İhsan Doğramacı Bilkent University
Electrical and Electronics Engineering Department
EEE 493 - Industrial Design Project
Committee Meeting 2 Report
January 2020

Project Title: Accompanying Humans and Achieving Designated Tasks with Autonomous Mobile Robots using Swarm Intelligence

Group Members: Arda Yüksel, Bilgehan Başpınar, Cevahir Köprülü, Mert Acar, Oğuz Altan

Academic Advisor: Dr. Aykut Koç

Company Advisors: Dr. Bilge Kaan Görür & Muhammed Yüksel

Teaching Assistant: Emir Ceyani

Company Name: ROKETSAN A.Ş.

Company Info: ROKETSAN is a major Turkish weapons manufacturer and defense contractor based in the central Anatolian province of Ankara, incorporated in 1988 by Turkey's Defense Industry Executive Committee (SSİK) in order to establish the nation's industrial base on rocket technology. ROKETSAN is best known for its vast range of unguided rockets as well as laser and infrared guided missiles such as Cirit and UMTAS. The company also produces subsystems for Stinger and Rapier missiles and provides technology and engineering solutions for other integrated civilian and military platforms [1].

Contents

List of Figures

List of Tables

1 Project Summary

This project aims to to implement mobile robots that can participate in swarm behaviour based activities, in collaboration with ROKETSAN. These autonomous mobile land robots are intended to track a moving object, which is human in our case, and adjust their motion parameters such as speed and direction, with respect to the motion of the tracked object. Additional features of these robots are the ability to avoid nearby obstacles by detecting and dodging them, and to smoothly move on the terrain. The tracking part is to be performed by using Computer Vision and object tracking algorithms, using the real – time image data captured by an action camera, and the control of the robot is to be first simulated on AI – robotics simulation software Gazebo and afterwards to be implemented on the framework Robot Operation System (ROS). Lastly, for the obstacle avoidance concern, a Laser Imaging Detection and Ranging (LIDAR) sensor is to be used to detect the obstacles nearby the robot [2] and inform the control system on the robot. The software components of this project are running on AI-specialized mini computer Nvidia Jetson Nano, the brain of the robot. For the mechanical part of the robot, Dagu Wild Thumper 6WD All-Terrain Chassis [3] is to be used with specified motor drivers and battery, to provide optimal performance. The validation of the proposed implementation is to be tested on various simulation programs such as aforementioned software Gazebo. Computer Vision algorithms are first to be tested on team members' personal computers, before implementing them on Jetson Nano [4] and the mechanical structure is to be tested on the laboratory environment. The robot is expected to track humans, avoid obstacles and run on terrain with reasonable speed and performance, carrying pre-determined items.

2 Motivation and Novelty

These autonomous mobile land robots are intended to be used for various military services of Türk Silahlı Kuvvetleri (TSK), or Turkish Armed Forces, in English. ROKETSAN is trying to create and implement autonomous systems that can be used for several tasks such as accompanying soldiers in the field such as tracking and scouting, transporting equipment and following troops, search and rescue operations and gathering military intelligence. Carrying out these tasks with military personnel risks casualties and is inefficient for economic reasons. This issue has significant importance for national armies in the world and various military and defense companies around the globe try to solve this issue by developing different technologies [5] that aims to decrease the aforementioned costs. One of the solutions is developing autonomous mobile robots [6] that can be used to yield lower costs and an effective alternative for human life and this solution is what ROKETSAN intends to obtain in collaboration with our project team. To increase the functionality and enriching these proposed autonomous systems, we also work on implementing swarm behavior and joint intelligence that significantly boost the robustness of the performance and increase the feasibility of accomplishing military tasks, stated above. As a consequent step of successfully completing our project, ROKETSAN plans to use these autonomous mobile land robots in the military field as a stand-alone project, meaning that these explained autonomous systems are to be sustainable by

themselves without needing any human assistance. As swarm intelligence is one of the main parts of our project, ROKETSAN may increase the number of robots in the swarm and add additional features and capabilities to the robots which can be useful in the military service.

The target end-user of our product is the military personnel of TSK. According to our scenario, these autonomous robot swarms have several duties in the field, such as following the soldier that the swarm is assigned to with avoiding obstacles, as well as calculating and tracking its path autonomously, carrying military equipment, understanding military personnel's specific directions such as stop, go right and left and proceed. Therefore, the military service, especially soldiers in the field will use and command these swarms. As it is told by our ROKETSAN advisors, the cost of the robots that we work on in our project is affordable and cost efficient, specifically the hardware components and equipment are financially efficient and we use free open source software technologies. Therefore, as ROKETSAN produces the defense industry technologies for TSK [7], it can also be stated that the only cost of this project is the total cost of used equipment. The robots in this project are autonomous and therefore, there is no need for any technical instructions for using it in the field, any soldier can use it without having any technical background. A situation where the military personnel with technical background may be needed is the case of any breakdown and malfunctioning of the systems. As the robots in our projects consist of separate modules, the repair is not complex and finding the cause of malfunctioning is not time consuming.

As we have shown and explained in our project presentations, there are several projects which intend to develop similar technologies to our, such as human identifying drones [8] and drone swarms moving in pre-determined trajectory to accomplish various tasks [9]. Comparing them to our project, firstly it can be stated that our project includes many stages and parts, while these similar projects perform only a few methods that we want to have in our project. In other words, these similar projects intend to develop one or two of the functionalities that we intend to have, they do not contain all of the techniques and algorithms that we aim to have on our robots. Therefore, the cost of our systems may be different from those of these projects. An example project, called Real-time 3D Human Tracking for Mobile Robots with Multisensors, developed in The University of Technology, Sydney, Australia, aims to perform similar tasks like we do [10]. Comparing speed and maximum weight carrying capabilities, the robot mentioned in this paper can speed up to 2.88 km/h and carry up to 2 kg, meanwhile our robot can speed up to 7 km/h and carry up to 5 kg. Another advantage of our project is that due to its highly complex structure and integrity of different tasks and functionalities, it can accomplish a high variety of tasks, while this may also result in a disadvantage as due to high complexity of our project, the implementation and maintenance is more difficult to handle.

Due to the integrity of many technologies and complex structures, our project can bring many novelties and innovations. For example, the concept of autonomous robots having swarm intelligence is quite innovative in the military technologies in the world. The aforementioned similar products have several problems especially in the implementation of swarm technology, for instance, the importance of avoiding collisions of swarm members arises frequently. In our project, we intend to solve these problems by developing new algorithms and technologies. For the patent concerns, there are not any patented solutions to the stated problems. However, we think that especially the swarm technology part in our project may be a powerful candidate for the patent, as it brings many

innovations and novelties.

3 Requirements

3.1 Functional Requirements

1. Our robots should be able to follow a soldier that walks or starts to run therefore the mechanical units are expected to achieve 5 km/h speed. The specifications for these units state the maximum speed as 7 km/h. However, under load and on terrain conditions, we require the robot not to drop below 5 km/h.

2. Mechanical units are expected to carry 4-5 kg of load.

3. Our model is required to detect the target (marked soldier) from no further than 15 meters with a minimum confidence of 45 percent.

4. Under full operation the algorithm is expected to run at 12 frames per second.

5. As for the simulation part, we require a LIDAR scan rate of 7-9 Hz.

3.2 Non-Functional Requirements/Constraints

1. We are planning to make 2 robots for 3323 TL, including the components that ROKETSAN has provided to us. We do not have a financial limitation imposed by the company, the only thing that they say about the financial part of the project is that the units that we want to purchase are reasonable and we can go on as we planned.

2. Each robot is expected to carry 4-5 kg load. Therefore, taking the motor capabilities into consideration, the total weight of the chassis and mechanical units better not exceed 3 kg. Dagu Wild Thumper 6WD, the robot chassis, weights 2.5 kg and we plan to mount Jetson Nano, LiPo battery, motor driver, powerbank, camera and Lidar on the robot chassis, therefore the total weight of the robot without any additional load is expected to be about 3 kg.

3. Mobile Units are expected to draw a maximum of 18 A of current under full load and no more than 6.6 A during jump-start. These units are expected to operate for approximately 25 - 30 mins with one charge of LiPo Battery.

4. Robots are designed to help armed forces by following them on terrain operations, therefore a robot chassis with tires that are suitable for terrain conditions picked for such missions.

5. Our plan A is to use a LiPo battery as the power supply of the robots, yet, LiPo batteries can be somewhat dangerous to handle sometimes. Overcharging a LiPo battery can even cause fire, thus one needs to be cautious working on it.

6. We do not have any health constraints or requirements.

7. As mentioned earlier, our main objective is to help armed forces by following them on terrain operations. Therefore, if this project is to be used in real life, its design would be affected by global and political changes, which also would change the defending style.

8. Our project will be compatible with Standard for Connected, Automated and Intelligent Vehicles: Overview and Architecture, IEEE Std P2040.1 [11] and P2040.2 [12] in terms of automation.

4 Big Picture

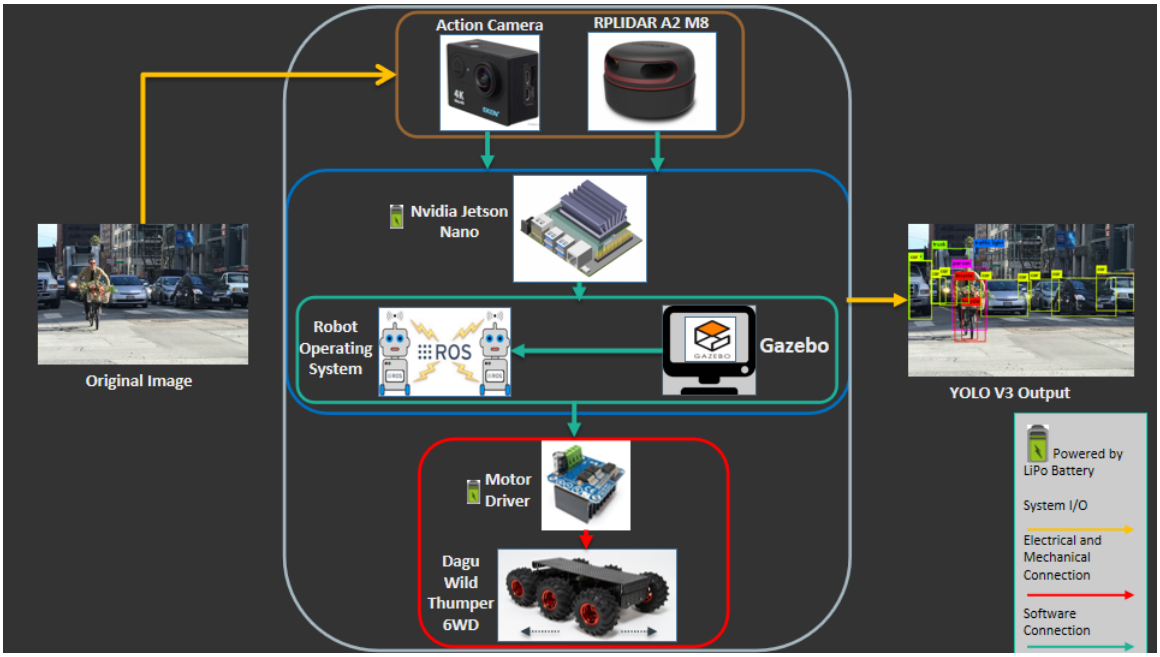

Figure 1: Big Picture

The system senses the world using action camera and LIDAR. Camera provides real-time video footage to the system and LIDAR detects the surrounding objects in the field. These data are processed in the Robot Operation System (ROS) working on Nvidia Jetson Nano computer. YOLO real-time object detection system software takes the input video footage taken with action camera and detects the human in the video frame. Meanwhile, the point cloud data taken with LIDAR is converted to useful data in ROS and the nearby obstacles are detected. Finally, the control algorithms working on ROS leads the robot and perform motion operations, for instance tracking the computed path and avoiding obstacles. These control algorithms provide digital input data for the motor drivers that controls motors of the robot. The power of the system is provided by the LiPo Battery. For the simulation purposes, Gazebo Simulation Software is used, running on our computers. In this simulations, the virtual model of the robot in our project is tested in a virtual world with several obstacles. The developed technologies and algorithms is first tested

in these simulations and possible malfunctions are observed and debugged. Consequently, these technologies are deployed to real robots controlled by ROS.

5 Methods and Implementation Details

5.1 Work Breakdown Structure and Project Plan

This project integrates four different milestones. These milestones are chosen as the building blocks of the project as a whole and indicate main objectives of the project. Each milestone will be broken down to be explained in detail and given details about the time and people allocation. These allocations will be demonstrated using a Gantt chart integrate the progress better. The Gantt chart now demonstrates the individuals assigned to sub-tasks. In addition to that, instead of demonstrating the schedule by the months and now it is in 2 weeks format to show the progress in details.

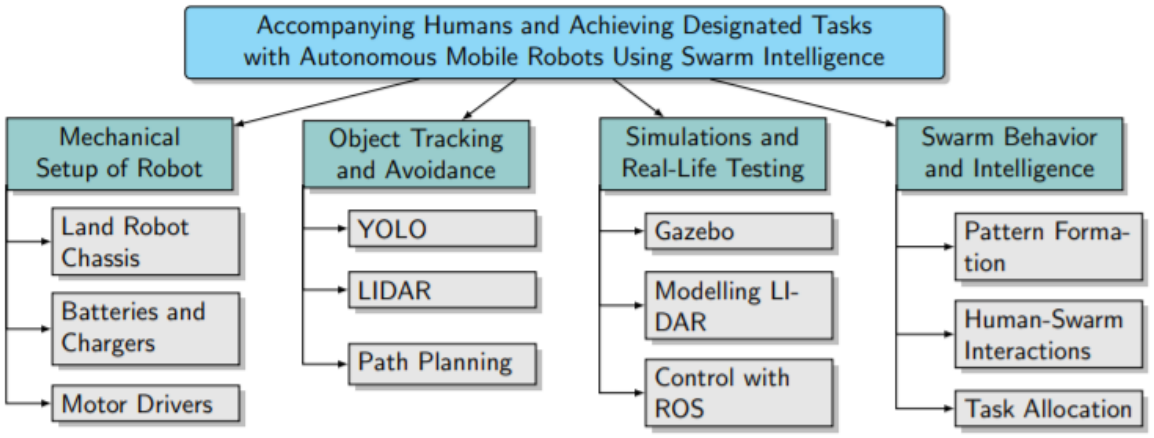

Figure 2: Work Breakdown Structure of The Project

5.1.1 Mechanical Setup of Robot

Mechanical setup of robot comprises three sub-tasks: Land Robot Chassis Assembly, Batteries and Chargers, Motor Drivers Selection and Testing. The plan can be seen in the gantt Chart provided in Figure 3

5.1.1.1 Land Robot Chassis

Land Robot Chasis subtask includes the assembly of the robot setup by combining the robot parts. Firstly the robot model is selected and then the orders are commenced. The model for this part

is changed into Dagu Wild Thumper 6WD 34:1 compared to Committee Meeting 1. The main criterion is the robot's ability to handle the terrain conditions as the main scenario demands. The selected model satisfies this condition. This sub-task is completed before the 1st of January, 2020. The Dagu Model is assembled. Therefore, criteria of success which was the establishment of working model is accomplished. This milestone was under Oğuz Altan's responsibility.

5.1.1.2 Batteries and Chargers

In this part, using the model selected for the first sub-task under Mechanical Setup of the Robot, the necessary voltage and current level for each component is calculated. For the calculations datasheet of the Dagu Wild Thumper is used as basis. The assembled robot is tested for calculated values and equipment compatible with the measured values are selected. The purchase of the batteries and chargers are expected be completed by the 15th of January. For the batteries, LiPo batteries are chosen.

The timeline associated with this sub-task is expanded due to the change in model selected for Chassis. This sub-task is under Oğuz Altan's responsibility. Success criteria for this sub-task is confirming the desired power levels through the purchased batteries and chargers.

5.1.1.3 Motor Drivers

In this part, the motor drivers for the selected Chassis is aimed to be selected. In order to reach a conclusion data sheet of the selected chassis is examined. Then, the robot is tested in the lab for both Batteries and Chargers and Motor Drivers sub-tasks. After the testing procedure necessary components are identified for the purchase. The order of the components are expected be given. This sub-task will be completed before 15th of January.

The timeline associated with this sub-task is expanded due to the change in model selected for Chassis. This sub-task is under Oğuz Altan's responsibility. Success criteria for this sub-task is moving the robot through the usage of purchased motor drivers.

5.1.2 Object Tracking and Avoidance

Under this milestone, YOLO, LIDAR and Path Planning are given as sub categories. The previous categories were YOLO, Mean Shift Tracking, LIDAR and Potential Fields Method. The mean shift tracking algorithm is discarded as it was replaced with a simpler algorithm using YOLO output. Potential Field Method is found to be insufficient for the scope of the project. Therefore it is adjusted as Path Planning. The timelines are altered in order to compensate the changes as it can be seen under Figure 3.

5.1.2.1 YOLO

The first sub-task concerns about getting YOLO environment ready. YOLO normally comes with a fairly simple installation process. However, using default installation process, it does not process real time data. Therefore one needs to compile it from source using Nvidia's CUDA and OpenCV

which is also built from source with CUDA. Once these steps are achieved then a webcam feed can be an input to YOLO network and real time object detection can be used.

This subsection now includes the tracking part by replacing the Mean Shift Tracking compared to Committe Meeting 1. Currently using the location of the target provided by YOLO, the angle for rotation is calculated as opposed to the usage of Mean Shift Tracking.

The criteria of success was achieving 12 FPS in Computer Vision applications with 45 percent success rate in the range of 15 meters. This criteria has been tested both for live feed and sample video taken from the Action Camera which can be seen in the equipment list. This sub-task is completed before the 15th of December. It was assigned to Mert ACAR.

5.1.2.2 LIDAR

The second sub-task consists of selecting the LIDAR and visualizing the real world data using the selected LIDAR. The LIDAR was selected before Committee 1 as RPLIDAR A2M8. After obtaining the model from ROKETSAN, using the provided SDK, real world data is analyzed through the model. The obtained data is tested and compared to expected outcomes. Visualization is achieved.

The success criteria was providing the LIDAR output that can be used for path planning case. After comparing the obtained data to the expected output accurate representation of the real world is attained. This milestone was expected to be completed before 1st of January. This aim is achieved by Bilgehan Başpınar.

5.1.2.3 Path Planning

This milestone is replaced from Potential Field Method to Path Planning and its timeline is extended by 2 weeks due to change in the application procedure of the algorithm. It is found that previous milestone had local-minima problem which can cause dead-ends in the Path Planning procedure. Path Planning is expected to sustain robot to follow the target without interfering obstacles in the path. This part integrates the LIDAR and YOLO outcomes. It integrates Timed Elastic Band algorithm for that purpose.

Due to increasing complexity of this sub-task 3 people are assigned to this implementation: Arda Yüksel, Cevahir Köprülü and Bilgehan Başpınar. Previously, it was assigned to Cevahir Köprülü and he provided the reason the to drop the Potential Field Method for the sake of the project. As it is explained, Path Planning will be succeeded after implementation of the algorithm that can follow the target and avoid obstacles accurately. This criteria will be tested in the Gazebo Simulations. This task will be accomplished before 1st of February.

5.1.3 Simulations and Real-Life Testing

The simulation and real-life testing of the project will be broken down to three sub-tasks which are Gazebo, Control with ROS and Modelling Lidar. As now the gantt chart shows the middle of the months the deadlines for the milestones can be seen more accurately. The simulation part will cover the span of project as the testings will commence for the second semester as well.

5.1.3.1 Gazebo

In this part the installation and basic comprehension of the ROS and Gazebo was aimed initially. This milestone is expanded into demonstration of the DAGU Wild Thumper with the selected LIDAR as the products are decided. After the selection of the equipment, Gazebo environment are prepared for the testing of possible simulations.

This sub-task was given to Arda Yüksel before Committee Meeting 1. After the necessity of simulations in Path Planning algorithms, this milestone is now corralated to Path Planning Simulations in Gazebo. Thus, its timeline is expanded into the beginning of 1st of February, 2020. The success criteria was running sample simulations in Gazebo for Wild Thumper and provide feedback for the applied algorithms. Now this task is under Arda Yüksel and Cevahir Köprülü.

5.1.3.2 Control with ROS

The aim of this sub-task is to simulate the PID controller associated with the selected chassis. It is expected to be completed before 15th of February, 2020. The success criteria is obtaining stable control over the designed DAGU Wild Thumper Gazebo Simulation. This sub-task is assigned to Oğuz Altan.

5.1.3.3 Modelling LIDAR

This milestone aims to obtain the LIDAR data of the simulated robot and visualize the outputs in Point Cloud format through RVIZ. In addition to that, using programs to obtain the LIDAR outputs via ROS was expected to be completed.

Bilgehan Başpınar was responsible for Modelling LIDAR and she finished her duties before the 1st January, 2020 as expected.

5.1.4 Swarm Behaviour and Intelligence

The last milestone of the project is achieving swarm behaviour with multiple robots. In this task 2 robots are expected commence activities using Swarm Intelligence. This milestone is divided into three sub-tasks which are Pattern Formation, Task Allocation and Human-Swarm Interaction. All of these sub-tasks will be launched in the second semester.

5.1.4.1 Pattern Formation

The aim of this sub-task is to control the positions of the 2 robot system with respect to each other using Beacons. It is expected to be completed before 1st of May, 2020. The success criteria is the application various patterns for specific operations which will be decided in the second semester. This sub-task is assigned to Arda Yüksel and Bilgehan Başpınar.

5.1.4.2 Task Allocation

The aim of this sub-task is to control the work distribution and master-slave relationship of the robots. It is expected to be completed before 1st of June, 2020. The success criteria is the achieving dynamic allocation of master and slave robot. This sub-task is assigned to Cevahir Köprülü and Oğuz Altan.

5.1.4.3 Human Swarm Interactions

The aim of this sub-task is to control the master robot through gestures. It is expected to be completed before 1st of June, 2020. It is an optional task that will be decided on the state of the project in second semester. This sub-task is assigned to Mert Acar.

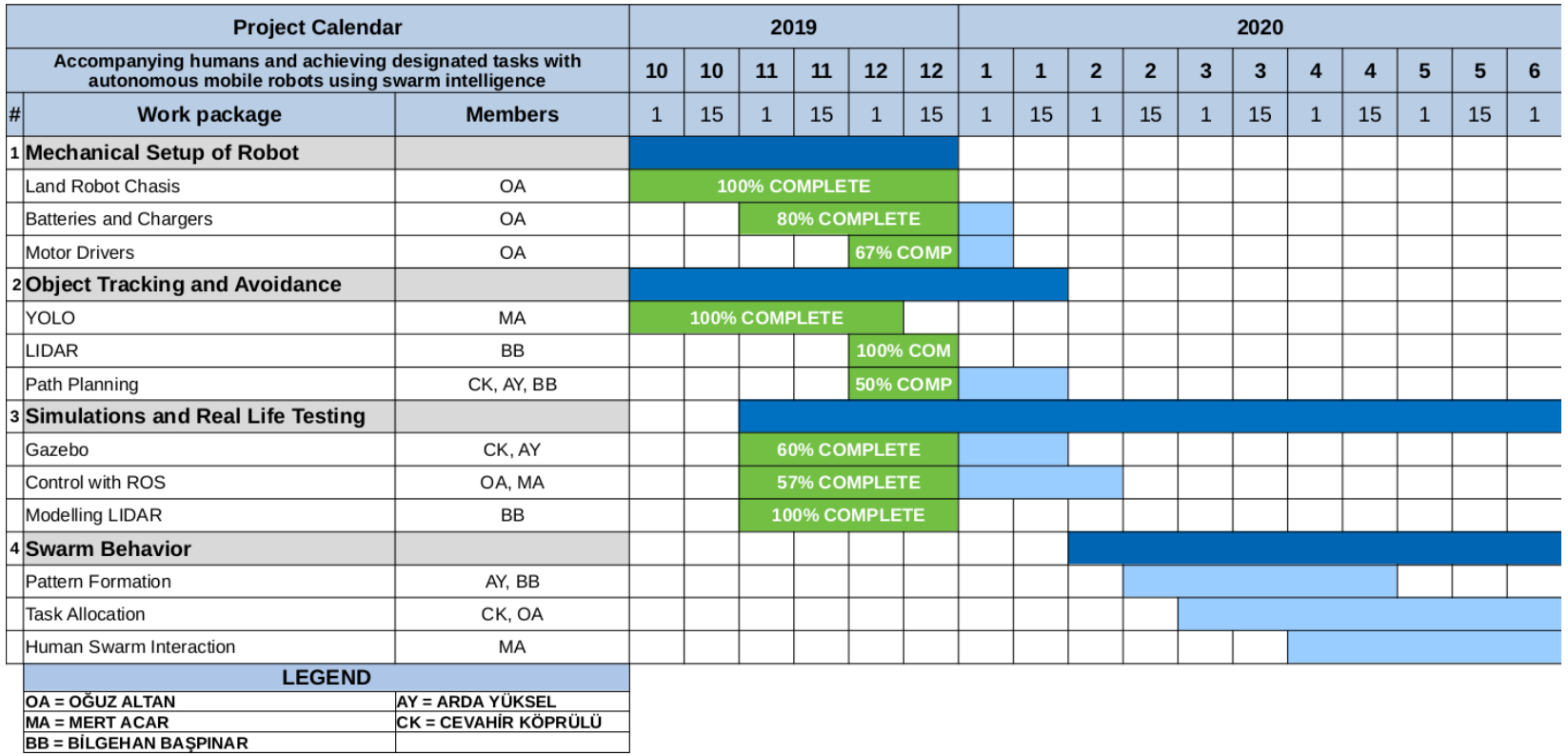

Figure 3: Gantt Chart of The Project

5.2 Methods and Progress

5.2.1 Mechanical Setup of Robot

5.2.1.1 Land Robot Chassis

Up to CM1

The mechanical structure and chassis of the robot is discussed and several mechanical designs are proposed, considering specifications and requirements. The robot has to carry at least 4-5 kg load and in addition, move on a terrain with obstacles, without any malfunctioning. the first plan was to construct the robot ourselves, by first making the design, then obtaining the required construction materials and finally build the robot. After discussions with ROKETSAN, the plans have been changed and it is decided to make a market research about a robot chassis that is appropriate for aforementioned mechanical requirements. Dagu Wild Thumper 4WD is selected as the robot chassis.

Between CM1 and CM2

The model has been updated to Dagu Wild Thumper 6WD from 4WD, with the proposal of ROKETSAN.

Figure 4: Dagu Wild Thumper 6WD All-Terrain Chassis, 34:1

The gear ratio is also updated to 34:1. This robot chassis is expected to carry around 7 kg of payload. In addition, it is a differential-drive chassis, meaning that turning is accomplished by driving the motors on the two sides of the platform at different speeds. For instance, steering to right is managed by changing the relative speeds of right three and left three wheels of robot, where the speed of right ones are higher than the left ones.

Future Plans

When all of the necessary components are obtained, they will be mounted on top of the robot chassis and the electrical connections will be completed.

5.2.1.2 Batteries and Chargers

Up to CM1

Two candidates for battery were discussed, which were LiPo battery and NiMH (Nickel–metal hydride) battery. As LiPo battery is lighter and more robust compared to NiMH battery, LiPo is selected as the battery of the system. The exact product is planned to be selected after having decided the robot chassis, motors and motor drivers.

Between CM1 and CM2

For the powering part, LiPo (Lithium Polymer) battery technology is chosen as the power source for the motors, as these batteries are lightweight and robust. The 7.4V 2s 5200 mAh 30C LiPo battery is planned to be used in the project, it is suitable to the power specifications of the motors and the chosen motor driver. The estimated time of operation of the robot using mentioned LiPo battery is 25-30 minutes. For the time efficiency reasons, it is planned to acquire 2 of these LiPo batteries, so that while one of them is getting charged, the other one can be used to power up the robot. Finally, for powering the Jetson Nano, which consumes 10W and requires 5V/2A, the proposed solution is to use a powerbank, plugged to Jetson Nano via USB cable. The Xiaomi 10000 mAh powerbank , which can provide 5V/2A [13], is appropriate for this aim.

Figure 5: 2s 5200 mah 30C LiPo Battery 7.4V

Future Plans

The batteries will be brought and consequently they will be started to be tested on motor drivers and motors of the robot.

5.2.1.3 Motor Drivers

Up to CM1

The motor drivers were planned to be selected after having decided the robot chassis and motors.

Between CM1 and CM2

The three motors on each side of the robot are wired in parallel, therefore only two channels of motor control are required to get this chassis moving. The motors are intended for a maximum nominal operating voltage of 7.2 V (2V minimum), and each has a stall current of 6.6 A and a no-load current of 420 mA at 7.2 V. The motors briefly draw the full stall current when abruptly starting from rest (and nearly twice the stall current when abruptly going from full speed in one direction to full speed in the other). The BTS7960B motor drivers are chosen according to these specifications. These single-channel motor drivers can provide up to 40A and works in the the operating voltage interval 5.5V - 28V [14]. As the motor driver is single channel, it is planned to use 2 of these motor drivers, one for each side of the robot, namely 3 motors wired in parallel. In addition, the mentioned motor driver can be used with Pulse Width Modulation (PWM) technique, and this technique is planned to be used for the motion control of the robot.

Figure 6: BTS7960B 40A Motor Driver Module

Future Plans

As soon as the aforementioned motor drivers will be provided to us, they will be mounted on the robot chassis. The electrical connections will be completed and using PID controller with ROS, the motor drivers will control the motors and wheels of the robot.

For the mechanical setup of the robot, there are possible risks. As this part is about batteries, motors and motor drivers, which are all working on considerably high voltages, currents and amperes, there may arise electrical connection problems and power specification mismatching. For instance, if motor driver provides more current than motors of the robot can handle, it may burn the motors and rend them dysfunctional. Another problem about power mismatching may be the scenario that LiPo battery can not provide enough power, when it is on low charge, to the motor drivers, thus robot does not move at all, despite Jetson Nano works and can send appropriate

control commands. These can be solved by limiting the output current of the motor driver and replacing the current LiPo battery with the backup battery, respectively. Another bad scenario about LiPo batteries is caused by the nature of the battery itself. If the charge of the battery goes below a critical level, depends on the specifications of battery, to work with that battery reduces its lifespan and may even cause explosion [15]. To use LiPo batteries always above the critical charging level is an important practice.

5.2.2 Object Tracking and Avoidance

5.2.2.1 YOLO

Up to CM1

Target tracking consists of two fundamental parts, the 1st one being the detection of the target and the 2nd part being tracking the location of it. To construct a complete method, we begin with proposing the Deep Convolutional Neural Network architecture YOLO V3 as the image classifier/detector. YOLO V3 takes a high resolution image and outputs its predictions about objects that might be on the image, and about the bounding box of the object on the image. This bounding box prediction consists of predicting 4 coordinates (t_x, t_y, t_w, t_h) for each bounding box. Given the offset of the cell (c_x, c_y) and prior on width p_w and p_h, the predictions correspond to:

$$b_x = \sigma(t_x) + c_x \tag{1}$$
$$b_y = \sigma(t_y) + c_y \tag{2}$$
$$b_w = p_w e^{t_w} \tag{3}$$
$$b_h = p_h e^{t_h} \tag{4}$$
$$\tag{5}$$

An example of bounding box prediction is given below in Figure 7

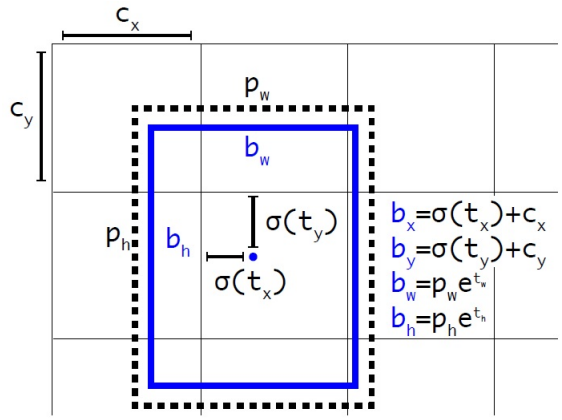

Figure 7: Bounding boxes with dimension priors and location predictions [16]

To deep deeper into the logic of class predictions, each box is designed to predict the classes the bounding box may contain, thus it essentially utilizes multilabel classification. This is quite helpful for our project, since although initially our objective is to detect a human-being, one of the complementary targets of the project is to predict various tools that the targeted human carries with him/herself. Considering the availability of multiclass prediction, we can benefit from Transfer Learning in order to classify more customized objects.

Considering the limitations of the main board, Jetson Nano, in order to avoid lagging in real-time detection, we will utilize another YOLO architecture called Tiny-YOLO, whose structure is given below in Figure 8:

```
layer     filters    size              input                output
    0 conv     16  3 x 3 / 1   224 x 224 x   3   ->   224 x 224 x  16
    1 max          2 x 2 / 2   224 x 224 x  16   ->   112 x 112 x  16
    2 conv     32  3 x 3 / 1   112 x 112 x  16   ->   112 x 112 x  32
    3 max          2 x 2 / 2   112 x 112 x  32   ->    56 x  56 x  32
    4 conv     16  1 x 1 / 1    56 x  56 x  32   ->    56 x  56 x  16
    5 conv    128  3 x 3 / 1    56 x  56 x  16   ->    56 x  56 x 128
    6 conv     16  1 x 1 / 1    56 x  56 x 128   ->    56 x  56 x  16
    7 conv    128  3 x 3 / 1    56 x  56 x  16   ->    56 x  56 x 128
    8 max          2 x 2 / 2    56 x  56 x 128   ->    28 x  28 x 128
    9 conv     32  1 x 1 / 1    28 x  28 x 128   ->    28 x  28 x  32
   10 conv    256  3 x 3 / 1    28 x  28 x  32   ->    28 x  28 x 256
   11 conv     32  1 x 1 / 1    28 x  28 x 256   ->    28 x  28 x  32
   12 conv    256  3 x 3 / 1    28 x  28 x  32   ->    28 x  28 x 256
   13 max          2 x 2 / 2    28 x  28 x 256   ->    14 x  14 x 256
   14 conv     64  1 x 1 / 1    14 x  14 x 256   ->    14 x  14 x  64
   15 conv    512  3 x 3 / 1    14 x  14 x  64   ->    14 x  14 x 512
   16 conv     64  1 x 1 / 1    14 x  14 x 512   ->    14 x  14 x  64
   17 conv    512  3 x 3 / 1    14 x  14 x  64   ->    14 x  14 x 512
   18 conv    128  1 x 1 / 1    14 x  14 x 512   ->    14 x  14 x 128
   19 conv   1000  1 x 1 / 1    14 x  14 x 128   ->    14 x  14 x1000
   20 avg                       14 x  14 x1000   ->  1000
   21 softmax                                        1000
   22 cost                                           1000
```

Figure 8: Tiny-YOLO Architecture [17]

Our first implementation was compiling YOLO on Jetson Nano with CUDA and OPENCV, and then giving the network a generic image of Figure 9:

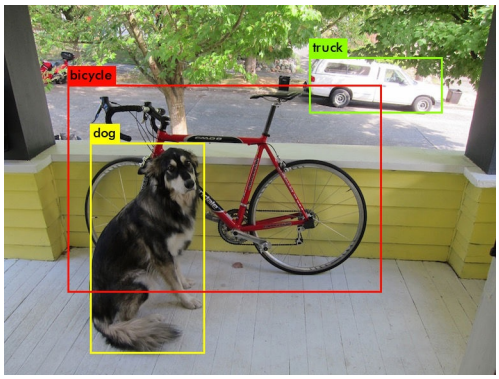

Figure 9: Tiny-YOLO Testing

Following this step, we connected our 4K action camera to Jetson Nano in order to do a real-time testing, and took the following frame given in Figure 10 from the tested live-feed:

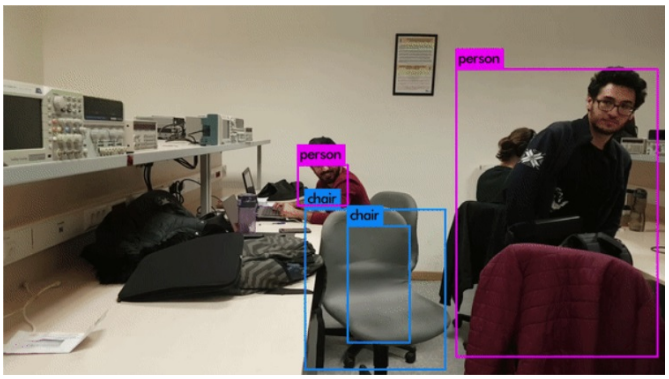

Figure 10: Tiny-YOLO Real-Time Test

As of this testing, we completed the real-time tests of Tiny-YOLO and successfully showed that Jetson Nano can run Tiny-YOLO in 20-22 frame per second. Therefore, our next step will be tracking already detected bounding boxes in a computationally efficient manner.

In order to fulfil this requirement a tracking algorithm called "Mean Shift Tracking" was to be employed. Mean shift is a semiautomatic tracking algorithm that is based on an iterative scheme.[18] The main idea behind mean shift is to maximize correlation between RGB color histogram of the original target in two consecutive frames.

In the mean shift algorithm the movement generated in each frame is in fact the gradient vector at a certain point on the Bhattacharyya coefficient surface. Therefore, tracking the object in the consecutive frame is actually the same as trying to find the peak along the gradient vector on the Bhattacharyya coefficient surface [18].

Between CM1 and CM2

The YOLO engine was constructed in Python programming language which is optimal for prototyping however, its lack of a compiler makes it rather slow for production grade development. Therefore, with appropriate changes, the code base was compiled in C language and the performance of the target detection was increased up to 55 frames per second. This performance increase is important because it provides a performance buffer for the more complex algorithms to be implemented. As the path planning and LIDAR sampling is incorporated to the engine the performance will suffer and this buffer will help Jetson Nano keep the minimum frames per second limit for reliable object detection (12 FPS).

The tracking was determined to be done using the mean-shift algorithm. However, this plan needs change due to the inferior nature of mean-shift algorithm. Since the algorithm relies on the simple borders that encapsulate the shape and color of the object, for complex targets like humans it is very difficult to obtain reliable results. Therefore, in an environment with multiple humans in the field of view of the robot, the master unit will determine the target it will follow with a

marker. This procedure will also be done using the robot's camera. The target will be equipped with a yellow-and-black-striped belt that will act as a marker for the robot. In the choice of the marker, the important criteria was it to be easily distinguishable in nature with a simple pattern. The detection of the marker is done by a kernel based line detector with parameters fine tuned to match the gradients of the striped belt. The kernel function is centered around the origin and matches the lines on the belt with a size of 7 x 7.

For scalability, multiple kernel functions with varying sizes like 3 x 3, 5 x 5 and 7 x 7 have planned to be employed however, as the target moves further away from the robot's camera, the smaller kernels did not perform as expected while also damaging the performance heavily since the smaller kernel convolutions are expensive to compute. Therefore, a 7 x 7 fixed symmetric kernel is employed with the color matching algorithm of the mean-shift method to compensate for the scalable target detection problem. However, the color matching presents itself with calibration and performance fluctuation under different light conditions. In order to aid this problem, a reflector coating will be used to help the robot distinguish the pattern.

Once the engine is performing as expected, the output of the algorithm is ported to the ROS framework. This framework provides the simple messaging service between computation nodes. A computation node is a program that controls an aspect of the robot's main framework. The differential drive node that is incorporated in Gazebo simulation software is responsible for driving the motor drivers of the robot and expects the commands from the ROS messaging service. By piping the control output from the algorithm to this messaging service the simulation is connected to the camera module and the simulated unit tracks the person in front of the camera.

Future Plans

Since the frame rate and confidence requirements are met, Future plans for this objective include the integration of all the control modules on Jetson Nano to be stress tested for reliable performance.

5.2.2.2 LIDAR

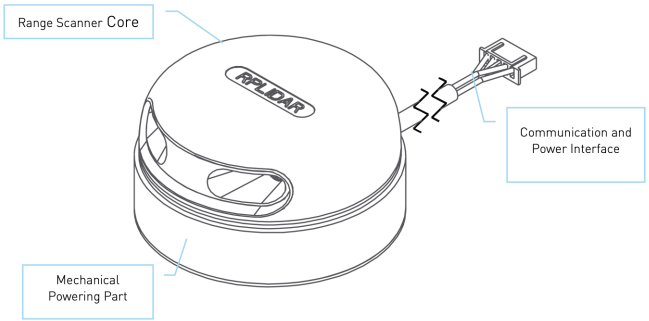

Figure 11: The structure of RPLIDAR A2M8 [19]

For tracking and object avoidance purposes, a Lidar of model RPLIDAR A2M8 is being used. RPLIDAR A2M8 measures the distance varying with the angle by using laser triangulation ranging

principle. During every ranging process, the RPLIDAR emits modulated infrared laser signal and the laser signal is then reflected by the object to be detected. The returning signal is then sampled by vision acquisition system in RPLIDAR and the DSP embedded in RPLIDAR starts processing the sample data and outputs distance value and angle value between object and RPLIDAR via communication interface. [19]

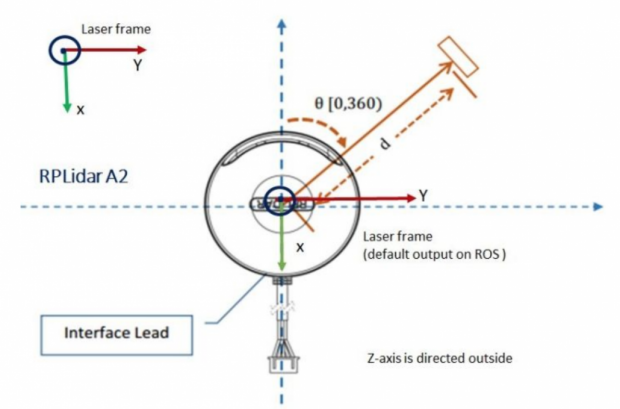

Figure 12: The structure of the data given by RPLIDAR A2M8 [19]

This (distance, angle) data in the form of a point cloud is to be used by the path planning algorithm.

Up to CM1

In this period, the working principle of Lidar was studied. Although RPLIDAR A2M8 has already relevant documentation and libraries such as and most most importantly SDK, it is not compulsory to complete this step, yet it is important to beware of the working algorithm in the sense that if a problem occurs, we should at least be able to know or search where it is. In this period, the algorithmic system behind Lidar that links the output to the input was found to be as follows:

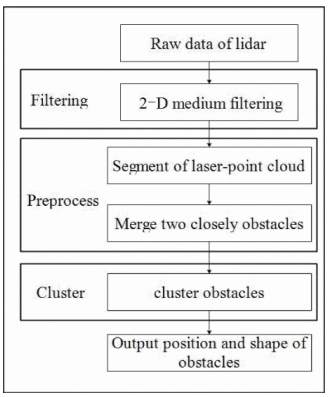

Figure 13: Working principle of Lidar [20]

As it can be understood from the figure above, the output which consists of positions and shapes of obstacle is obtained from the raw data in three major steps: filtering, preprocessing and clustering. In the filtering part the main objective is to remove the noise which is supposed to ease the preprocessing step, and for that, median liters are being used. On the other hand, each data point in the distance map generated by the Lidar has a coordinate. In the second step which is preprocessing, the relative distance to the Lidar corresponding to one coordinate exceeds the maximum range, then such points are placed into different blocks. If it is not, such points are placed into the same block. The final step is done to simplify the complex point cloud data. For that, the blocks generated in the preprocessing step are classified into line, circle and rectangle categories based on the detection algorithm. [21]

Between CM1 and CM2

RPLIDAR has been modeled in simulation which is to be mentioned later and the actual Lidar has been successfully operated using the SDK written for the RPLIDAR A2 family.

Inside the SDK there are three main functions defined: ultrasimple, simplegrabber and framegrabber. Among these, ultrasimple simply connects to an RPLIDAR device and outputs the scan data to the console. Simplegrabber, demonstrates the process of getting RPLIDAR's serial number, firmware version and healthy status after connecting the PC and RPLIDAR. Then the demo application grabs two round of scan data and shows the range data as histogram in the command line mode which is built with asterisk symbols. The final one, framegrabber, shows the laser scans in the GUI such as the following image:

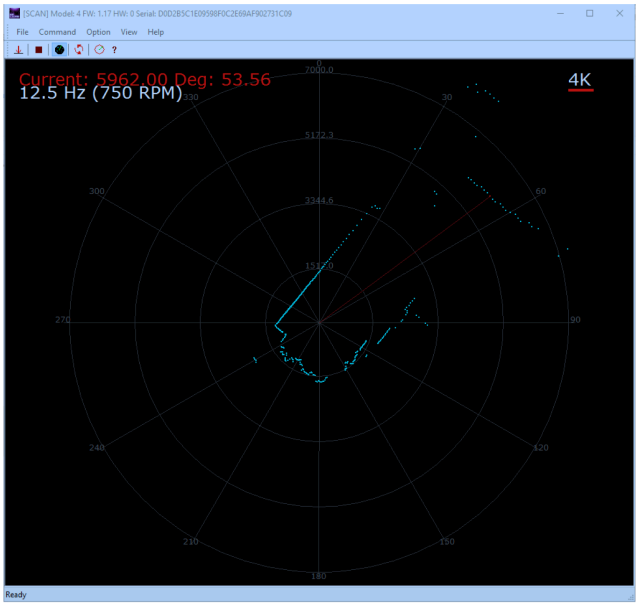

Figure 14: The debugging GUI of RPLIDAR [19]

Future Plans

The modeling of the RPLIDAR in the simulation environment Gazebo and getting data from the real Lidar goals are achieved. From now on, what is supossed to be done is to integrate the real and simulation Lidar data into the path algorithm once it is mastered.

As for the possible risks part, since our actual Lidar has already shown to work, the only risk may occur in the future when integrating all these submodules of the robots together. For instance, Jetson Nano may not be able to supply all the Lidar, camera etc. This can be solved decreasing the resolution of the data to be processed or slowing the rate of the Lidar data flowing to Jetson Nano.

5.2.2.3 Path Planning

Integrating target tracking and obstacle avoidance tasks into the motion of the robot is a critical stage of this project. Using the information provided by target tracking and obstacle detecting LIDAR modules, a mobile robot should able to construct a 2D map that demonstrates the relative position, velocity and acceleration of any object. Essentially, a mobile robot should be capable of planning its path through this map by analyzing the information delivered by aforementioned modules.

Up to CM1

The literature research we had done up until the Committee Meeting 1 showed us that, for path planning for obstacle avoidance and target tracking, "Potential Field Method" was quite a clearly defined and easy-to-implement method.

Yin - Yin [22] defines an attractive potential function with respect to the relative position, velocity and acceleration between the goal and the mobile robot, additionally a repulsive potential function considering aforementioned relative differences between the obstacle and the mobile robot in a dynamic environment (i.e. moving target and obstacles). This method benefits from a virtual force, calculated from the potential field of the ego robot, to plan the ego robot's motion with right positions and velocities to provide a similar motion trend with the goal and a contrary trend with surrounding obstacles.

There are 3 preconditions which need to be satisfied to make the algorithm work:

- Assumption 1: The mass m_r, position q and velocity v of the robot are known.
- Assumption 2: The position q_g, velocity v and acceleration a_g of the goal are known.
- Assumption 3: The obstacles are convex polygons whose shapes, positions q_{obs}, velocities v_{obs} and accelerations a_{obs} are known.

However, our research following CM1 guided us to understand the limitations of this method more evidently. Koren and Borenstein [23] show that considerable defects are found due to the inherent limitations of the method. Even though Potential Field Method seems to be elegant and

simplistic enough to attract many people, there are a couple of crucial complications that forestall its usefulness:

- Local minima may occur when a robot ends up stuck at a dead-end. Although there are heuristic solutions, they often cause non-optimal global paths.
- When there are closely spaced obstacles ahead of the robot, it may not find any passage due to the combined repulsive force which may be greater than the attractive force.
- Oscillations may occur when the robot is at a narrow passage or in the presence of obstacles.

As we became aware of these limitations, and the complexity of the solutions that attempt to solve these defects of the method, it seemed clear to us that another method should be proposed for path planning task.

Between CM1 and CM2

ROS provides us a great opportunity to construct a well-connected system. In order to exploit the advantages of using ROS, built-in Navigation package is determined to be utilized. This package requires information from odometry (data from the motion sensors), sensor streams such as laser scan, and a goal pose as its input in order to output velocity commands which are sent to the mobile base. One of the main perks of Navigation stack, it allows us to utilize ”move_base” package which is a major component of the Navigation stack.

Provided a goal in the 2D world, the ”move_base” package enables the mobile base to act in order to reach the goal. The implementation behind it is linking a global and local planner which require 2 costs map, global cost-map and local cost-map.

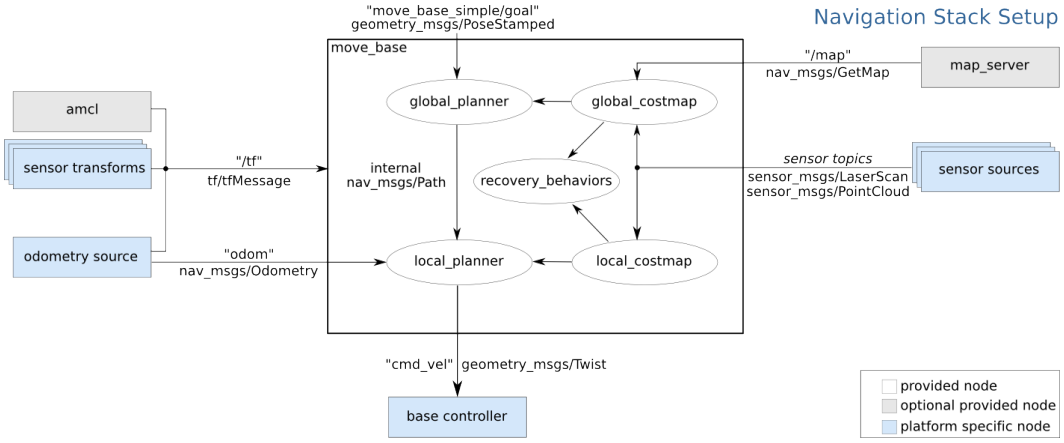

Figure 15: A high-level view of the move_base node and its interaction with other components [24]

In order to have costmaps, we implement ”costmap_2D” package, which constructs a 2D costmap by processing the sensor data such as laser scan or point cloud, which can be created from Lidar data. An example is given below:

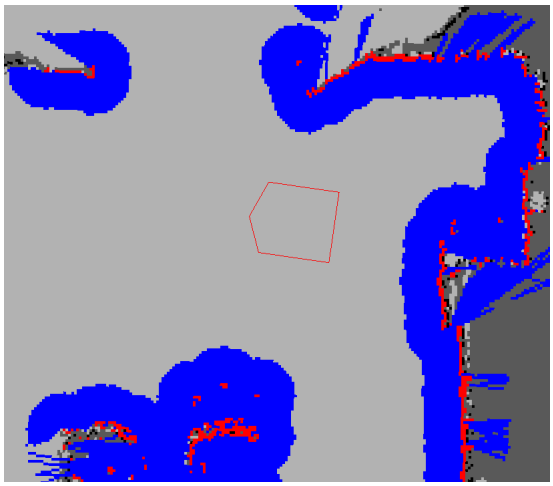

Figure 16: A costmap example [25]

In Figure 16, the obstacles are represented by the red cells, the blue cells are the regions where the center of the robot should never be in to prevent collisions, and the red polygon is the footprint of the robot. Essentially, to avoid collisions, the footprint should never intersect a red cell.

Our new proposed method which replaces the Potential Field Method is Timed Elastic Band (TEB), which in its essence, built-upon the aforementioned internal packages of ROS.

In their paper, Rösmann et al. [26] show that in the context of Model Predictive Control (MPC), a joint trajectory representation by merging control inputs, time intervals and states is built in order to plan time-optimal trajectories. In the control loop of TEB, MPC combines the planning with the state feedback, thus it approximates the time-optimal trajectory solution of analytical analysis in a few iterations during its runtime.

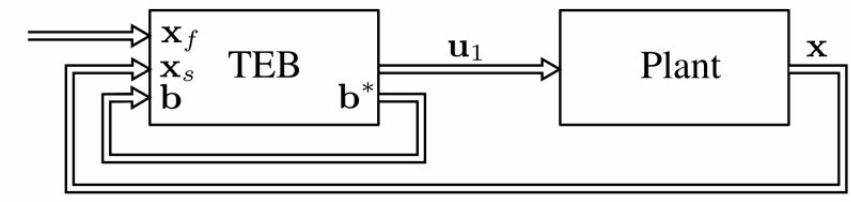

Figure 17: Closed-Loop TEB Control [26]

where x_f is the final state, b^* is the optimal trajectory, u_1 is the imminent control action of the planned sequence, x_s is the initial state and x is the state output of the plant.

An online optimal local planner is implemented for navigation in TEB package which is a plugin for the ROS navigation package. A global planner constructs an initial trajectory which is optimized during the runtime of TEB considering the following:

- Minimization of the trajectory execution time
- Maximization of the separation from obstacles

- Compliance with kinodynamic constraints

A multi-objective optimization problem is solve to obtain the optimal trajectory, and since the weights of the optimization problem is accesible by the user, we can determine the general behaviour of the trajectories and thus fine-tune it for customization purposed.

The optimal trajectory is efficiently obtained by solving a sparse scalarized multi-objective optimization problem. The user can provide weights to the optimization problem in order to specify the behavior in case of conflicting objectives.

In order to solve the local minima problem, an extension to TEB is developed. In paralel, the extension develops a subset of trajectories and the local planner selects the globally optimal one among the candidates.

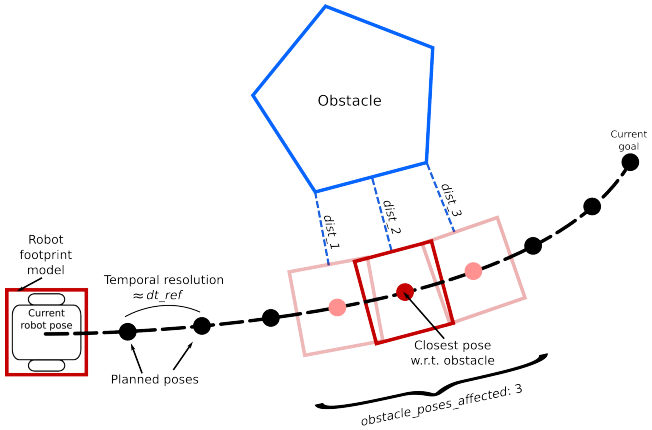

Figure 18: A snapshot of a TEB planning scenario [27]

The demo we made for CM2 presentation is given below in Figure 19. A 2-wheel differential-drive non-holonomic robot is used to test TEB method. As in the real-life scenario, we are not going to have a map of the area that the mobile robots are moving around, in this Gazebo simulation, an empty map, which means an empty global costmap, is given to the global planner. Therefore, the global planner does not possess any information about the objects existing in the simulation world. Considering Figure 15, the global planner of TEB takes the information of an empty map, the sensor information and the location of the goal, which is demonstrated with the green arrow on Figure 19, thus provides a global trajectory to the local planner.The local planner of TEB optimizes the global trajectory utilizing the sensor information which can consist of laser scan or point cloud data. The optimized local trajectories are feed to the mobile base as commands of navigation via ”cmd_vel” channel which is used to transfer velocity commands.

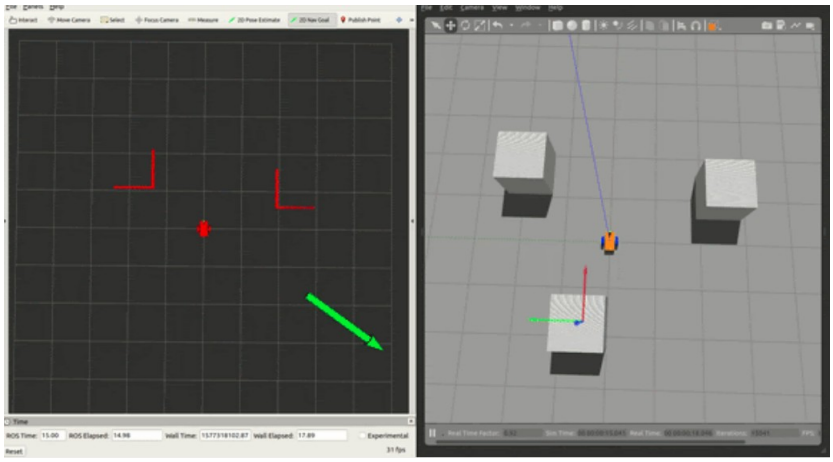

Figure 19: A snapshot of a TEB demo from CM2 Presentation

Future Plans

Our short-term plan is to implement the TEB method to the Dagu Wild-Thumper robot in Gazebo environment. Considering the fact that this robot model has the RPlidar model on top of it, we will tune the parameters of the TEB method according the rate of scan, maximum range and various other specifications of the Lidar model. Additionally, regarding the 4-wheel drive structure of the Dagu model on Gazebo, changes be needed to the maneuver commands directed by the TEB node.

As explained in the WBS section, we will be transferring these tested modules to the real setup at the beginning of February. The progress that has been made only covers the obtacle avoidance when a simple goal location is given to the TEB module. Therefore, the implementation to the real setup will require the integration of the Computer Vision module to provide goal locations in a 2D map that demonstrates the surrounding of the mobile robot.

Possible risks that might occur are as follows:

1. Non-smooth trajectories due to sub-optimal tuning of TEB algorithm
2. Inaccurate information about the location of the target when there are blocking objects in front of the Lidar, while the target is clearly detected by the Computer Vision (CV) module
3. Due to computational cost, the planning frequency (or the frequency of inputting a new target location) might be low, which means that at the end of an implemented path, the robot might be further away from the target, as the time gap between having new target location inputs is quite wide.

Possible solutions for the risks are given below, respectively:

1. Some parameters might be tuned to provide high tolerance, which will help TEB module to output smooth trajectories, whereas the trade-off is passing too close to surrounding obstacles and/or completing the path at a point which might be further away from the target location.

2. Although the angle of the target is known as CV module is working, the last distance information might be given again at least to pass the obstacle with local planner's avoidance mechanism.

3. The number of parameters to be considered may be decreased to help the computation process, which cause a trade-off of accurate tracking or avoidance performance.

5.2.3 Simulations and Real Life Testing

5.2.3.1 Gazebo

Gazebo is a platform that works as simulation atmosphere for the Robotic Operating System(ROS). ROS is for the back-end development area which the control algorithms and Artificial Intelligence techniques are implemented. For the testings of real-world like environment, Gazebo appears to be decent alternative. Implemented algorithms are tested upon 3D modeled robots and used peripherals. Sample image for a gazebo simulation can be demonstrated as such:

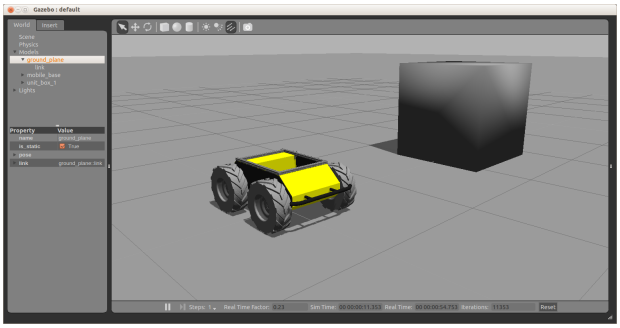

Figure 20: Gazebo Simulation Software [28]

Up to CM1

All the member who have been working on ROS and Gazebo completed setup process and launched empty simulations on the software. The ROS version for the all member are selected as the current version of ROS which was Melodic. Using the documentation provided for the installation and initial launch methods are studied.

Gazebo simulations are done on a sample model called Husky Robot which is the robot provided in the Figure 20. Husky online tutorial is completed from SmartLab website[28] by Arda Yüksel. The online documentation regarding the usage of peripherals such as Action Camera and LIDAR is analyzed. As the robot model was decided in the Land Robot Chassis sub-task of Mechanical Setup, pre-built DAGU Wild Thumper models for Gazebo are searched. Up to CM2,a working Wild Thumper Model usge instead of Husky Robot was aimed.

Between CM1 and CM2

During the period between CM1 and CM2, various robot models are tested such as Turtlebot and online tutorials in the fields of navigation, SLAM and robot control is done through usage of Turtlebot and a designed 2 wheeled robot model. In order generate this model online tutorial that covers the basics of robot design for Gazebo are studied by Arda Yüksel, Mert Acar, Bilgehan Başpınar and Cevahir Köprülü from MooreRobots[29]. Since the fundamentals that the tutorial covers are also related to the development of LIDAR, Path Planning and Camera Output Integration over ROS, all the member associated with the relative tasks completed this set of tutorials successfully. Using their knowledge, Arda Yüksel and Cevahir Köprülü generated a sample 2 wheeled robot to comprehend the robot design in Gazebo. The model simulations can be seen from the image below as:

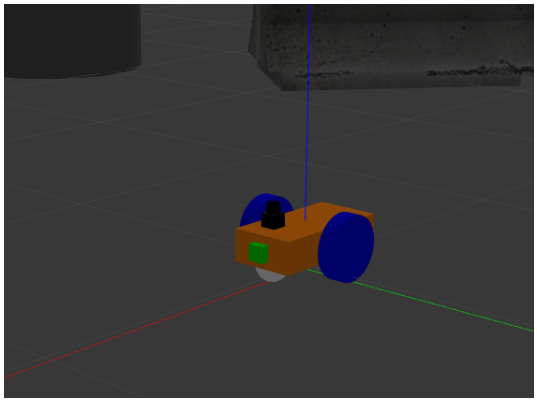

Figure 21: 2 Wheeled Robot with Hokuyo Laser[29]

By using this model, movement and output channels of simulations are learned. Furthermore, by experimenting in this setup the concept such as Navigation, Odometry and Teleop operations which are fundamental for ROS are examined. In this model, aside from the 2 wheeled robot model, one black Hokuyo Laser, which is built-in model for SLAM is generated to study the basics of LIDAR integration. In addition to this model, Cevahir Köprülü found a DAGU Wild Thumper 4WD model that works in previous versions of ROS.

The URDF file, which contains the information related to DAGU model and its components such as wheels and peripherals, is analyzed and adjustments are done in order to generate a launch file compatible with ROS Melodic. After these adjustments, using the world launch file designed for the 2 wheeled setup, DAGU Wild Thumper is generated as such on Gazebo Environment:

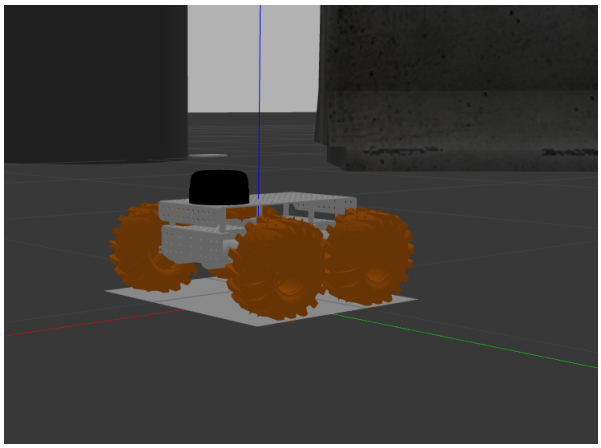

Figure 22: DAGU Wild Thumper 4WD on Gazebo

Using the established robot model, Arda Yüksel generated the ROS essential tools which are publishers and subscribers. ROS publishers return the outcome of the peripherals such as LIDAR and subscribers send commands to robot and their components such as motor drivers defined in URDF file. The nodes of the current setup is given as:

```
lordspoiler@arda-GT60:~/mybot_ws$ rostopic list
/clock
/cmd_vel_out
/diff_drive_controller/cmd_vel
/diff_drive_controller/odom
/diff_drive_controller/parameter_descriptions
/diff_drive_controller/parameter_updates
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
/imu
/joint_states
/odom
/rosout
/rosout_agg
/scan
/tf
```

Figure 23: Nodes of the Simulation

In the setup /scan and /diff_driver/cmdare crucial. In their simulation, Bilgehan uses /scan data to visualize the LIDAR output as in the Point Cloud Format. The cmdis necessary to move the robot. These input and output channels are tested for the usage in the Integration ROS with YOLO and LIDAR by Arda YÜKSEL. After confirming their validity, the simulation files are used in other sections.

Future Plans

Difficulties that are observed for the implementation of the /odom sub-node. It is essential for the Path Planning and due lack of working /odom publisher the testing procedure is done on Turtlebot instead of the implemented DAGU Model. In future, all the sub-notes will be working properly and thus the algorithms will be tested in the generated Gazebo environment.

In addition to that, the current robot model is designed as DAGU Wild Thumper 4WD while the selected model is DAGU Wild Thumper 6WD. Since the robot is generated using the rendered robot found by Cevahir Köprülü, alterations in design requires basic knowledge of AutoCAD software. The current design includes the integration of model and URDF files. It can be challenging to alter the model according to the real world setup. However, the number of wheels does not alter the simulations and testing procedure due to the number of controllers will be used for real world setup. Currently, two controllers are designed which are for controlling the left and right wheels. In the simulations, /diff_driver_controller works in the same respect. Therefore, the difficulties related to the differences between real-world and simulation robot will be solved by manipulating the drivers, if 6WD robot cannot be generated.

5.2.3.2 Control with ROS

Up to CM1

The need for controllers and possible motions that has to be performed by robot were discussed.

Between CM1 and CM2

Designing a PID (proportional–integral–derivative) controller using ROS to control the robot is planned. As selected motor drivers can be controlled with Pulse Width Modulation (PWM) technique, the controller can control the robot by outputting PWM signals and send command to motor drivers.

Future Plans

The PID controller will be designed on ROS. This controller will control all of the motions of the robot, taking the input from the computer vision system in the robot and by running control algorithms, it will output control commands. This commands will be taken as input by motor drivers and the motors and wheel of the robot will move according to these control commands, hence robot will have an algorithmic motion. During the controlling process of the simulated robot, we may encounter issues due to the usage of Wild Thumper 4WD in Gazebo. To overcome this issue, the concept of Skid Skewer Driver control which allows multiple wheel setups to work properly both in simulations and real world testing can be integrated if necessary.

5.2.3.3 Modelling Lidar

One of our project's major parts is implementing Lidar since the autonomy of our robots is based on tracking and obstacle avoidance, therefore path planning. As mentioned, for the path planning algorithm we used to plan to use Potential Field Method and we have switched it to Timed Elastic Band Method, which requires a data flow in the form (distance, angle) and in the structure of sensormsgs/LaserScan Message, which is a class in the Lidar ROS library.

Up to CM1

In this period, since our simulation skills were not good enough, we could only manage to simulate a Lidar in Gazebo, but not the one that we planned to use, that is RPLIDAR A2M8. Still, at the time it was important to be able to simulate a Lidar in the sense of learning generating simulations that are close to our case. Therefore, the following simulation and visualization was made using a 180-degree Lidar:

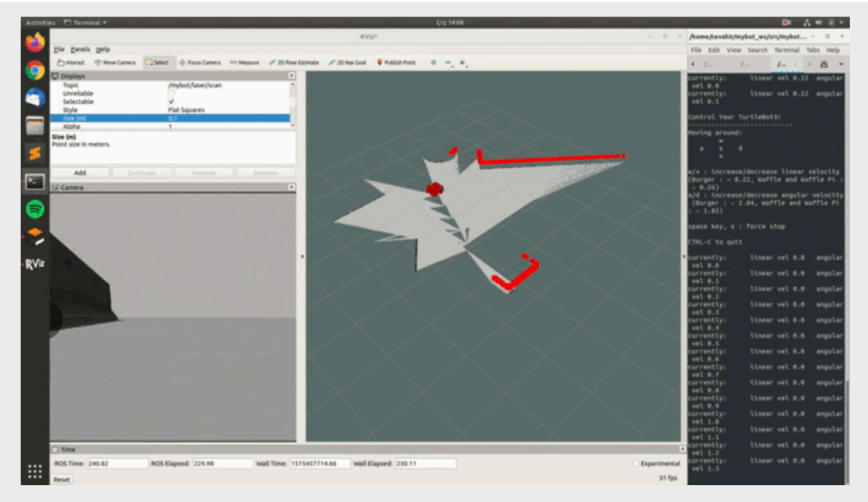

Figure 24: Lidar simulation result before CM I using a simpler Lidar Gazebo model

Between CM1 and CM2

In the progress meeting, when performing a simple simulation a simple laser model was used as the initial step. After that, this simple laser model was successfully replaced by the real Gazebo model of RPLIDAR A2M8 and using and modifying the ROS node written for RPLIDAR [30], a subscriber node was written in order to be able to listen the data coming from the "/scan" channel, in the structure of sensormsgs/LaserScan Message, published by the Lidar.

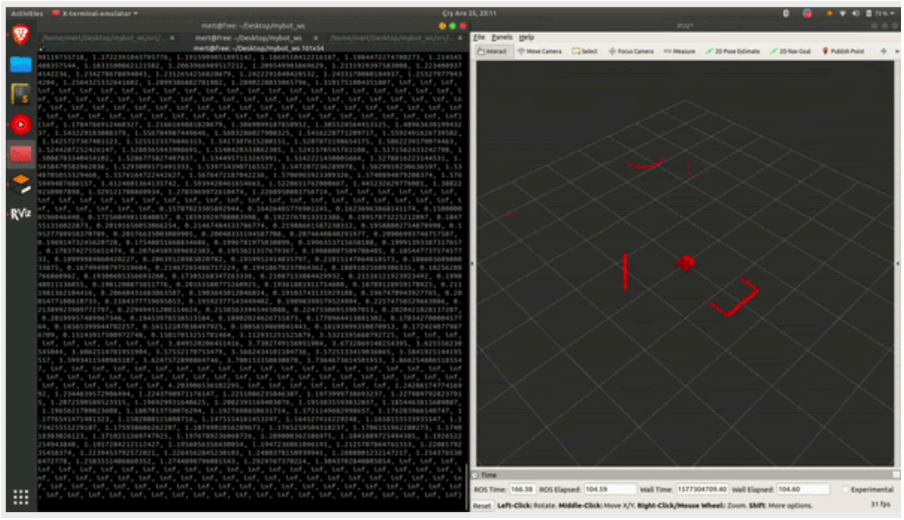

Figure 25: The RPLIDAR data flowing on the left, RViz visualization on the right which shows the obstacles in the simulation world

As it can be seen from the figure above, we are now able to get the Lidar data in the form we need for the path planning algorithm and we can visualize this data on RViz, using the real model for RPLIDAR A2M8 this time as opposed to the case before the Committee Meeting I.

Future Plans

Similar to the real Lidar case mentioned before, from now on, we need to work on integrating the simulation Lidar data into the path planning algorithm once the Timed Elastic Band algorithm understood in detail.

As for the possible risks part, our overall simulation is almost done except for the incompleteness of path planning algorithm. Therefore, the only problem may occur when adding path planning algorithm. In order to solve that, the Lidar data can be modified according to the algorithm in terms of rate and type.

All the member who have been working on ROS and Gazebo completed setup process and launched sample programs given in the online tutorials. Furthermore, previously designed models for the robot that will be used throughout the project have been found. Members of the group now is in the learning process and they try to generate various environments that can be used for modeling the military operation areas. Since the robot design was not determined till the later stages, instead of focusing on the robot the world building was the focus.

In order to perform better in real world applications, peripherals such as Action Camera and Lidar will be implemented upon the model and by the Computer Vision sub team works on the object detection, individuals who work on the ROS/Gazebo environment will focus on the object detection. As object detection requires, comprehension of the SLAM algorithm, by completing the simulations improvements in the other fields will be achieved.

One of the future development areas, swarm intelligence will be tested firstly on Gazebo and

ROS, since it is easier to generate robots in simulations. By testing multiple algorithms that are covered in the next part, possible developments will be carried out.

The limitations of the simulations are quite clear. As these programs are designed, the world building prepares models for the general cases. For more specification and testing, real world testing is also essential. Creating a real world like environment and designing robot for simulations are also difficult. In order to solve these issues, further work distribution within ROS/Gazebo software are needed for specializing in multiple areas.

5.2.4 Swarm Intelligence

Swarm Intelligence is one of the key elements of the project that differs itself from the other contemporary projects in the field. For multiple and varying number of robots in the tasks, it is crucial to have a general structure for tasks that is mentioned in previous subsections. In order to generate a working model that can deliberately organize work distribution with multiple robots and control their fundamental interactions with humans assigned for the operations, swarm intelligence will be implemented for the project.

Up to CM1

As Brambilla [31] defines it, swarm intelligence is "an approach to collective robotics that takes inspiration from the self-organized behaviors of social animals". This concept mainly developed by imitating insect swarms on the basis of their communication within swarm and work distribution. Its applications on robotics is contemporary research topic and throughout this project, it is expected to be implemented for assisting as robot group for military operations. As it can be seen from the Work Breakdown Structure and Gantt Chart, the application procedure has not been initiated for this task and thus it is in early development stage.

During research process for the swarm intelligence, the properties of possible robot swarms and specifications for the design procedure have been established. It is expected for robot swarms to fulfill the following requirements[32]:

- **Autonomous:** Robots should be able to decide on the actions given circumstances without human control
- **Ability to comprehend environment:** Robots should be able to respond to nature through data processing and decision making algorithms
- **Identical robots:** Robots should have similar equipment in order to take measures in case of possible losses in swarm
- **Local interaction abilities:** Robots should have a local network to share information
- **Cooperation abilities:** Robots should be able to perform operations cooperating with each other

For the development of the robot swarm with multiple robots, these properties will be taken into consideration. Since the robots must act on without any human control autonomous property is a must regardless the usage of swarm intelligence. In addition to that ability to comprehend environment is also a crucial factor which will be implemented before the development of swarm intelligence through SLAM and Computer Vision techniques.

Robots are expected to switch roles as one can be master or slave for different tasks, therefore identical robots will make it easier to implement varying work distribution mechanic. Local interaction abilities are planned to perform via networks which will be covered later on. Cooperation abilities is one of the important aspects which will allow exploration tasks to perform more efficiently.

One of the most important reasons that swarm intelligence is essential for the future of the project is that its viability in behavior predictions for multi robot systems. For this task following algorithms and techniques will be analyzed and simulated:

- **Probabilistic finite state machine design(PFSM):** According to input data, robots alter their behavior in a predefined probabilistic finite state machine model.
- **Reinforcement learning:** Through trial and error in simulations, swarm learn to tackle sample situations.
- **Evolutionary robotics:** Effectiveness of the behavior is tested using evolutionary computation

Of all the algorithms mentioned above, PFSM is crucial since it appears in most of the researches on the swarm intelligence as Brambilla[31] describes. It allows robots to select appropriate behaviours with the probabilities attained during the simulation process. For the generation of the probabilities reinforcement learning and evolutionary robotics are used.

Future Plans

During the simulations, ROS and Gazebo will used simultaneously to evaluate the performance of the behavior selection. One other key feature of the swarm intelligence is its ability to communicate with other members of swarm. For this task, **Mobile Ad-Hoc Networks(MANET)** are expected to implemented in later stages of development. MANET is a communication protocol that allows agent to enter and leave at any time. Due to possibility of malfunctioning of the robots, its viability to be established in those circumstances are vital for the operations.

Swarm intelligence has potential risks that have to be tackled in the application and simulation process. As it was mentioned, the communication within swarm is volatile and due to absence of rooter it is not as secure as the other systems.[33] In order to tackle this issue, secure communications will be searched. In addition to that, implementing the aforementioned algorithms will take considerable amount of time. To resolve this issue, viable options will be analyzed through contemporary researches and viable options will be tested in simulations. Since the project will work with smaller swarms, simplifications for the learning procedure can be done.

6 Final Demo

In the final demo, two mobile robots will act in coordination to follow a predetermined human leader. The mobile robot chassis is based on Dagu Wild Thumper 6WD model. On top of the chassis a set of camera, Lidar, motor drivers, LiPo batteries, beacon and Jetson Nano will be mounted. The venue for the final demo will be Marmara Cafeteria or EB building due to including rigid obstacles and concrete or non-level grass terrain on a sloped surface.

One of the robots will be the master controlling the formation and the other one will be the slave taking orders from the master robot. Master robot is going to track the marked person and give relevant orders to the slave robot while avoiding static (poles, benches etc.) and dynamic obstacles (people walking around). The person to be followed was initially planned to be marked with a black yellow striped belt where it may be switched to a reflection belt according to the feedback taken in Committee Meeting II. During the demo, master and slave robots can alter roles in case of the difficulties observed in the mission. An example of this can be the master robot being surrounded or held captive by obstacles/adversarial humans. Thus, the task of following the human leader will be carried by the prior slave.

As a matter of fact, this demo is appropriate for both Industrial Project mentors and other individuals who are not very familiar with the scope of the project. That's because, the demo is based on demonstrating the functionality and the practical purpose of the project to non-experts as well.

7 Equipment List

Equipment	Cost (in TL)	Obtained
Nvidia Jetson Nano (for First Robot)	1115	by purchase
Nvidia Jetson Nano (for Second Robot)	No Cost	from ROKETSAN
Dagu Wild Thumper 6WD All-Terrain Chassis (x2)	No Cost	from ROKETSAN
Eken H9R 4K Action Camera (x2)	756	by purchase
RPLIDAR A2 M8 (x2)	No Cost	from ROKETSAN
Beacon (x2)	No Cost	from ROKETSAN
LiPo 7.4V 5200 mAh 2S (x4)	1072	by purchase
BTS7960B 40A Single Channel Motor Driver (x4)	180	by purchase
Xiaomi 10000 mAh Power Bank (x2)	200	by purchase
Total Cost	3323	

Table 1: BoM of the Project

Single robot consists of one Nvidia Jetson Nano, one Dagu Wild Thumper 6WD All-Terrain Chassis, one Eken H9R 4K Action Camera, one RPLIDAR A2 M8, two LiPo 7.4V 5200 mAh 2S, two BTS7960B 40A Single Channel Motor Drivers and one Xiaomi 10000 mAh Power Bank. The Bill of Material is given in Table 1 consists of all the materials for two robots. Currently, our project

objective is to build at least two robots in order to construct the swarm behaviour. For this swarm purpose, ROKETSAN has provided us two beacons.

References

[1] Wikipedia.org. *ROKETSAN*. URL: https://en.wikipedia.org/wiki/ROKETSAN.

[2] Angelo Nikko Catapang and Manuel Ramos.
"Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle".
In: *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)* (2016). DOI: 10.1109/iccsce.2016.7893614.

[3] Pololu Robotics and Electronics.
Dagu Wild Thumper 6WD All-Terrain Chassis, Silver, 34:1. URL: https://www.tme.eu/Document/3042ea29739d84f52c2c511fe0a15337/DAGU-RS003B75.pdf.

[4] Nvidia. *Jetson Nano*. URL: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/.

[5] NATO. *Autonomous Systems: Issues for Defence Policymakers*.
URL: https://www.act.nato.int/images/stories/media/capdev/capdev_02.pdf.

[6] Michael Barnes et al.
"Designing for Humans in Autonomous Systems: Military Applications". In: (Mar. 2014).
DOI: 10.13140/2.1.3500.7688.

[7] ROKETSAN. *ROKETSAN: About Us*. URL: http://www.roketsan.com.tr/en?current_ajax_id=9868&current_ajax_page_type=page.

[8] Oli Moser. *Introduction to Computer Vision With OpenCV and Python*. URL: https://dzone.com/articles/introduction-to-computer-vision-with-opencv-and-py.

[9] Brian Barrett. *Inside the Olympics Opening Ceremony World-Record Drone Show*.
URL: https://www.wired.com/story/olympics-opening-ceremony-drone-show/.

[10] Mengmeng Wang et al.
"Real-time 3D Human Tracking for Mobile Robots with Multisensors". In: (Mar. 2017).
DOI: 10.13140/2.1.3500.7688.

[11] *2040.1 - Standard for Connected, Automated and Intelligent Vehicles: Taxonomy and Definitions*. URL: https://standards.ieee.org/project/2040_1.html.

[12] *2040.2 - Standard for Connected, Automated and Intelligent Vehicles: Testing and Verification*. URL: https://standards.ieee.org/project/2040_2.html.

[13] Ltd. Xiaomi Communications Co. *10000mAh Mi Power Bank 2S User Manual*.
URL: https://i01.appmifile.com/webfile/globalimg/UK/Manual/10000mAh_Mi_Power_Bank_2S.pdf.

[14] NovalithIC. *BTS 7960 High Current PN Half Bridge Motor Driver*.
URL: http://www.robotpower.com/downloads/BTS7960_v1.1_2004-12-07.pdf.

[15] Brian Schneider. *A Guide to Understanding LiPo Batteries*.
URL: https://rogershobbycenter.com/lipoguide.

[16] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement".
In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767.
URL: http://arxiv.org/abs/1804.02767.

[17] Joseph Redmon. URL: https://pjreddie.com/darknet/tiny-darknet/.

[18] Yimei Kang, Wandong Xie, and Bin Hu.
"A scale adaptive mean-shift tracking algorithm for robot vision".
In: *Advances in Mechanical Engineering* 5 (2013), p. 601612.

[19] SLAMTEC.
RPLIDAR A2 Low Cost 360 Degree Laser Range Scanner Introduction and Datasheet.
URL: https://cdn.sparkfun.com/assets/e/a/f/9/8%20LD208_SLAMTEC_rplidar_datasheet_A2M8_v1.0_en.pdf.

[20] Yan Peng et al.
"The obstacle detection and obstacle avoidance algorithm based on 2-D lidar".
In: *2015 IEEE International Conference on Information and Automation* (2015).
DOI: 10.1109/icinfa.2015.7279550.

[21] Angelo Nikko Catapang and Manuel Ramos.
"Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle".
In: *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)* (2016). DOI: 10.1109/iccsce.2016.7893614.

[22] Lu Yin and Yixin Yin. "An improved potential field method for mobile robot path planning in dynamic environments".
In: *2008 7th World Congress on Intelligent Control and Automation* (2008).
DOI: 10.1109/wcica.2008.4593709.

[23] Y. Koren and J. Borenstein.
"Potential field methods and their inherent limitations for mobile robot navigation".
In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation* ().
DOI: 10.1109/robot.1991.131810.

[24] *ROSWiki.* URL: http://wiki.ros.org/move_base?distro=melodic.

[25] *ROSWiki.* URL: http://wiki.ros.org/costmap_2d?distro=melodic.

[26] Christoph Rosmann, Frank Hoffmann, and Torsten Bertram.
"Timed-Elastic-Bands for time-optimal point-to-point nonlinear model predictive control".
In: *2015 European Control Conference (ECC)* (2015). DOI: 10.1109/ecc.2015.7331052.

[27] *ROSWiki.* URL: http://wiki.ros.org/teb_local_planner?distro=melodic.

[28] SMARTlab-Purdue. *SMARTlab-Purdue/ros-tutorial-gazebo-simulation.*
URL: https://github.com/SMARTlab-Purdue/ros-tutorial-gazebo-simulation/wiki/Sec.-2:-Driving-the-Husky-robot-in-Gazebo.

[29] Rpw. URL: http://moorerobots.com/blog.

[30] ROS.org. *Rplidar Ros Package Summary.* URL: http://wiki.ros.org/rplidar_ros.

[31] Manuele Brambilla et al. *Swarm robotics: a review from the swarm engineering perspective.* Jan. 2013. URL: https://link.springer.com/article/10.1007/s11721-012-0075-2.

[32] Erol Şahin et al. *Swarm Robotics.* Jan. 1970. URL: https://link.springer.com/chapter/10.1007/978-3-540-74089-6_3.

[33] Muddassar Farooq and Gianni A. Di Caro. "Routing Protocols for Next-Generation Networks Inspired by Collective Behaviors of Insect Societies: An Overview". In: *Natural Computing Series Swarm Intelligence* (), pp. 101–160. DOI: 10.1007/978-3-540-74089-6_4.