

**АУТОНОМНА НАВИГАЦИЈА РОБОТА МЕЂУ РЕДОВИМА ПОЉОПРИВРЕДНИХ
ЗАСАДА****AUTONOMOUS ROBOT NAVIGATION BETWEEN ROWS OF CROPS**Милош Николић, Милутин Николић, *Факултет техничких наука, Нови Сад***Област – МЕХАТРОНИКА**

Кратак садржај – Рад описује аутономну навигацију UGV робота међу редовима пољопривредних засада. Задатак рада био је истражити најзаступљеније приступе и роботе који се примењују за наведени задатак. Тако су за праћење редова засада упоређене две методе, и то: традиционална метода обраде слике и метода машинског учења. Као посебан задатак издвојило се препознавање краја реда и прелазак робота из једног реда у други. За препознавање краја реда коришћен је RTK-GPS сензор, док је за генерисање и праћење путање која води робота у следећи ред засада, коришћен NAV 2 Waypoint Follower заснован на PurePursuit алгоритму. Цео систем је интегрисан у ROS 2 окружењу. Приступу су тестирани у стварном и симулационом окружењу. За симулационо окружење коришћена је NVIDIA Isaac Sim симулација из које је и прикупљан семантички скуп података за тренирање YOLOv8 модела за праћење редова засада. На крају је урађена евалуација метода и њихово упоређивање.

Кључне речи: робот, UGV, аутономна навигација, машинско учење, обрада слике, ROS 2, симулација

Abstract – The paper describes autonomous UGV robot navigation among rows of crops. The objective of the study was to explore the most relevant approaches and robots used for this task. Two methods were compared for row tracking: traditional image processing and machine learning methods. A specific task included recognizing the end of row crops and navigating the robot from one row to another. An RTK-GPS sensor was used for detecting the end of the row crops, while a NAV 2 Waypoint Follower based on the PurePursuit algorithm was embedded for generating and tracking paths which are guiding the robot to the next row crop. The entire system was integrated into the ROS 2 environment. The approaches were tested in both real and simulation environments. NVIDIA Isaac Sim simulation was used for the simulation environment, from which semantic datasets were collected for training the YOLOv8 row crop tracking model. Finally, an evaluation and comparison of the methods were conducted.

Keywords: robot, UGV, autonomous navigation, machine learning, image processing, ROS 2, simulation

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Милутин Николић, ванр. проф.

1. УВОД

Недавни изазови као што су климатске промене, доступност радника у пољопривреди и растући трошкови производних ресурса (на пример ђубрива, пестицида, хербицида) додатно су повећали притисак на пољопривредну индустрију [1].

Типична и погоднија окружења за рад и примену робота јесу индустријска окружења која имају веома контролисане и стандардизоване услове. Иако постоје пластеници у којима се могу достићи услови доста слични условима рада који су заступљени у индустријским окружењима, аграрне средине представљају отежавајући фактор, где постоје веома променљиви услови који утичу на рад робота и његово понашање. Као неке од најбитнијих карактеристика које дефинишу рад робота у овим условима јесу величина и конфигурација терена, врста засада, рад са прехранбеним производима, рад у окружењу са људима, разноврсност задатака, временски услови и слично.

Тако се издвајају два главна типа робота за које се пројектују модули за посебне намене и који се користе у аграрним срединама за пољопривредне послове, а то су UAV (енгл. *Unmanned Aerial Vehicle*) и UGV роботи (енгл. *Unmanned Guided Vehicle*). У овом раду је, сходно задатку, стављен акценат на UGV роботе и то на посебно пројектоване мобилне платформе.

2. РОБОТ И ПОЉЕ ЗАСАДА

Засади соје имају кључну улогу у светској прехранбеној индустрији. Они се гаје у редовима што омогућава лакшу обраду, боље управљање коровом и оптимално искоришћење земљишта и ресурса.

Робот који је изабран да се креће међу редовима засада соје јесте Спесбот робот који на основу механичке конфигурације припада мобилним роботима са диференцијалним погоном и као такав има својствене карактеристике.

2.2. Симулација

Пошто је за кретање робота кроз поље засада соје неопходно да се робот креће уз помоћ модела који је истрениран на означеним сликама, најпогодније решење представља NVIDIA Isaac Sim симулација која је намењена посебно за роботске системе засноване на AI технологији. Унутар ње коришћен је Replicator. Он представља алат који служи за генерисање семантичких података, направљен да подржи развој и обуку модела машинског учења, посебно у областима као што су машинска визија и роботика. Он

аутоматски обележава генерисане слике, што значи да свака слика долази са релевантним подацима као што су позиције објеката, ознаке класа и друге информације које су корисне за обуку модела машинског учења. У овом случају, он даје податке везане за класу земље и соје.

Пошто је као главно окружење које спаја све роботске периферије изабран *ROS 2*, његово повезивање са симулацијом је било лако успоставити коришћењем алата графикана акција (*Action Graph*). Он омогућава визуелно програмирање користећи интегрисане чворове који су повезани заједно у графички интерфејс. Уз помоћ ње је покретан процес за прикупљање семантичких података, као и скрипта за аутоматско генерисање поља засада. Симулационо окружење са Спесбот роботом и пољем засада соје у којем је вршено тестирање алгоритама за кретање приказано је на слици 2.1.



Слика 2.1. *NVIDIA Isaac Sim* симулационо окружење са Спесбот роботом и пољем засада соје.

3. ROS 2

Главно окружење за развој и повезивање свих периферија у једну смислену целину јесте *ROS 2*, који је постао стандард у роботским системима и као такав омогућава широки спектар алата и подокружења који олакшавају рад са роботима. Концепт *ROS 2* окружења је такав да сваки модул има засебан пакет који може бити написан у различитим програмским језицима као што су *Python* и *C++*. *ROS 2* омогућава комуникацију између различитих компоненти роботског система путем система за размену порука (теме, сервиси и акције) и тако олакшава синхронизацију између сензора, актуатора и различитих алгоритама. Уз помоћ система за размену порука могуће је и лако интегрисати роботски систем са жељеном симулацијом.

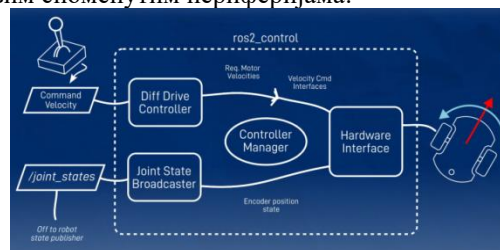
Коришћена је верзија *ROS 2 Humble*, која је компатибилна са *Ubuntu 22.04* оперативним системом у оквиру које су задане команде из конзоле (терминала) за покретање разних процеса.

3.1. ROS 2 управљање

За једноставно управљање различитим хардверским периферијама као што су мотори, сензори и актуатори коришћено је окружење *ROS 2* управљање.

Као што је раније споменуто Спесбот робот је робот са диференцијалним погоном, и зато се користи контролер за диференцијалне погоне (*Diff Drive Controller*) који садржи алгоритме за управљање. Са обзиром да посматрамо кретање мобилне платформе у равни, релевантне су линеарне брзине у x и у y правцу,

као и угаона брзина око z осе. Узимајућу у обзир и кинематска ограничења диференцијалног погона (немогућност бочног кретања робота, тачније по y правцу роботског координатног система), добијамо два слободна параметра за управљање робота и то брзином робота у x правцу (v_x) и угаоном брзином око z осе (ω_z). Они представљају улаз у контролер за диференцијални погон који затим, за задате параметре, рачуна брзине левог и десног мотора који се даље шаљу хардвер интерфејсу. Када је у питању управљање робота у симулацији, са обзиром да не постоји мотор који покреће робота, већ се команде за управљање задају преко *ROS 2* поруке на тему *cmd_vel*, управљање је базирано само на управљању преко *ROS 2* тема. Из тог разлога се користи хардвер интерфејс заснован на темама [2]. За објављивање стања зглобова, користи се *Joint State Broadcaster* контролер. Како би се успоставила контрола и синхронизација над хардвер интерфејсом и контролерима постоји периферија за управљање *Controller Manager*. На слици 3.1. приказана је архитектура *ROS 2* управљања примењена на роботу са свим споменутим периферијама.



Слика 3.1. Архитектура *ROS 2* управљања примењена на роботу [3].

Описан је процес којим се улазне команде линеарне и угаоне брзине (црвена и плава стрелица респективно) претварају у одговарајуће брзине левог и десног точка. Тип *ROS 2* поруке који садржи ове податке и који се објављује на тему за управљање јесте *geometry_msgs/msg/Twist*. Међутим, битно је напоменути да оне као такве нису најпогодније за управљање јер је неопходно задавати низ команди које би довеле робота у жељену позицију. Зато контролер диференцијалног погона објављује трансформацију између координатних система *odometry* и *base_link* која је неопходна како би се користило *NAV 2* окружење које омогућава интуитивније начине за задавање циљева робота.

3.2. NAV 2

NAV 2 је софтверски пакет за навигацију робота у *ROS 2* окружењу, који омогућава управљање мобилним роботима у различитим условима на начин који је сигуран и ефикасан. Он пружа све неопходне компоненте за планирање путање, локализацију, избегавање препрека и управљање кретањем. Главни модул у склопу *NAV 2* окружења који је коришћен јесте *Waypoint Follower*. Он прима низ тачака кроз које робот треба да прође у одређеном редоследу и тако генерише жељену трајекторију. Праћење трајекторије се врши применом *PurePursuit* алгорита на основу кога се генерише жељена линеарна и угаона брзина за управљање роботом. Као најважнији

параметар *PurePursuit* контролера јесте *LookAheadDistance*, који говори роботу на којој раздаљини од тренутне позиције робота, он треба да узима у обзир одговарајућу тачку са задате трајекторије.

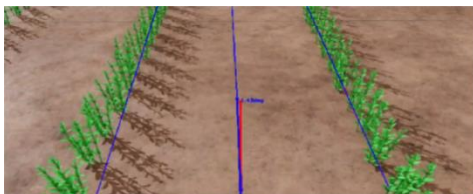
4. ПРАЋЕЊЕ РЕДОВА ЗАСАДА

4.1. Хафова трансформација

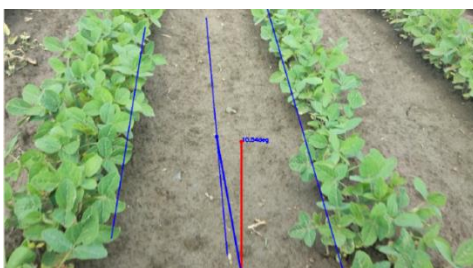
Поступци за обраду слике се могу разложити у неколико корака. Редослед корака и број корака се може разликовати у зависности од окружења у ком се налазе, али је генерализовано следећим редоследом:

1. Прикупљање и учитавање слика
2. Предпроцесирање слике
3. Филтрирање и уклањање шума
4. Бинаризација слике
5. Препознавање ивица
6. Препознавање линија

Комбинацијом ових корака се добијају линије од интереса. Тако је резултат Хафове трансформације у симулационом окружењу приказан на слици 4.1. док је резултат Хафове трансформације у стварном окружењу приказан на слици 4.2.



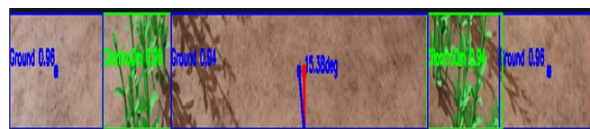
Слика 4.1. Резултат Хафове трансформације за кретање робота међу редовима пољопривредних засада у симулационом окружењу.



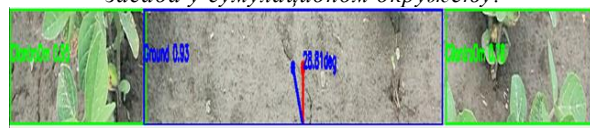
Слика 4.2. Резултат Хафове трансформације за кретање робота међу редовима пољопривредних засада у стварном окружењу.

4.2. YOLOv8

За коришћење методе машинског учења неопходан скуп података над којим се она тренира. Скуп податка може бити из симулационог или стварног окружења или њихове комбинације. Конкретно, модел који је коришћен добијен је из симулације над скупом података од 250 слика, за класе земља и першун. Крива прецизност – одзив је дала вредност од 0,980 просечне прецизности при прагу од 0,5 за све класе, што је одлична вредност за показатеља перформанси модела у препознавању објеката. *F1* крива је показала да је најбоља вредност прага за све класе 0,701 јер тада је највећа вредност *F1* мере. Слика која приказује резултат истренираног *YOLOv8* модела у симулационом окружењу приказана је на слици 4.3. док је резултат истренираног *YOLOv8* модела у стварном окружењу приказана је на слици 4.4.



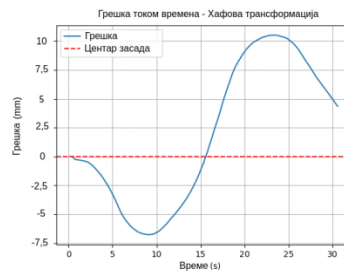
Слика 4.3. Резултат истренираног *YOLOv8* модела за кретање робота међу редовима пољопривредних засада у симулационом окружењу.



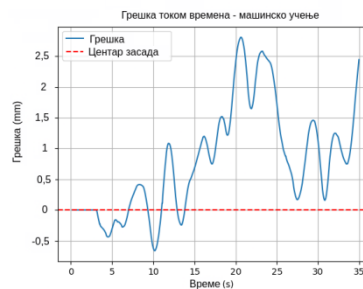
Слика 4.4. Резултат истренираног *YOLOv8* модела за кретање робота међу редовима пољопривредних засада у стварном окружењу.

5. УПОРЕЂИВАЊЕ МЕТОДА ЗА ПРАЋЕЊЕ РЕДОВА ЗАСАДА

За евалуирање коришћених метода, примењено је одступање, односно грешка праћења предвиђене навигационе линије у односу на своју референтну вредност током времена. Битно је напоменути да је референтна вредност добијена на основу *RTK-GPS* сензора са обзиром да су координате референтне путање кретања робота (центра засада) познате.



Слика 5.1. Одступање стварне путање кретања робота од референтне током времена за традиционалну методу



Слика 5.2. Одступање стварне путање кретања робота од референтне током времена за методу машинског учења.

На слици 5.1. вредност грешке, у најнеповољнијем случају, иде и до 10 *mm* у односу на своју референтну вредност, док на слици 5.2. она достиже вредност од 2,5 *mm*. Разлика вредности грешке двеју метода, у најнеповољнијем случају, износи 7,5 *mm*. Осим тога, код методе машинског учења, вредност грешке чешће конвергира ка референтној вредности. Из наведеног следи да је метода машинског учења прецизнија и поузданија у односу на традиционалну методу.

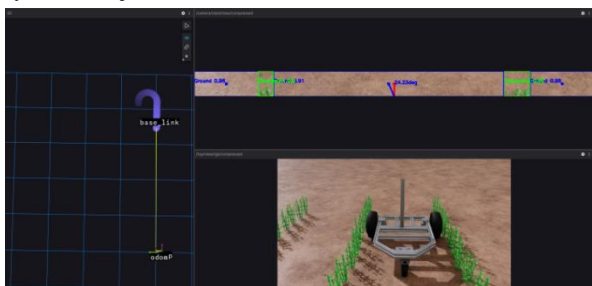
6. ПРЕПОЗНАВАЊЕ КРАЈА РЕДА

Постоји више начина на које се може препознати крај реда, као што је примена *GPS* сензора, *ArUco* маркера, података са камере, *LiDAR* сензора и слично.

Међутим, коришћене је GPS сензор и то *RTK-GPS* јер се уз помоћ њега могу евалуирати и коришћене методе праћења редова засада. *RTK-GPS* је напреднија верзија стандардног *GPS* система који користи технику кинематике у реалном времену да би значајно повећао прецизност позиционирања.

6.1. *RTK-GPS*

Са обзиром да *RTK-GPS* сензор шаље координате у облику географске ширине, дужине и надморске висине. Трансформацију из симулације *map* -> *base link* је неопходно прво конвертовати у географске координате и објављивати их на *ROS 2* тему у облику *NavSatFix* поруке. На тај начин је имплементиран *RTK-GPS* сензор у самој симулацији. Затим је неопходно географске координате, са поменуте *ROS 2* теме, конвертовати у координате релативне *map* координатном систему, односно референтној базној станици у реалном окружењу. За конвертовање између географских координата и координата у одређеном координатном систему коришћена је *pymap3d* библиотека [4] која је писана у *Python* програмском језику. Коришћењем овог приступа, роботски систем из симулационог окружења се може лако интегрисати у стварно окружење. Такође, овиме је решен проблем препознавања краја реда где робот када пређе референтне координате које представљају крај засада, активира се *NAV 2* алат за генерисање путање. Тренутак када робот препозна крај реда и генерише путању за кретање робота визуализована је у *Foxglove* програму за визуализацију роботских података, и илустрована је на слици 6.1.



Слика 6.1 Изгенерисана *NAV 2* путања која је визуализована у *Foxglove* програму.

7. ЗАКЉУЧАК

У циљу решавања актуелних пољопривредних изазова и проблема у индустрији, коришћен је *UGV* Спесбот робот који спада у групу посебно пројектованих пољопривредних платформи. Истражени су и упоређени различити приступи и технологије за аутономну навигацију робота међу редовима пољопривредних засада. Примењене методе укључују Хафову трансформацију и *YOLOv8* модел за препознавање објеката. Обе методе су тестиране у симулационом и стварном окружењу, а резултати су показали да свака метода има своје предности и ограничења у зависности од специфичних услова рада. Посебан акценат је стављен на *RTK-GPS* сензоре и њихов значај за евалуацију прецизности метода праћења редова засада и улогу при преласку између редова.

Примењена метода машинског учења, конкретно *YOLOv8*, показала се као поузданија и прецизнија у препознавању редова засада у односу на традиционалну методу. Конкретно модел машинског учења је био прецизнији за приближно *7,5 mm* у односу на традиционалну методу гледајући најнеповољнији случај. Осим тога брже је конвергирао ка референтној вредности. Ипак, приступ *YOLOv8* модела који користи слику димензија (*1280*80*), иако смањује количину података коју је неопходно истренирати, у реалним окружењима може бити ограничење када се на путу робота налази нежељени засад. Са друге стране је битно нагласити да је Хафова трансформација, осим што се показала као мање прецизна и поуздана, веома је осетљива на мале промене у окружењу што је чини непогодном за примену у односу на методу машинског учења.

Такође, скуп података за тренинг који је неопходан, може се лако добити из *IsaacSim* симулације. Рад је показао да није неопходно прикупљати хибридни скуп података, већ ако се симулација добро подеси за различите услове, може се на лак и брз начин сакупити велики скуп података на основу кога се врши даље тренирање. Тиме се олакшава цео процес, где није неопходно излазити у поље засада и прикупљати податке за тренирање.

На крају, овај рад доприноси развоју паметне пољопривреде, где аутономни роботизовани системи могу значајно побољшати ефикасност, смањити трошкове, заштитити животну средину и унапредити продуктивност и одрживост пољопривредне производње.

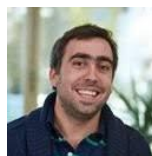
8. ЛИТЕРАТУРА

- [1] Mustafa Ozkan Yerebakan и Boyi Hu. „Human–Robot Collaboration in Modern Agriculture: A Review of the Current Research Landscape”. У: *Advanced Intelligent Systems* (2024.), стр. 2300823.
- [2] *topic_based_ros2_control*. https://github.com/PickNikRobotics/topic_based_ros2_control/tree/main Accessed: 2024-06-10.
- [3] *ROS2 Control*. <https://www.youtube.com/watch?v=4VVrTCnxvSw&t=130s>. Accessed: 2024-06-07
- [4] *pymap3d*. <https://github.com/geospace-code/pymap3d/tree/0bb4350d45929ebff69047260f7934b30b5e9bae>. Accessed: 2024-06-27.

Кратка биографија:



Милош Николић рођен је 04. јула 1999. године у Ваљеву. Дипломски рад на Факултету техничких наука из области роботике одбранио је 2022. године. контакт: nikolicmilos3610@gmail.com



Милутин Николић рођен је 18. јула 1984. године у Новом Саду. Докторску дисертацију одбранио је 15. маја 2015. године. Јула месеца 2020. године стекао је звање ванредног професора. Предаје неколико предмета из области роботике на Факултету техничких наука у Новом Саду. контакт: milutinn@uns.ac.rs