**UNIVERSITATEA “POLITEHNICA” DIN TIMISOARA**

**FACULTATEA DE AUTOMATICA SI CALCULATOARE**

**INGINERIA SISTEMELOR**

**PROJECT MBAPPE**

**PROIECT SINCRETIC I**

**AUTORI: Handrugan Radu**

**Leahu Jonathan**

**Humaciu Denis**

**Coordonatori: As. Ing. Emil VOISAN, Liviu-Aniel ALEXA**

**CUPRINS**

[**Introducere** 3](#_Toc188458444)

[**Prezentarea temei** 4](#_Toc188458445)

[**Tehnologii utilizate** 5](#_Toc188458446)

[**Ghidul programatorului** 6](#_Toc188458447)

[**Ghidul utilizatorului** 7](#_Toc188458448)

[**Testare si punere in functiune** 7](#_Toc188458449)

[**Prezentarea echipei** 8](#_Toc188458450)

[**Concluzie** 8](#_Toc188458451)

[**Bibliografie** 9](#_Toc188458452)

# **Introducere**

Acest proiect are ca scop explorarea si dezvoltarea competentelor in domeniul roboticii si al programarii prin utilizarea robotului de tip TurtleBot 3, care este o platforma robotica versatila, ideala pentru studierea algoritmilor de navigatie, mapare si control autonom.

In cadrul acestui proiect, ne propunem sa programam robotul pentru a indeplini o sarcina specifica primita in cadrul cursului, cum ar fi evitarea obstacolelor, explorarea autonoma a unui mediu necunoscut sau implementarea diferitelor functii de mobilitate. Prin implementarea acestor functionalitati, vom intelege mai bine integrarea senzorilor, controlul motoarelor si utilizarea framework-ului ROS (Robot Operating System).

TurtleBot 3 este o platforma robotica open-source, dezvoltata pentru cercetare, educatie si aplicatii de robotica. Este populara datorita dimensiunilor compacte, flexibilitatii si compatibilitatii cu Robot Operating System (ROS).

Caracteristici principale:

-Design modular: TurtleBot 3 poate fi personalizat cu usurinta. Se livreaza in doua modele de baza: Burger (versiune compacta) si Waffle (versiune mai mare si mai puternica).

-Compatibilitate cu ROS: Este complet integrat cu ROS, ceea ce permite utilizatorilor sa implementeze si sa testeze diverse algoritmi avansati.

-Senzori si camere: Utilizate pentru detectarea obstacolelor si mapare, dar si pentru procesare vizuala, detectare de obiecte si recunoastere.

-Motoare DC si controler: Au ca si scop controlul precis al miscarii, fiind capabil sa execute sarcini precum urmarirea unei traiectorii.

-Conectivitate: Suporta Wi-Fi si poate fi controlat de la distanta printr-un PC sau un dispozitiv mobil.

# **Prezentarea temei**

Proiectul propus, “FOTBALIST”, are la baza utilizarea unui robot TurtleBot 3 pentru a simula o actiune specifica unui joc de fotbal, utilizand tehnici de procesare vizuala si control autonom. Scopul principal al robotului este de a detecta si manipula o minge verde, avand ca punct final plasarea acesteia intre doi cilindri rosii, ce joaca rolul unei porti.

**Descrierea detaliata a sarcinilor:**

* Detectarea mingii rosii:

Robotul TurtleBot va folosi camera integrata pentru a captura imagini din mediul inconjurator. Aplicand tehnici de procesare a imaginii, robotul va identifica mingea verde pe baza caracteristicilor vizuale ale acesteia (culoare si forma). Algoritmul de procesare a imaginii va extrage informatiile necesare pentru a localiza mingea in spatiul 3D, astfel incat sa poata directiona robotul catre ea.

* Directionarea robotului spre minge:

Odata ce mingea a fost detectata, robotul va calcula traiectoria optima pentru a ajunge la aceasta, folosind algoritmi de navigatie si evitare a obstacolelor. Robotul va ajusta miscarile sale pentru a se alinia si a se apropia de minge, pregatindu-se pentru urmatoarea actiune.

* Rotirea in jurul mingii:

Dupa ce ajunge la minge, robotul va incepe sa se roteasca in jurul acesteia pentru a simula comportamentul unui fotbalist care isi pregateste o lovitura. Aceasta rotire va fi realizata cu ajutorul senzorilor si algoritmilor de control al miscarii, asigurandu-se ca robotul mentine o distanta constanta fata de minge.

* Detectarea celor 2 cilindri rosii:

In etapa urmatoare, robotul va trebui sa localizeze cei 2 cilindri rosii, care reprezinta poarta. Folosind aceleasi tehnici de procesare vizuala, robotul va identifica pozitia cilindrilor si va calcula un traseu pentru a-i aborda.

* Impingerea mingii intre cilindri:

Dupa ce cei 2 cilindri rosii au fost detectati si localizati, robotul va directiona mingea spre acest spatiu, urmand sa o impinga intre cilindri, simuland o reusita intr-un joc de fotbal. Controlul precis al miscarii si al fortei aplicate va fi esential pentru a atinge acest obiectiv.

**Metode utilizate:**

* Procesare vizuala a imaginii: Detectarea mingii rosii si a cilindrilor rosii va fi realizata folosind algoritmi de procesare a imaginii, precum detectarea culorilor, identificarea contururilor si segmentarea imaginii.
* Navigatie autonoma: Algoritmii de control vor asigura miscarea robotului intr-un mod precis si eficient, permitandu-i sa se deplaseze intre diferitele etape ale sarcinii.
* Controlul miscarii: Robotul va utiliza motoarele si senzorii pentru a controla directia si viteza miscarilor sale, fiind capabil sa navigheze in jurul mingii si sa o impinga spre poarta.

# **Tehnologii utilizate**

* **ROS 2**

ROS 2 este a doua generație a sistemului de operare robotic open-source, dezvoltat pentru a îmbunătăți funcționalitățile și performanța oferite de prima versiune (ROS 1). Este un ecosistem modular care permite dezvoltatorilor să construiască și să implementeze aplicații avansate de robotică. Printre principalele caracteristici ale acestuia se regasesc comunicarea in timp real, ROS 2 utilizeaza middleware-ul DDS (Data Distribution Service) care ofera suport nativ pentru comunicatii distribuite in timp real, necesare pentru aplicatii critice, compatibilitatea cu mai multe sisteme de operare cum ar fi Windows, Linux, MacOs, facilitand astfel dezvoltarea pe mai multe platforme. Fata de precedentul sau, ROS1, acesta vine cu imbunatatiri si pe partea de conectivitate, permitand controlul de la distanta prin intermediul WI-FI de pe un computer sau un dispozitiv mobil.

* **Visual Studio Code**

Pentru partea de cod am utilizat ca si limbaj de programare Python, un limbaj de programare de nivel inalt, cunoscut pentru sintaxa sa simpla si lizibila, iar ca si mediu de lucru am utilizat Visual Studio deoarece este un editor de cod sursa usor, rapid si extensibil ce poate fi utilizat pentru o gama larga de limbaje de porgramare, inclusive Python.

* **Gazebo**

Gazebo este un simulator avansat pentru roboti, utilizat pe scara larga in cercetare si educatie pentru testarea algoritmilor de robotica intr-un mediu virtual inainte de implementarea pe hardware real. Este open-source si ofera un mediu realist pentru simulari complexe, fiind adesea folosit impreuna cu Robot Operating System (ROS).

# **Ghidul programatorului**

Acest ghid ofera informatii esentiale despre implementarea proiectului „Fotbalist” utilizand TurtleBot 3, ROS si Gazebo. Scopul principal al proiectului este ca robotul sa detecteze o minge verde folosind camera integrata, sa se deplaseze catre aceasta, sa se roteasca in jurul ei pentru a identifica cei doi cilindri rosii (poarta) si sa impinga mingea intre acestia.

* Setarea mediului de dezvoltare

Se instaleaza ROS 2 si Gazebo pe sistemul de operare (in cazul de fata a fost folosit Ubuntu. Se configureaza TurtleBot 3 conform instructiunilor din TurtleBot 3 E-Manual, apoi se creeaza un workspace ROS pentru gestionarea pachetelor.

* Implementarea detectarii vizuale

Utilizeaza OpenCV pentru procesarea imaginilor capturate de camera robotului si se implementeaza detectarea culorilor pentru a recunoaste mingea verde si cilindrii rosii, urmand ca mai apoi sa fie implementata partea ce se va ocupa de miscarea robotului.

* Planificarea miscarii robotului

Va fi dezvoltat un algoritm pentru deplasarea catre mingea detectata folosind topicuri ROS pentru controlul motoarelor robotului, iar apoi va fi implementata o secventa care asigura miscarea de rotatie in jurul mingii pentru a localiza poarta.

* Simularea si testarea

Se va rula proiectul in Gazebo pentru a verifica functionalitatea functiei de detectare vizuala si de deplasare. Se ruleaza consecutiv proiectul, incercand sa fie gasita aparitia unor posibile bug-uri sau alte probleme ce tin de functionalitatea si integritatea robotului. In urma acestor testari, cand se va ajunge la un model stabil si functional se va face implemetarea acestuia pe un robot fizic, utilizand obiecte reale (minge si cilindrii).

# **Ghidul utilizatorului**

Pentru a rula simularea acestui proiect in Gazebo sunt necesari urmatorii pasi:

1. Se deschide un terminal si se va introduce urmatoarea comanda pentru a lansa Gazebo: ros2 launch turtlebot3\_gazebo turtlebot3\_world.launch.py
2. Se va deschide simularea in Gazebo unde se va regasi robotul TurtleBot3 cat si sfera verde (ce reprezinta mingea de fotbal) si cei 2 cilindrii rosii reprezentadn poarta de fotbal. Daca se doreste modificarea pozitiei mingii / portii va fi utilizat meniul din Gazebo.
3. Se va monitoriza modul in care robotul functioneaza. Fiind un robot autonom, nu este necesara interventia utilizatorului asupra acestuia. Dupa ce acesta termina procesul robotul se va opri.
4. Pentru a relua acest proces este necesara restartarea mediului virtual de simulare Gazebo.

# **Testare si punere in functiune**

Testare

* Am efectuat teste unitare pentru fiecare modul principal: procesarea imaginilor, detectarea formelor, logica de deplasare.
* Teste integrate: Verificarea comunicarii intre noduri ROS si raspunsurile corespunzatoare la intrarile utilizatorului.

Punere in functiune

* Instalati ROS2 si mediul Gazebo.
* Descarcati si configurati proiectul folosind comanda colcon build.
* Lansati nodurile folosind ros2 launch.

# **Prezentarea echipei**

Numele proiectului: **Project Mbappe**

Membrii echipei: **Handrugan Radu, Humaciu Denis, Leahu Jonathan**

Contributii si dezvoltarea proiectului:

* Instalarea sistemului de operare (Linux) si a celorlalte programe necesare in dezvoltarea proiectului: Leahu Jonathan
* Crearea si implementarea functiilor necesare pentru functionarea robotului: Humaciu Denis, Handrugan Radu, Leahu Jonathan
* Testarea si rezolvarea problemelor aparute in timpul developarii: Humaciu Denis
* Prezentarea Word si Powerpoint pentru prezentarea proiectului: Handrugan Radu

# **Concluzie**

“Project Mbappe” a reprezentat o experienta interesanta de a incerca si aplica tehnologii in domeniul roboticii si procesarii vizuale cu care nu am mai avut de aface pana in acest moment. Utilizand robotul TurtleBot 3 si functiile acestuia de a detecta si manipula un obiect in mod autonom, am reusit realizarea scenariului care simuleaza o actiune specifica unui joc de fotbal.

Prin realizarea acestui proiect, am demonstrat nu doar eficienta robotului in realizarea sarcinilor vizate, dar si importanta integrarii si optimizarii algoritmilor de procesare a imaginii si navigatie autonoma. Proiectul a contribuit, de asemenea, la aprofundarea cunostintelor in domeniul inteligentei artificiale si controlului robotic, oferind o baza solida pentru aplicarea unor tehnici similare cu roboti autonomi ce pot fi utilizati in diverse domenii industriale, automotive sau de divertisment.

# **Bibliografie**

* Lentin Joseph (2015), *Mastering ROS for Robotics Programming*
* A. Martinez, E. Fernández (2013), *Learning ROS for Robotics Programming*
* M. Quigley, B. Gerkey, W. D. Smart, *Programming Robots with ROS*
* Robotis e-Manual: <https://emanual.robotis.com/>
* The Construct: <https://www.theconstruct.ai/turtlebot3/>
* Gazebo: <https://gazebosim.org/home>
* ROS Wiki - TurtleBot 3: <https://wiki.ros.org/turtlebot3>
* ROS Discourse: <https://discourse.ros.org/>
* Robotics Back-End: <https://www.youtube.com/watch?v=Gg25GfA456o>