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

EEE 493 - Industrial Design Project
Committee Meeting 1
Report

Arda Yüksel
Bilgehan Başpınar
Cevahir Köprülü
Mert Acar
Oğuz Altan

Electrical and Electronics Engineering
İhsan Doğramacı Bilkent University
Ankara, Turkey
November 2019

Contents

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 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 4WD All-Terrain Chassis 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 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 that aims to decrease the aforementioned costs and risks. One of the solutions is developing autonomous mobile robots 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.

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 and drone swarms moving in pre-determined trajectory to accomplish various tasks. 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. The 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 and a sample frequency of 4 kHz.

6. We also expect the simulation to model the real word terrain conditions accurately.

3.2 Non-Functional Requirements/Constraints

1. We are planning to make one of the robots for 3664 TL (excluded Lidar coming from the company) and we can make 2 or 3 robots as far as we talked with the company. 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. We do not have a size or weight constraint imposed by the company, yet mechanical units are 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. The robot that we are going to purchase is about 2 kg and we are going to mount Jetson Nano, battery, camera, Gimbal and Lidar, which will be 3 kg approximately.

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

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. In addition, we are planning to use a Gimbal in order to adapt well with stabilization measures for the image processing operations on rugged terrains.

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 in terms of automation.

4 Methods and Implementation Details

4.1 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.

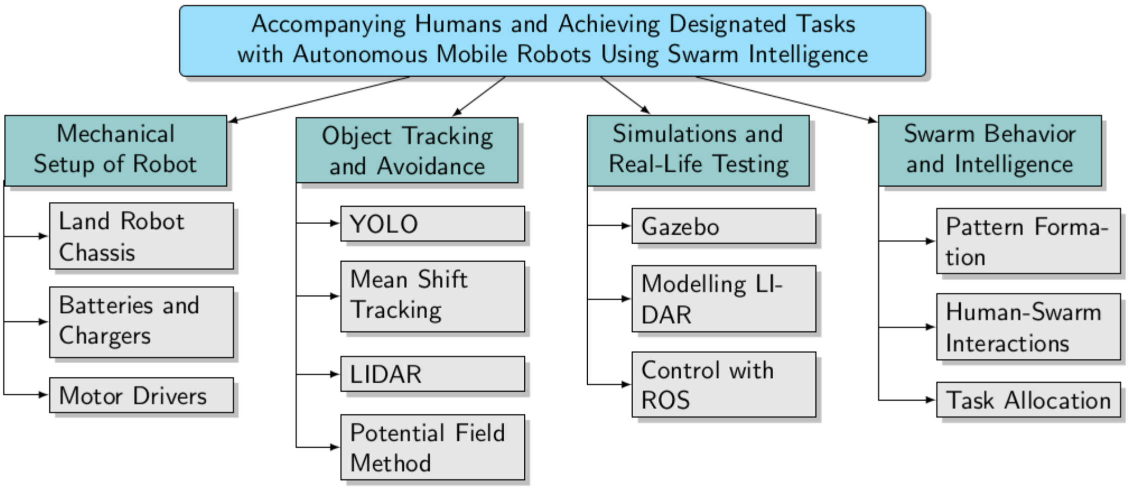

Figure 1: Work Breakdown Structure of The Project

4.1.1 Mechanical Setup of Robot

Mechanical setup of robot comprises three sub-tasks: Land Robot Chassis Assembly, Batteries and Chargers, Motor Drivers Selection and Testing. For land robot chassis the main criterion is the robot's ability to handle the terrain conditions as the main scenario demands. Therefore an aluminium construction robot chassis with independently moving wheels is chosen. The model is Wild Thumper 4WD manufactured by Dagu. It has a gear ratio of 75:1 and can sustain 11kg-cm of torque. It is expected to carry around 7 kg of payload. The motors draw 420 mA of current when the robot is not loaded with payload. This number goes up to 6.6 A when under full load. During initial movement and direction changes the motors draw a momentary current of 18 A. Therefore the second sub-task, Batteries and Chargers, need to address these constraints. The choice for the land unit's power need is two Lithium Polymer batteries. LiPo batteries are chosen because of their ability to sustain necessary currents and their capacity for operation time. The batteries are to be bought and tested before deployment. Once the first two sub-tasks are achieved, the motor drivers with appropriate current and voltage regulators are going to be bought and used for control purposes. This milestone is under Oğuz Altan's responsibility and given one month for the whole testing and integrating process. This plan can be seen in the Gannt Chart provided in Figure 2.

4.1.2 Object Tracking and Avoidance

Object tracking and obstacle avoidance consist four sub-task underneath: You Only Look Once (YOLO) network setup, mean shift tracking, LIDAR setup and potential field method. 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. The second sub-task is to develop and integrate the mean shift tracking algorithm that is used to track the detected objects. This algorithm only uses the camera feed information and the object tracking output from YOLO. However, the land unit also needs to be aware of the distance between itself and the objects around its environment. Therefore, a LIDAR is going to be integrated to map out the environment as a point cloud in two dimensional space. Using the information from the LIDAR, the land robot needs to be able to avoid objects on its path while tracking the target. In order to achieve this task potential field method will be integrated to the control mechanism which generates a potential field on which the objects constitute a negative force and the robot takes the most positive route. For the first two sub-tasks Mert Acar will be responsible with a time expectancy of two months. Bilgehan Başpınar will work on LIDAR integration for an expected duration of one month. Then the potential field method will be integrated by Cevahir Köprülü within one month. The exact durations can be seen in the Gannt Chart provided in Figure 2. The success criteria for this milestone is achieving an image processing performance of 12 frames per second and implementing a functioning system efficiently so that the performance of tracking does not alter too much.

4.1.3 Simulations and Real-Life Testing

The simulation and real-life testing of the project will be broken down to three sub-tasks. The first one is to setup necessary development environments like Gazebo and Robot Operating System (ROS) as well as to create an accurate model of the land robot. Developing the algorithms mentioned in Object Tracking and Avoidance milestone an accurate modelling of the LIDAR that we are going to use will be needed. After the LIDAR model is done the land unit will be controlled with ROS integrating the algorithms. Necessary simulations are going to be evaluated in compliance with the performance criteria of modelling the world accurately and being able to control the robot in accordance with the tracking and object avoidance algorithms. The ROS integration will be done by Arda Yüksel, Bilgehan Başpınar and Oğuz Altan. The expected duration for this milestone is extended over six months.

4.1.4 Swarm Behaviour and Intelligence

The last milestone of the project is achieving swarm behaviour with multiple robots. This milestone is divided into three sub-tasks. The first one is pattern formation. Each unit in the swarm needs to be aware of the other units around it in order to proceed with coordination. Then the milestone progress goes on with task allocation where the intelligent decision of assigning tasks to different units in the swarm is aimed to be achieved. Finally the project wraps with human-swarm interaction development to meet the project demand that swarm needs to accompany humans in a coordinated way in doing designated jobs. The swarm behavior will be designed and implemented by Arda Yüksel and Mert Acar. The second semester in the project will be assigned just for this goal. The criteria of success for this milestone is being able to establish master-slave behaviour and coordinated formation and movement with at least two mobile units.

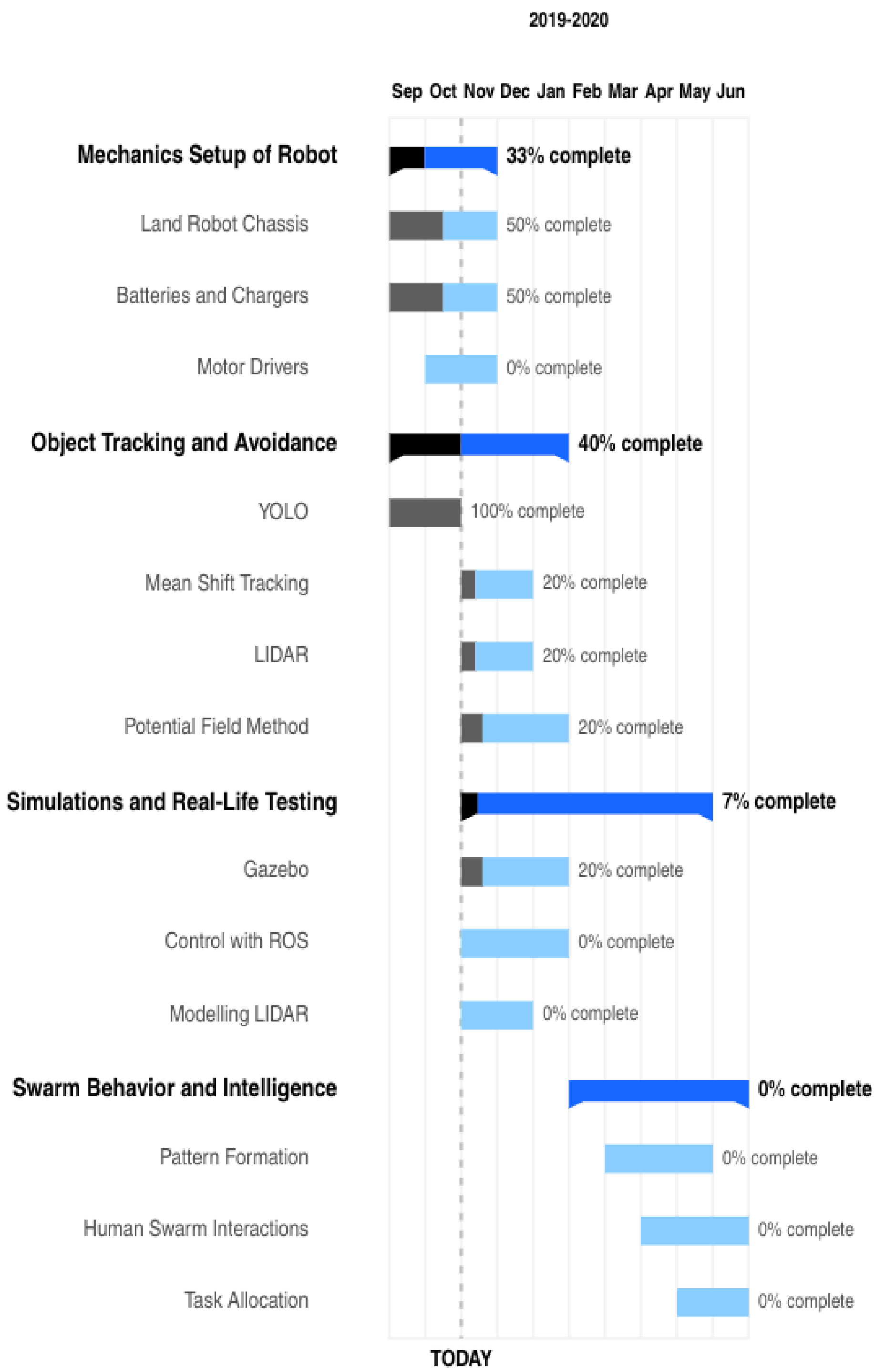

Figure 2: Gannt Chart for the Project

4.2 Methods and Progress

4.2.1 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping(SLAM) is an algorithm implemented via the usage of Lidar data to categorize the obstacles and generated indoor and outdoor maps according to the identified objects. In this project, rather than mapping aspect, obstacle identification will be its main use. For this task, **RPLIDAR A2 M8 device** will be used. This device, as it will be explained later on, has not been purchased yet. Therefore, its ROS/Gazebo model is used as a basis for the explanations.

The LIDAR device, which is provided with its own ROS package, works as the following:

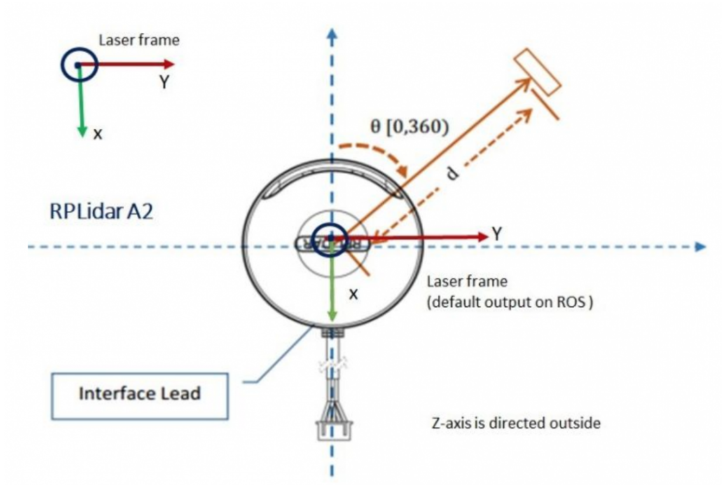

Figure 3: RPLIDAR A2 M8 Structure

From the device the angle and distance to each object can be acquired. This data is used for mapping the area in SLAM algorithm. In the project, the distance and angle will be used for control algorithms as well since it can be used for path finding and stabilizing the robot system.

Up to this point, the general structure of the LIDAR has been studied and its inputs and outputs are analyzed for further use. Its simulations will be done through ROS/Gazebo. Due to existence of the pre-built model for the LIDAR and its SDK in ROS, simulations are expected to be completed easily. For object detection procedure following algorithm is selected to be implemented during the simulations:

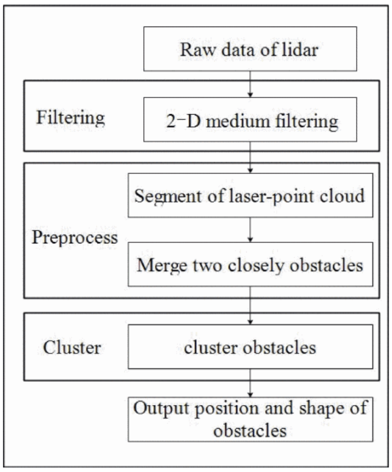

Figure 4: Obstacle Detection Algorithm [5]

Filtering aspect is completed via usage of **2-D Median Filters** and the raw data of LIDAR is its input. The median filter allows removal of the noises and noise peaks and eases pre-processing of the data. In the software implementation, a dynamically allocated array takes the values of the current distance point and its two neighbors. The array is sorted using the **QuickSort** method and the median replaces the value of the current distance point.

For the pre-processing of the data, each point or sample in the distance map has a relative position to the LIDAR system computed from the distance and the angle of the sample. Thus, each point or sample has a coordinate. If the distance exceeded a given maximum, the points were segmented and placed into different blocks. Similarly, if the distance was less than the given maximum, the points were merged into one block.

In the final stage which is the clustering, each block after pre-processing was subjected to shape association.[2] It has three shaping rules: circle, line and rectangle for simplicity. Each shape corresponded to a classification defined for the obstacle detection algorithm which results in simplification the complex LIDAR point cloud.

4.2.2 Target Tracking

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 proposin g 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:

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

An example of bounding box prediction is given below in Figure 5

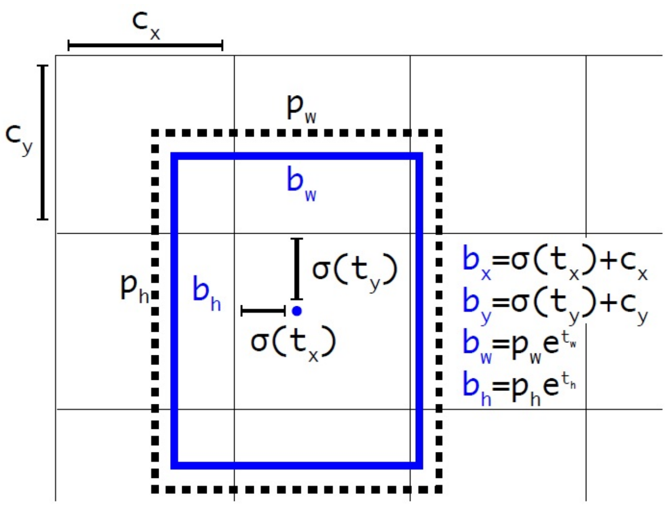

Figure 5: Bounding boxes with dimension priors and location predictions [6]

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 6:

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 6: Tiny-YOLO Architecture tiny$_{y}olo$

Our first implementation was compiling YOLO on Jetson Nano with CUDA and OPENCV, and then giving the network a generic image of Figure 7:

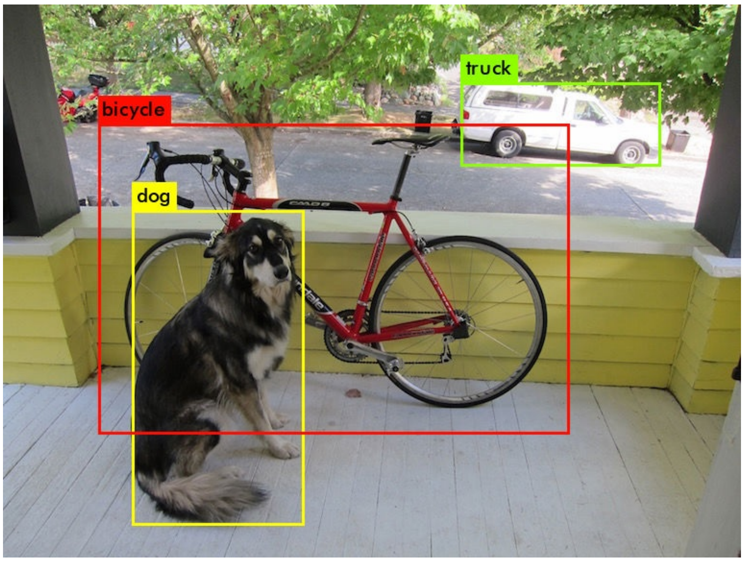

Figure 7: Tiny-YOLO Testing

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 8 from the tested live-feed:

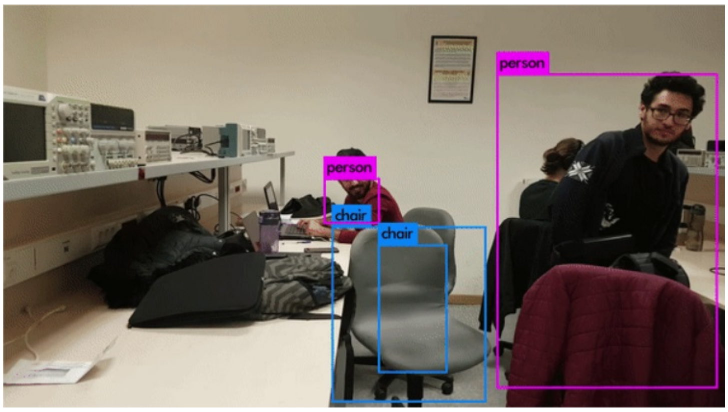

Figure 8: Tiny-YOLO Real-Time Test

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" will be employed. Mean shift is a semiautomatic tracking algorithm that is based on an iterative scheme.[4] The main idea behind mean shift is to maximize correlation between RGB color histogram of the original target in two consecutive frames.

The employed kernel function $k_h(|(x-x_i)/h|^2)$ is symmetrically centered at point x to make the algorithm indifferent to rotation or shift. The points x are drawn from $S = \{x_i\}_{i=1,\ldots,n}$ which are independent and identically distributed random variables with an unknown density function *f(x)*

$$f(x) = \frac{1}{MxN} \sum_{i=1}^{MxN} k_h \left(\left| \frac{x - x_i}{h} \right|^2 \right) \tag{6}$$

where the size of the object to be tracked is *MxN* pixels and h is the bandwidth with *h>0.* Here h is used to normalize the sample weights and also gives an idea for the size and shape of the target object. [4] Target candidates chosen one at a time around the target point centre mapped from the previous frame. Bhattacharyya coefficients are then applied to find the similarity between candidate in the current frame and target in previous frame. [4] The Bhattacharyya coefficient is defined as follows:

$$d(y) = \sqrt{1 - \rho(y)} \tag{7}$$

where,

$$\rho(y) \equiv \rho(p(y), q) = \sum_{b=1}^{B} \sqrt{p_b(y) q_b} \tag{8}$$

Here q represents the target model, $\rho(y)$ represents the candidate model and B represents the bins that are used to calculate the model. [4]

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. [4]. Figure 9 shows a Bhattacharyya coefficient surface for the left hand side frame. Here the target is the solid square. Assuming the kernel size is the same as the target object the bounding blue box around the square shows the kernel.

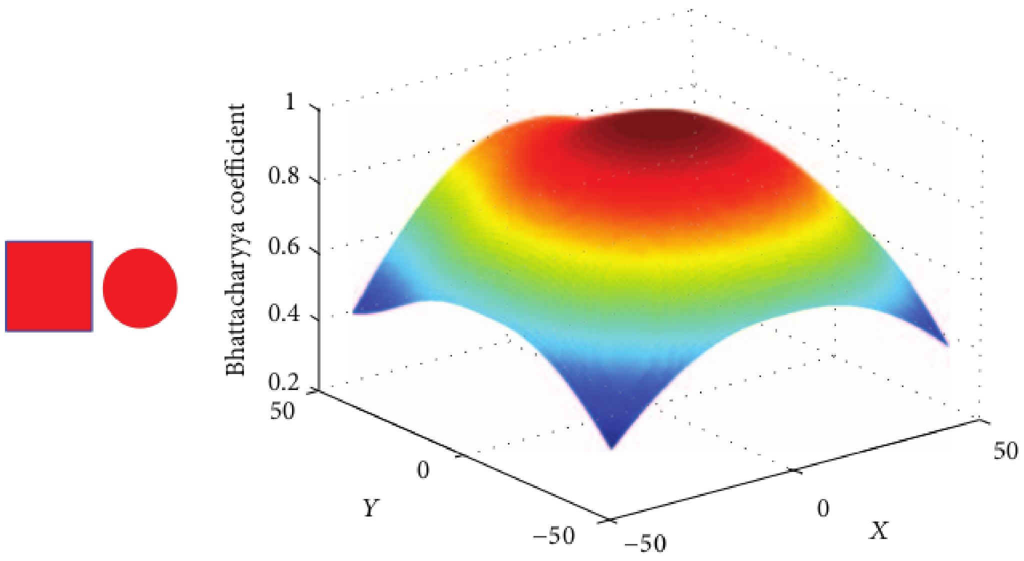

Figure 9: Input frame and Bhattacharyya coefficient surface obtained from the frame showing the bounding kernel assuming kernel size is the same as the target object size

4.2.3 Path Planning for Obstacle Avoidance and Tracking

Integrating target tracking and mapping tasks into the motion of the robot is a critical task of this project. Using the information provided by target tracking and mapping 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. The proposed method for path planning for obstacle avoidance and tracking is "Potential Field Method".

Yin - Yin [9] 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.

$$U = U_{attrative} + U_{repulsive} \tag{9}$$

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.

Based on this assumptions, the potential field given in Equation 9 can be found.

The attractive potential function is defined with respect to the relative distance, velocity and acceleration from the goal:

$$U_{att}(q, v, a) = \xi_q ||q - q_g||^i + \xi_v ||v - v_g||^j + \xi_a ||a - a_g||^k \tag{10}$$

where, ξ_q, ξ_v, ξ_a are positive constant scaling factors and i, j, k are scalar positive parameters.

Now, the virtual attractive force can be found by calculating the Laplacian of U_{att} with respect to the ego robot's position, velocity and acceleration.

$$\begin{aligned} F_{att}(q, v, a) &= -\nabla_q U_{att}(q, v, a) - \nabla_v U_{att}(q, v, a) - \nabla_a U_{att}(q, v, a) \\ &= F_{attq}(q) + F_{attv}(v) + F_{atta}(a) \\ &= \xi_q i ||q - q_g||^{i-1} e_{qrg} + \xi_v j ||v - v_g||^{j-1} e_{vrg} + \xi_a k ||a - a_g||^{k-1} e_{arg} \end{aligned} \tag{11}$$

where e_{qrg}, e_{vrg} and e_{arg} are unit vectors whose directions are the relative directions of the position, velocity and acceleration, respectively, of the goal with respect to the ego robot.

It can be concluded that F_{attq} tries to make the robot reach the position of the goal, F_{attv} tries to make the robot move with same velocity as the goal and F_{atta} tries to make the robot keep the same moving trend as the goal.

Below, a representation of F_{att} is given in a 2D workspace:

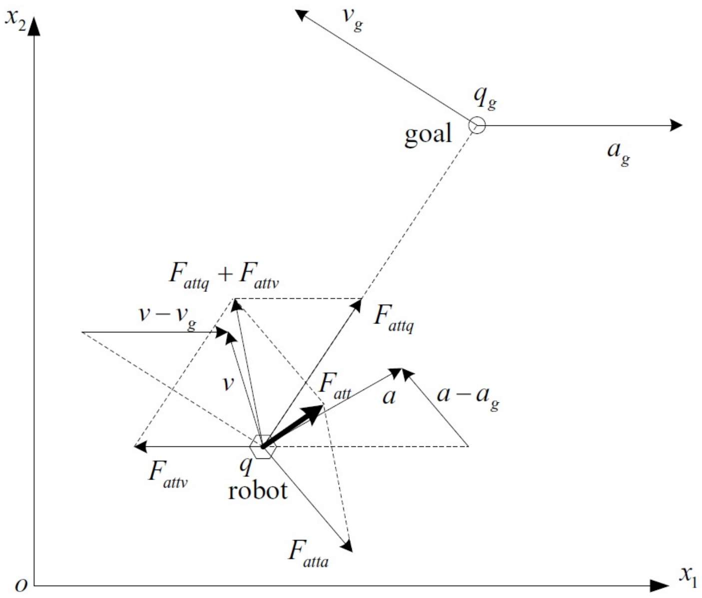

Figure 10: The process to get the attractive force in 2-D Workspace [9]

The repulsive potential function is defined as follows:

$$U_{rep}(q,v,a) = \begin{cases} \eta_1(\frac{1}{\rho_{obs}-R_{rob}} - \frac{1}{\rho_0}) + \eta_2 v_{ro} + \eta_3 a_{ro}, & \text{if } (\rho_{obs} - R_{obs}) <= \rho, v_{ro} > 0, a_{ro} > 0 \\ \eta_1(\frac{1}{\rho_{obs}-R_{rob}} - \frac{1}{\rho_0}) + \eta_2 v_{ro}, & \text{if } (\rho_{obs} - R_{obs}) <= \rho_0, v_{ro} > 0, a_{ro} <= 0 \\ 0, & \text{if } (\rho_{obs} - R_{obs}) > \rho_0 \text{ or } v_{ro} <= 0 \end{cases} \tag{12}$$

where R_{rob} is the radius of the robot, ρ_0 is a positive constant reflecting the influence range of the obstacle, and η_1, η_2 and η_3 are positive scaling factors, ρ_{obs} is the distance between the obstacle and the robot, v_{ro} and a_{ro} are the relative velocity and acceleration between the robot and obstacle.

Again, to find the virtual repulsive force, we need to calculate the Laplacian of the repulsive potential function:

$$U_{rep}(q,v,a) = \begin{cases} F_{repq} + F_{repv} + F_{repa}, & \text{if } (\rho_{obs} - R_{obs}) <= \rho, v_{ro} > 0, a_{ro} > 0 \\ F_{repq} + F_{repv}, & \text{if } (\rho_{obs} - R_{obs}) <= \rho_0, v_{ro} > 0, a_{ro} <= 0 \\ 0, & \text{if } (\rho_{obs} - R_{obs}) > \rho_0 \text{ or } v_{ro} <= 0 \end{cases} \tag{13}$$

where $F_{repq} = -\nabla_q U_{rep}$, $F_{repv} = -\nabla_v U_{rep}$ and $F_{repa} = -\nabla_a U_{rep}$.

In a similar sense to the attractive force, F_{repq} tries to keep the robot away from obstacle, F_{repv} will make the repulsive effect stronger and F_{repa} tries to make the robot keep an opposite moving trend from the obstacle.

Below, a representation of F_{rep} is given in a 2D workspace:

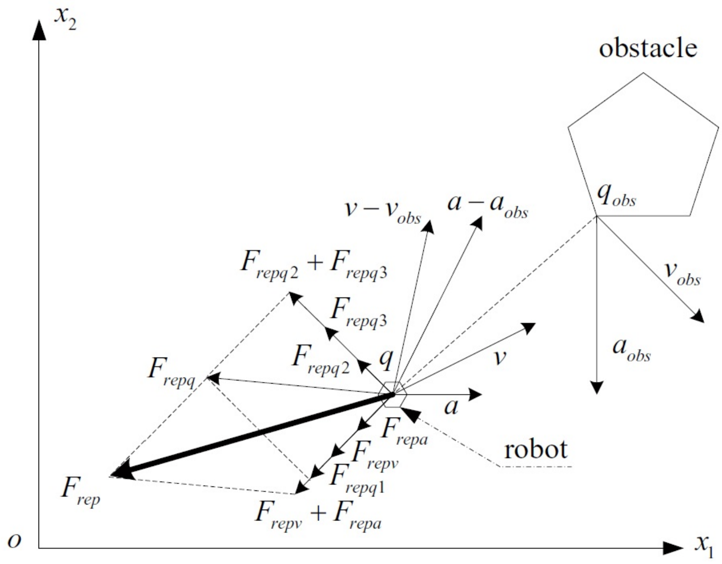

Figure 11: The process to get the repulsive force in 2-D Workspace [9]

One of the possible risks concering this method is that surrounding obstacle may not be always convex polygons. Therefore, an alternative method or a trivial solution might be needed. In this sense, our approach might be drawing a convex hull around the non-convex objects, which will satisfy the method's 3rd assumption.

4.2.4 Simulations on Gazebo Software

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:

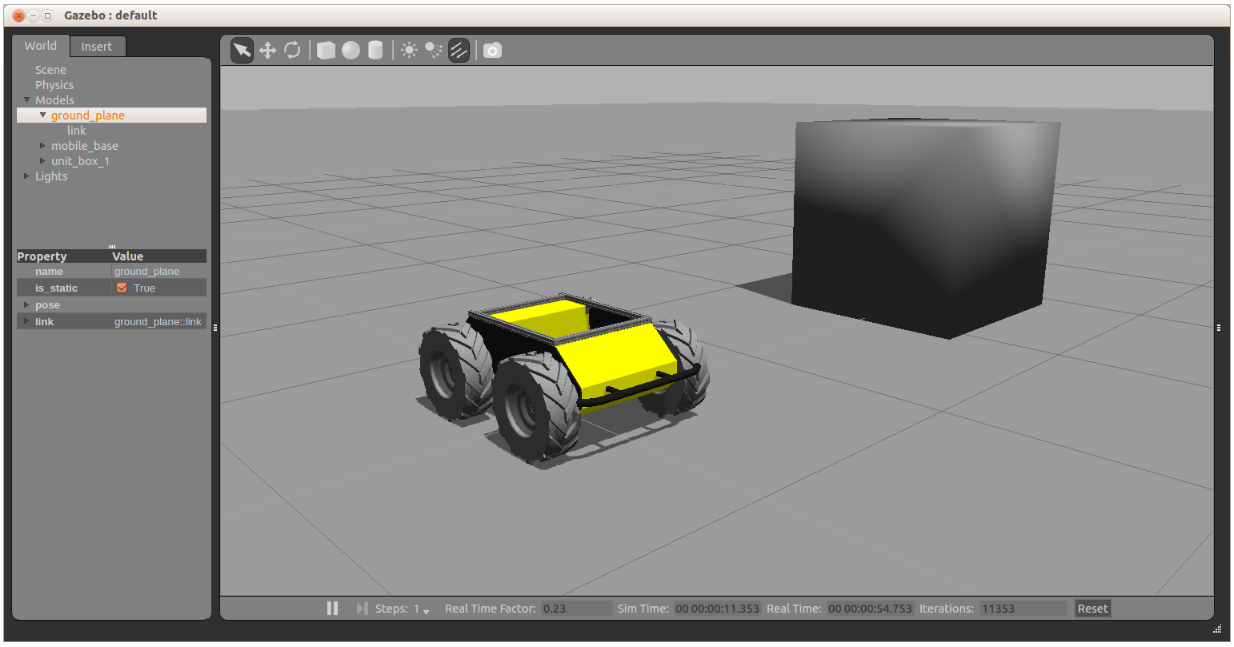

Figure 12: Gazebo Simulation Software [8]

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.

4.2.5 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.

As Brambilla [1] 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[7]:

- **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[1] 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.

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.[3] 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.

5 Equipment List

Equipment	Cost (in TL)	Obtained
Nvidia Jetson Nano	1115	by purchase
Eken H9R 4K Action Camera	378	by purchase
RPLIDAR A2 M8	No Cost	from ROKETSAN
Dagu Wild Thumper 4WD All-Terrain Chassis	1305	by purchase
FPV 2 Axis Brushless Gimbal With Controller	175	by purchase
Total Cost	2973	

Table 1: BoM of the Project

The Bill of Material is given in Table 5, which consists of all the materials 1 robot requires. Currently, our project objective is to build at least 2 robots in order to construct the swarm behaviour. Therefore the unit cost is estimated to be 2973 TL for one robot, without considering the battery cost.

Note: The motor drivers and battery of the robots are to be decided after performance measurements and tests on the Dagu Wild Thumper 4WD All-Terrain Chassis and to be consequently added to the equipment list.

References

[1] 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.

[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] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.