] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[20] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] D. E. Liddle, "The wider impact of Moore's law," *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 28–30, 2006.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834–848, 2017.

[23] H. Noh, S. Hong, B. Han, "Learning Deconvolution Network for Semantic Segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1520–1528.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[25] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3–4, pp. 121–136, 1975.

[26] V. Nair, G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.

[27] K. He, X. Zhang, S. Ren, J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[28] D.-A. Clevert, T. Unterthiner, S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[30] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, J. Feng, "Deepvit: Towards deeper vision transformer," *arXiv preprint arXiv:2103.11886*, 2021.

[31] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[33] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, pmlr, 2015, pp. 448–456.

[34] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, J. Dean, "Carbon Emissions and Large Neural Network Training," *arXiv preprint arXiv:2104.10350*, pp. 1–22, 2021,

[35] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *arXiv preprint arXiv:1602.07360*, pp. 1–13, 2016,

[36] X. Zhang, X. Zhou, M. Lin, J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6848–6856.

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[38] N. Ma, X. Zhang, H.-T. Zheng, J. Sun, "Shufflenet v2: Practical guidelines for efficient CNN architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 116–131.

[39] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1580–1589.

[40] J. Chen, S. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, S.-H. G. Chan, "Run, don't walk: chasing higher FLOPS for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12021–12031.

[41] M. Petrović, A. Jokić, Z. Kulesza, Z. Miljković, "Deep learning of mobile service robots," in *Book Service robots – Advances in Research and Applications*, Nova Science Publishers, New York, 2021, pp. 77–97.

[42] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, C. Cadena, "SegMap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2–3, pp. 339–355, 2020,

[43] M. Stojanovic, U. Padmanabhan, "Object localization within a semantic domain," U.S. Patent-10,452,927, 2019

[44] M. Petrović, Z. Miljković, A. Jokić, "Efficient Machine Learning of Mobile Robotic Systems based on Convolutional Neural Networks," in *Artificial intelligence for Robotics and Autonomous Systems. Studies in Computational Intelligence*, A. Koubaa and A. T. Azar, Eds., Springer, 2023, pp. 1–26.

[45] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, J. Garcia-Rodriguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation," *arXiv preprint arXiv:1704.06857*, 2017,

[46] J. Long, E. Shelhamer, T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 3431–3440.

[47] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," *arXiv preprint arXiv:1606.02147*, 2016,

[48] E. Romera, J. M. Alvarez, L. M. Bergasa, R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.

[49] P. Hu, F. Perazzi, F. C. Heilbron, O. Wang, Z. Lin, K. Saenko, S. Sclaroff, "Real-time semantic segmentation with fast attention," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 263–270, 2020.

[50] L. Falaschetti, L. Manoni, C. Turchetti, "A Low-Rank CNN Architecture for Real-Time Semantic Segmentation in Visual SLAM Applications," *IEEE Open Journal of Circuits and Systems*, 2022.

[51] O. Ronneberger, P. Fischer, T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, 2015, pp. 234–241.

[52] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223.

[53] F. Dustin, "Hello AI world NVidia Jetson," https://github.com/dusty-nv/jetson-inference.

[54] Z. Miljković, A. Jokić, M. Petrović, "Image Registration Algorithm for Deep Learning-Based Stereo Visual Control of Mobile Robots," in *Deep Learning for Unmanned Systems. Studies in Computational Intelligence*, A. Koubaa and A. T. Azar, Eds., Springer, Cham, 2021, pp. 447–479.

[55] A. Dame, E. Marchand, "Mutual information-based visual servoing," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 958–969, 2011,

[56] S. Kirkpatrick, C. D. Gelatt Jr, M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[57] M. Styner, C. Brechbuhler, G. Szekely, G. Gerig, "Parametric estimate of intensity inhomogeneities applied to MRI," *IEEE Transactions on Medical Imaging*, vol. 19, no. 3, pp. 153–165, 2000,

[58] A. Jokić, M. Petrović, Z. Miljković, "Methods for visual servoing of robotic systems: A state of the art survey," *Tehnika*, vol. 73, no. 6, pp. 801–816, 2018,

[59] F. Chaumette, S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006,

[60] Z. Miljković, N. Vuković, M. Mitić, B. Babić, "New hybrid vision-based control approach for automated guided vehicles," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 1–4, pp. 231–249, 2013,

[61] C. Collewet, E. Marchand, "Luminance: A new visual feature for visual servoing," in *Visual Servoing via Advanced Numerical Methods*, Springer London, 2010, pp. 71–90.

[62] S. K. Nayar, S. A. Nene, H. Murase, "Subspace methods for robot vision," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 750–758, 1996,

[63] V. Kallem, J. P. Swensen, G. D. Hager, N. J. Cowan, "Kernel-Based Visual Servoing," in *IEEE/RSJ*

International Conference on Intelligent Robots and Systems, 2007, pp. 1975–1980.

[64] C. Collewet, E. Marchand, F. Chaumette, "Visual servoing set free from image processing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 81–86.

[65] C. Collewet, E. Marchand, "Photometric Visual Servoing," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 828–834, 2011,

[66] D. Bertrand, E. Marchand, "Visual Servoing using the Sum of Conditional Variance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 1689–1694.

[67] G. Silveira, E. Malis, "Direct visual servoing: Vision-based estimation and control using only nonmetric information," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 974–980, 2012,

[68] G. Caron, E. Marchand, E. M. Mouaddib, "Photometric visual servoing for omnidirectional cameras," *Autonomous Robots*, vol. 35, no. 2–3, pp. 177–193, 2013,

[69] M. Bakthavatchalam, F. Chaumette, E. Marchand, "Photometric moments: New promising candidates for visual servoing," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 5241–5246.

[70] C. Teulière, E. Marchand, "A dense and direct approach to visual servoing using depth maps," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1242–1249, 2014.

[71] N. Crombez, G. Caron, E. M. Mouaddib, "Photometric Gaussian mixtures based visual servoing," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5486–5491.

[72] Q. Bateux, E. Marchand, "Histograms-Based Visual Servoing," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 80–87, 2017,

[73] Q. Bateux, E. Marchand, "Direct visual servoing based on multiple intensity histograms," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6019–6024.

[74] Q. Bateux, E. Marchand, "Particle filter-based direct visual servoing," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4180–4186.

[75] N. Crombez, E. M. Mouaddib, G. Caron, F. Chaumette, "Visual Servoing With Photometric Gaussian Mixtures as Dense Features," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 49–63, 2019,

[76] L.-A. Duflot, R. Reisenhofer, B. Tamadazte, N. Andreff, A. Krupa, "Wavelet and Shearlet-based Image Representations for Visual Servoing," *The International Journal of Robotics Research*, vol. 38, no. 4, pp. 422–450, 2019.

[77] E. Marchand, "Direct Visual Servoing in the Frequency Domain," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 620–627, 2020,

[78] M. Adjigble, B. Tamadazte, C. de Farias, R. Stolkin, N. Marturi, "3D Spectral Domain Registration-Based Visual Servoing," *arXiv preprint arXiv:2303.15857*, 2023.

[79] Y. Chen, M. Q.-H. Meng, L. Liu, "Direct Visual Servoing Based on Discrete Orthogonal Moments," *arXiv preprint arXiv:2304.14012*, 2023.

[80] Ž. Jakovljević, P. Petrović, *Prepoznavanje kontaktnih stanja u robotizovanoj montaži*. Monografija, Univerzitet u Beogradu, Mašinski fakultet, 2011.

[81] Z. Jakovljevic, R. Puzovic, M. Pajic, "Recognition of planar segments in point cloud based on wavelet transform," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 342–352, 2015.

[82] S. Bhattacharya, R. Murrieta-Cid, S. Hutchinson, "Optimal paths for landmark-based navigation by differential-drive vehicles with field-of-view constraints," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 47–59, 2007,

[83] D. P. Tsakiris, P. Rives, C. Samson, "Extending visual servoing techniques to nonholonomic mobile robots," in *The Confluence of Vision and Control*, London: Springer, London, 1998, pp. 106–117.

[84] A. S. Popa, M. Popa, I. Silea, A. Varlan, "Indoor navigation of a wheeled mobile robot along Visual Routes," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 3354–3359.

[85] G. Lopez-Nicolas, C. Sagues, J. J. Guerrero, D. Kragic, P. Jensfelt, "Nonholonomic epipolar visual servoing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2378–

2384.

[86] G. López-Nicolás, N. R. Gans, S. Bhattacharya, C. Sagüés, J. J. Guerrero, S. Hutchinson, "Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 4, pp. 1115–1127, 2010,

[87] E. Geometry, D. Tick, A. C. Satici, J. Shen, N. Gans, "Tracking Control of Mobile Robots Localized via Chained Fusion of Discrete and Continuous," *IEEE Transactions on Cybernetics*, vol. 43, no. 4, pp. 1237–1250, 2013.

[88] K. Wang, Y. Liu, L. Li, "Visual servoing trajectory tracking of nonholonomic mobile robots without direct position measurement," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 1026–1035, 2014,

[89] X. Zhang, Y. Fang, B. Li, J. Wang, "Visual Servoing of Nonholonomic Mobile Robots With Uncalibrated Camera-to-Robot Parameters," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 390–400, 2017,

[90] J. Chen, B. Jia, K. Zhang, "Trifocal Tensor-Based Adaptive Visual Trajectory Tracking Control of Mobile Robots," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3784–3798, 2017,

[91] F. Ke, Z. Li, H. Xiao, X. Zhang, "Visual Servoing of Constrained Mobile Robots Based on Model Predictive Control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1428–1438, 2016.

[92] H. Xiao, Z. Li, C. Yang, L. Zhang, P. Yuan, L. Ding, T. Wang, "Robust Stabilization of a Wheeled Mobile Robot Using Model Predictive Control Based on Neurodynamics Optimization," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 505–516, 2017,

[93] F. Ke, Z. Li, C. Yang, "Robust Tube-based Predictive Control for Visual Servoing of Constrained Differential-Drive Mobile Robots," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3437–3446, 2018,

[94] B. Li, X. Zhang, Y. Fang, W. Shi, "Visual Servoing of Wheeled Mobile Robots Without Desired Images," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 2835–2844, 2018,

[95] B. Li, X. Zhang, Y. Fang, W. Shi, "Visual Servo Regulation of Wheeled Mobile Robots with Simultaneous Depth Identification," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 460–469, 2018,

[96] R. S. Sharma, S. Shukla, L. Behera, V. K. Subramanian, "Position-based visual servoing of a mobile robot with an automatic extrinsic calibration scheme," *Robotica*, vol. 38, no. 5, pp. 831–844, 2020.

[97] R. Perez-Alcocer, L. G. Montoya-Villegas, A. E. Lopez-Martinez, J. Moreno-Valenzuela, "Saturated visual-servoing control strategy for nonholonomic mobile robots with experimental evaluations," *IEEE Access*, vol. 9, pp. 130680–130689, 2021.

[98] K. Xing, Y. Lin, D. He, W. Dong, "Visual servoing control of omnidirectional mobile robots based on disturbance observer," *Measurement and Control*, vol. 56, no. 5–6, pp. 1016–1025, 2023.

[99] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, P. Corke, "The limits and potentials of deep learning for robotics," *International Journal of Robotics Research*, vol. 37, no. 4–5, pp. 405–420, 2018,

[100] Z. Miljković, M. Mitić, M. Lazarević, B. Babić, "Neural network Reinforcement Learning for visual control of robot manipulators," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1721–1736, 2013,

[101] T. Lampe, M. Riedmiller, "Acquiring visual servoing reaching and grasping skills using neural reinforcement learning," *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2013,

[102] M. Mitić, Z. Miljković, "Neural network learning from demonstration and epipolar geometry for visual control of a nonholonomic mobile robot," *Soft Computing*, vol. 18, no. 5, pp. 1011–1025, 2014,

[103] F. Zhang, J. Leitner, M. Milford, B. Upcroft, P. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," *arXiv:1511.03791*, 2015.

[104] H. Pandya, K. M. Krishna, "Graph Based Visual Servoing for Object Category," *Proceedings of the Advances in Robotics. ACM*, p. 48, 2017,

[105] A. Saxena, H. Pandya, G. Kumar, A. Gaud, K. M. Krishna, "Exploring Convolutional Networks for

End-to-End Visual Servoing," *arXiv:1706.03220*, 2017,

[106] A. X. Lee, S. Levine, P. Abbeel, "Learning Visual Servoing with Deep Features and Fitted Q-Iteration," *arXiv preprint arXiv:1703.11000*, 2017,

[107] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[108] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, "Training Deep Neural Networks for Visual Servoing," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1–8.

[109] M. Petrović, A. Mystkowski, A. Jokić, L. Đokić, Z. Miljković, "Deep Learning-based Algorithm for Mobile Robot Control in Textureless Environment," in *2020 International Conference Mechatronic Systems and Materials (MSM)*, Bialystok, Poland: IEEE, 2020, pp. 1–4.

[110] S. Felton, P. Brault, E. Fromont, E. Marchand, "Visual Servoing in Autoencoder Latent Space," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3234–3241, 2022,

[111] S. Felton, É. Fromont, E. Marchand, "Deep metric learning for visual servoing: when pose and image meet in latent space," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 741–747.

[112] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000,

[113] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Second Edi. Cambridge University Press, 2000.

[114] F. Chaumette, S. Hutchinson, "Visual servo control. II. Advanced approaches [Tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007,

[115] S. Hutchinson, G. D. Hager, P. Corke, "A Tutorial on Visual Servo Control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996,

[116] E. Malis, F. Chaumette, S. Boudet, "2D 1 / 2 Visual Servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 234–246, 1999.

[117] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB®*. Springer, 2017.

[118] J. C. A. Barata, M. S. Hussein, "The Moore–Penrose pseudoinverse: A tutorial review of the theory," *Brazilian Journal of Physics*, vol. 42, pp. 146–165, 2012.

[119] M. Mitić, Z. Miljković, "Bio-inspired approach to learning robot motion trajectories and visual control commands," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2624–2637, 2015,

[120] G. Bradski, "The openCV library," *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[121] A. Jokic, A. Khazraei, M. Petrovic, Z. Jakovljevic, M. Pajic, "Cyber-Attacks on Wheeled Mobile Robotic Systems with Visual Servoing Control," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 6342–6348.

[122] E. Malis, M. Vargas, "Deeper understanding of the homography decomposition for vision-based control," *[Research Report] RR-6303*, INRIA, p. 90, 2007.

[123] É. Marchand, F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53–70, 2005,

[124] A. H. Abdul Hafez, S. Achar, C. V. Jawahar, "Visual servoing based on Gaussian mixture models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 3225–3230.

[125] E. Rohmer, S. P. N. Singh, M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 1321–1326.

[126] A. Jokić, M. Petrović, Z. Miljković, "Semantic segmentation based stereo visual servoing of nonholonomic mobile robot in intelligent manufacturing environment," *Expert Systems with Applications*, vol. 190, p. 116203, 2022,

[127] L. Đokić, A. Jokić, M. Petrović, Z. Miljković, "Stereo vision-based algorithm for control of nonholonomic mobile robot," in *The Third International Students' Scientific Conference*

"Multidisciplinary approach to contemporary research - Cultural and Industrial Heritage," 2019, pp. 69–82.

[128] M. Petrović, "Veštačka inteligencija u projektovanju inteligentnih tehnoloških sistema," doktorska disertacija, Univerzitet u Beogradu - Mašinski fakultet, 2016.

[129] S. Mirjalili, J. S. Dong, *Multi-Objective Optimization using Artificial Intelligence Techniques*. Springer, 2020.

[130] M. Petrović, M. Mitić, N. Vuković, Z. Miljković, "Chaotic particle swarm optimization algorithm for flexible process planning," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 9–12, pp. 2535–2555, 2016,

[131] S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014,

[132] M. Petrović, Z. Miljković, A. Jokić, "A novel methodology for optimal single mobile robot scheduling using whale optimization algorithm," *Applied Soft Computing*, vol. 81, p. 105520, 2019,

[133] C. J. Chen, C. S. Tseng, "The path and location planning of workpieces by genetic algorithms," *Journal of Intelligent Manufacturing*, vol. 7, no. 1, pp. 69–76, 1996,

[134] O. Maimon, D. Braha, V. Seth, "Neural network approach for a robot task sequencing problem," *Artificial Intelligence in Engineering*, vol. 14, no. 2, pp. 175–189, 2000,

[135] A. Landrieu, Y. Mati, Z. Binder, "A tabu search heuristic for the single vehicle pickup and delivery problem with time windows," *Journal of Intelligent Manufacturing*, vol. 12, no. 5–6, pp. 497–508, 2001,

[136] J. Hurink, S. Knust, "A tabu search algorithm for scheduling a single robot in a job-shop environment," *Discrete Applied Mathematics*, vol. 119, no. 1–2, pp. 181–203, 2002,

[137] Q. V. Dang, I. E. Nielsen, K. Steger-Jensen, "Mathematical formulation for mobile robot scheduling problem in a manufacturing cell," *IFIP Advances in Information and Communication Technology*, vol. 384 AICT, pp. 37–44, 2012,

[138] Q. V. Dang, I. E. Nielsen, G. Bocewicz, "A genetic algorithm-based heuristic for part-feeding mobile robot scheduling problem," in *Trends in Practical Applications of Agents and Multiagent Systems*, Springer Berlin Heidelberg, 2012, pp. 85–92.

[139] Q. V. Dang, I. E. Nielsen, K. Steger-Jensen, O. Madsen, "Scheduling a single mobile robot for part-feeding tasks of production lines," *Journal of Intelligent Manufacturing*, vol. 25, no. 6, pp. 1271–1287, 2014,

[140] Q. V. Dang, L. Nguyen, "A Heuristic Approach to Schedule Mobile Robots in Manufacturing Environments," in *Procedia CIRP*, Elsevier B.V., 2016, pp. 390–395.

[141] M. Mousavi, H. J. Yap, S. N. Musa, S. Z. M. Dawal, "A fuzzy hybrid GA-PSO algorithm for multi-objective AGV scheduling in FMS," *International Journal of Simulation Modelling*, vol. 16, no. 1, pp. 58–71, 2017,

[142] M. Petrović, Z. Miljković, "Grey Wolf Optimization Algorithm for Single Mobile Robot Scheduling," in *Proceedings of 4th International Conference on Electrical, Electronics and Computing Engineering, IcETRAN*, 2017, p. ROI1.2.1-6.

[143] Y. Sun, S.-H. Chung, X. Wen, H.-L. Ma, "Novel robotic job-shop scheduling models with deadlock and robot movement considerations," *Transportation Research Part E: Logistics and Transportation Review*, vol. 149, p. 102273, 2021.

[144] Z. Liu, Q. Luo, L. Wang, H. Tang, Y. Li, "The Low-Carbon Scheduling Optimization of Integrated Multispeed Flexible Manufacturing and Multi-AGV Transportation," *Processes*, vol. 10, no. 10, p. 1944, 2022.

[145] Y. Yao, Q. Wang, C. Wang, X. Li, L. Gao, K. Xia, "Knowledge-based multi-objective evolutionary algorithm for energy-efficient flexible job shop scheduling with mobile robot transportation," *Advanced Engineering Informatics*, vol. 62, p. 102647, 2024.

[146] T. Zhang, G. Zhu, "Dual Resource Scheduling Problem of Machines and AGVs Based on Hybrid Discrete Salp Swarm Algorithm," *IEEE Access*, 2024.

[147] M. Petrović, A. Jokić, B. Babić, *Višekriterijumsko odlučivanje inteligentnog mobilnog robota na bazi*

metoda metaheurističke optimizacije i dubokog mašinskog učenja. Tehničko rešenje, Univerzitet u Beogradu - Mašinski fakultet, 2022.

[148] H. Ishibuchi, H. Masuda, Y. Tanigaki, Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Evolutionary Multi-Criterion Optimization. EMO 2015. Lecture Notes in Computer Science*, vol. 9019, A. Gaspar-Cunha, C. Henggeler Antunes, and C. Coello, Eds., Springer, Cham., 2015, pp. 110–125.

[149] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001.

[150] M. Petrović, N. Vuković, M. Mitić, Z. Miljković, "Integration of process planning and scheduling using chaotic particle swarm optimization algorithm," *Expert Systems with Applications*, vol. 64, pp. 569–588, 2016,

[151] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, p. 113609, 2021.

[152] Z. Miljković, M. Petrović, "Application of modified multi-objective particle swarm optimisation algorithm for flexible process planning problem," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 2–3, pp. 271–291, 2017,

Биографија

Александар (Владан) Јокић, маст. инж. маш, рођен је 16. октобра 1993. године у Београду, Република Србија. Основну школу „Ђуро Стругар“ и средњу школу „Политехника – школа за нове технологије“ завршио је у Београду.

На Универзитет у Београду – Машински факултет уписао се школске 2012/2013. године. Основне академске студије завршио је 2015. године са просечном оценом 8,38 (осам и 38/100). Школске 2015/2016. године уписао је Мастер академске студије на Катедри за производно машинство, а исте завршио 2017. године са просечном оценом 9,30 (девет и 30/100), одбранивши мастер рад (MSc) на тему „Визуелно управљање интелигентног мобилног робота у функцији терминирања унутрашњег транспорта“ из предмета Интелигентни технолошки системи, под менторством проф. др Зорана Миљковића, и то са оценом 10 (десет). Након дипломирања, школске 2017/2018. године, уписао је Докторске академске студије на Универзитету у Београду – Машинском факултету. Кандидат је 1.5.2018. године остварио право на стипендију за Докторске академске студије Министарства просвете, науке и технолошког развоја. Од 6.5.2019. године, запослен је на Машинском факултету у Београду, као истраживач-приправник, од 12.5.2021. године као истраживач-сарадник, а од 01.10.2021. године као асистент, на Катедри за производно машинство.

Од 1.5.2018. године до 31.12.2020. године био је ангажован на пројекту Технолошког развоја (ев. бр. ТР35004) који је финансирало Министарство просвете, науке и технолошког развоја. Од 1.1.2021. је укључен на пројекат „*Интегрисана истраживања у области макро, микро и нано машинског инжењерства*“ који је финансиран од стране Министарства просвете, науке и технолошког развоја односно Министарство науке, технолошког развоја и иновација. У периоду од 2020. до 2022. године био је укључен у истраживачки тим пројекта *„Deep Machine Learning and Swarm Intelligence-based Optimization Algorithms for Control and Scheduling of Cyber-Physical Systems in Industry 4.0“* (акроним – MISSION4.0, ев. бр. 6523109), којим је руководио проф. др Зоран Миљковић и који је финансиран од стране Фонда за науку Републике Србије. Од 1.1.2025. године ангажован је на пројекту „*Cybersecurity of Motion Control Systems in Industry 4.0*“ (акроним – MCSecurity, ев. бр. 17801), којим руководи проф. др Живана Јаковљевић и који финансира Фонд за науку Републике Србије. У периоду од 1.4.2022. до 30.6.2022. године остварио је мобилност на Дјук Универзитету из Дурхама, Сједињене Америчке Државе, кроз *Erasmus+ Higher education student and staff mobility between Programme and Partner Countries* (KA107) механизме. Током мобилности био је укључен у активности Лабораторије за кибернетско-физичке системе.

Активно је укључен у наставни процес Катедре за производно машинство, учествујући у реализацији свих видова вежби (аудиторне вежбе, лабораторијске вежбе, преглед самосталних задатака, преглед пројеката) на осам наставних предмета на ОАС и МАС студијама.

На основу резултата научно-истраживачког рада, као аутор или коаутор, објавио је 31 научни рад, реализовао је три техничка решења и објавио један скуп података. Од научних радова издвајају се два рада објављена у међународним часописима изузетних вредности, SCI-Web of Science® (M21a), један рад у врхунском међународном часопису категорије M21, као и један рад објављен на престижној конференцији у области роботике IROS2023.

У оквиру своје научно-истраживачке каријере, кандидат је добитник многобројних награда и захвалница, од којих се издваја Награда Задужбине Ђоке Влајковића, за најбољи научни рад младих научних радника Универзитета у Београду 2024. године.

образац изјаве о ауторству

Изјава о ауторству

Име и презиме аутора Александар Јокић

Број индекса Д31/2017

Изјављујем

да је докторска дисертација под насловом

Визуелно управљање мобилног робота базирано на биолошки инспирисаним техникама вештачке интелигенције

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

У Београду, 01.05.2025.

A. Jokić

образац изјаве о истоветности штампане и електронске верзије докторског рада

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Александар Јокић

Број индекса Д31/2017

Студијски програм Докторске академске студије - Машинско инжењерство

Наслов рада Визуелно управљање мобилног робота базирано на биолошки инспирисаним техникама вештачке интелигенције

Ментор Ванр. проф. др Милица Петровић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањивања у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, 01.05.2025.

A. Jokić

образац изјаве о коришћењу

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Визуелно управљање мобилног робота базирано на биолошки инспирисаним техникама вештачке интелигенције

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

2. Ауторство – некомерцијално (CC BY-NC)

3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, 01.05.2025.

A. Lozić

1. **Ауторство**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.