PROTOTYPING AND SYSTEM ENGINEERING Team ENTROPY

1st Abdullah Zafar
Electronic Enigenerring
Hochschule Hamm Lippstadt
Hamm, Germany
abdullah.zafar@stud.hshl.de

2nd Mustafa Touqir
Electronic Enigenerring
Hochschule Hamm Lippstadt
Hamm, Germany
mustafa.touqir@stud.hshl.de

3rd Dawar Zaman
Electronic Enigenerring
Hochschule Hamm Lippstadt
Hamm, Germany
dawar.zaman@stud.hshl.de

4th Syed Muhammad Saim
Electronic Enigenerring
Hochschule Hamm Lippstadt
Hamm, Germany
syed-muhammad.saim@stud.hshl.de

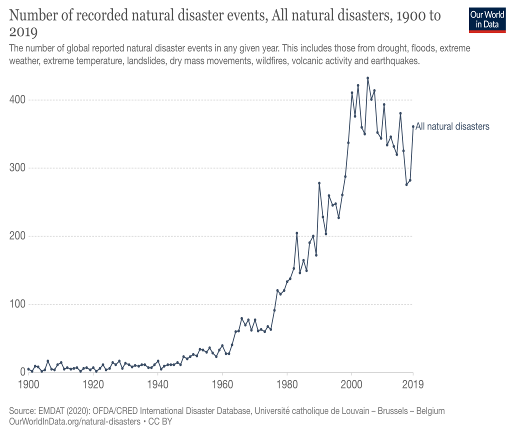
***Abstract*—In this paper we elaborate and discuss the development of a new rescue robot platform. The robot is designed to drive on land and water and even on a rugged landscape. Sensor fusion allows autonomous system to bring together inputs from multiple ultrasonic sensors, lidars and cameras to form a single model or image of the environment around the vehicle. Basic steering and experiments for the robot are also shown. Moreover, this document include a prototype model that is the best fit for the given scenario. Lastly, A robot simulation in C language is also included, that demonstrates how the robot will react in real life scenarios.**

I. Introduction

Natural disasters are on the rise due to multiple factors, including climate change, earthquakes and forest fires etc. The number of natural disasters has increased exponentially in the last two decades. It is always suggested that rescue operations should be on their toes, when such unfortunate incidents takes place. There some instances were human beings cannot physically perform a rescue operation themselves, mainly due to the fact that it is very dangerous for the rescue team as well. That is where rescue robots can play a pivotal role to minimize the damage caused by the disaster.

Looking back into history, Rescue robots were first used in the search for victims and survivors after the September 11 attacks at the world trade center in New York [1]. Many Government and private organizations have been recently involved in research and development for the production of a autonomous rescue robot. Figure 1 shows some statistics for number of natural disasters recorded from 1900 to 2019. As can be seen in the chart the number has increased exponentionally.

There is an estimate that how many people have been affected by urban disasters and may be affected in the future as mentioned in the 2005 World Disaster Report. Between 1995 and 2004, Over 900 000 people were reported killed from 1995 to 2004, with the total amount of disaster related damage estimated at 738 billion US dollars. Of the victims in urban disasters, only a small fraction may actually survive. Despite the fact that this is typically where the majority of casualties are found, only 20 percent of survivors of urban disasters emerge from the rubble's interior, creating motivation for robots that can probe deep within collapses. After 48 hours, the mortality rate rises and peaks, indicating that survivors who are not extricated within the first 48 hours are unlikely to survive more than a few weeks in the hospital. [2]



[3]

Fig. 1.

II. Applications of Rescue Robot

While the ultimate goal of rescue robots is to save lives, the motivation for individual robot designs and capabilities is determined by the tasks that they could perform. The following are some of the jobs that have been proposed for rescue robots. A search is a focused action that takes place inside a structure, in caverns or tunnels, or in the outdoors, with the goal of locating a victim or potential threats. The purpose of the search is to be as extensive and early as possible without endangering the victims or rescuers. Reconnaissance and mapping are subsets of search. It provides a basic overview of the situation to rescuers and acts as a reference for the devastation. The goal is to cover a big region of interest quickly and at the right resolution. Robotic machinery or exoskeletons can help remove debris faster. The goal is to carry larger amounts of rubble faster than could be done manually, yet with a smaller footprint than a standard construction crane. [2]

III. SYSTEM ENGINEERING

One of the most important phases of developing a rescue robot was to System Engineer the entire project. In this phase, Requirements and elicitation were chalked out, and the purpose of our rescue robot was developed. Amongst all the UML and Sysml diagrams, Use Case diagram, Activity diagram, Requirement Diagram, Constraint and Block Diagram, have been included in this document for explanatory reasons.

A. Use Case Diagram

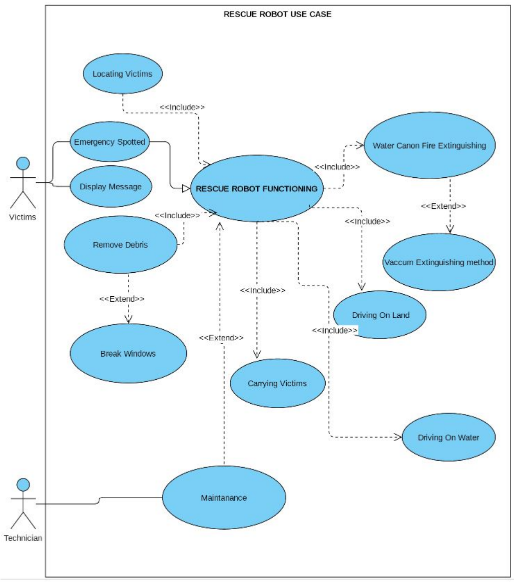


Fig. 2.

The use case diagram for our rescue robot has a main use case which is named as ‘RESCUE ROBOT FUNCTIONING’. It has a various include relations and several extend relations. The system has use cases such as locate victims, water canon fire extinguishing, driving on land, driving on water, carrying victims and remove debris. These use cases have an include relation with ‘RESCUE ROBOT FUNCTIONING’. There is a use case vacuum extinguishing method that has an <<exclude>> relation with the water canon fire extinguishing and the use case break down has an <<extend>> relation with the remove debris. In this diagram, two actors have been shown. Technician, which is an actor, has a relation with the use case maintenance whereas victims has relations with the use cases emergency spotted and display message. The victims, which is an actor, will display a message the rescue robot will get the location where an emergency is spotted. The diagram explains that our rescue robot has various functions for its functionality along the rescue journey. It will drive on land if the terrain is land whereas, it will drive on water if the terrain is water. When fire is erupted, it will be extinguished by water cannon. The fire can also be extinguished by vacuum extinguishing method. In case any hurdle, the rescue robot break those hurdles for example if there were windows then our rescue robot will break them as shown in the diagram. On reaching the victims at an emergency spot, the rescue robot will carry the victims and bring them to a safe place. In case of the any failure in the functionality of the our rescue robot, the technician, which is another actor, is responsible for the maintenance of the rescue robot and he will do the task promptly.

B. Activity Diagram

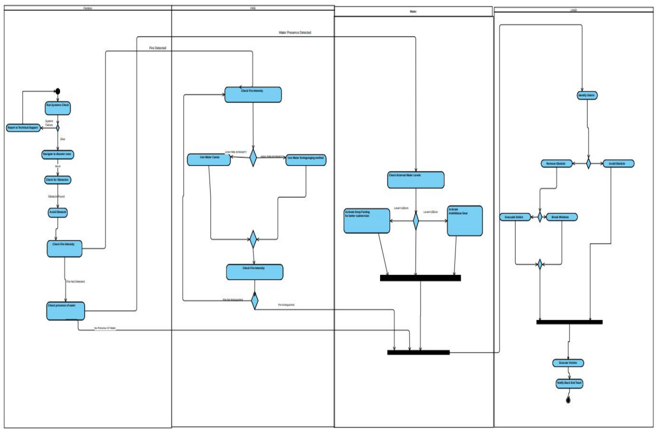


Fig. 3.

This is the activity diagram of our rescue robot that explains the functionality through subsequent activities. As you can see, it has four sections and the system moves back and forth according to the situation. The activity process begins in the first section i.e, *Partition*, and moves to the first activity *run system check*. If there is a failure, the process will move again to the initial position otherwise it will move to the next activity i.e, *navigate to disaster zone*. Then it will move to the next activity subsequently *check for obstacles* and then *avoid obstacle*.

When it reaches the activity of *check fire intensity*, the process will shift to the second section named as *FIRE*. Here, it will check the *intensity of fire*. If the intensity is normal and it is possible that fire can be extinguished by water then fire will be extinguished by the help of a *water cannon* otherwise if the intensity of fire is too high, fire will be extinguished by *fire extinguished method* which include the vacuum method (this method is mentioned in use case diagram). At the end of section two, the process will again check the intensity of fire. If the intensity is not reduced then the process will start over again to extinguish fire otherwise fire is extinguished and the process will combine with the synchronize bar in the third section.

Coming back to the first section, if the fire is not detected, the process will move to the next activity *check presence of water*. If the there is no water detected, the process will join with synchronize bar in the third section otherwise it has to pass the process in the third section named as *water*. In this section, there is a first activity *check external water levels* which checks the level of water. If water level is greater than 160 cm the process will move to the activity activate *deep fording for better submersion* and if the water level is greater than 185 cm the process will move to the activity *activate amphibious gear*. Eventually, the process will merge with a synchronize bar and that synchronize bar will merge with another final synchronize bar.

The final synchronize bar takes all of the previous processes with it and enter the whole activity process in the final section named as Land. The first activity mentioned in this section is *identify debris* moves to a decision node and our rescue robot, based on this activity diagram, decides whether to *avoid the obstacle* or *remove the obstacle* from its way. In case the obstacle needs to be removed, the process moves to another decision node which will decide to *evacuate debris* or *break windows*. If we move forward the process merges through a merge node and final combines with the synchronize bar. If our rescue robot has to avoid obstacle, based on this activity diagram, the rescue robot just simply avoid the obstacle and join the process into synchronize bar. The final synchronize bar moves to the subsequent activity *evacuate victim* and finalizing the whole process of activity diagram by notifying the *back end team*.

C. Requirement Diagram

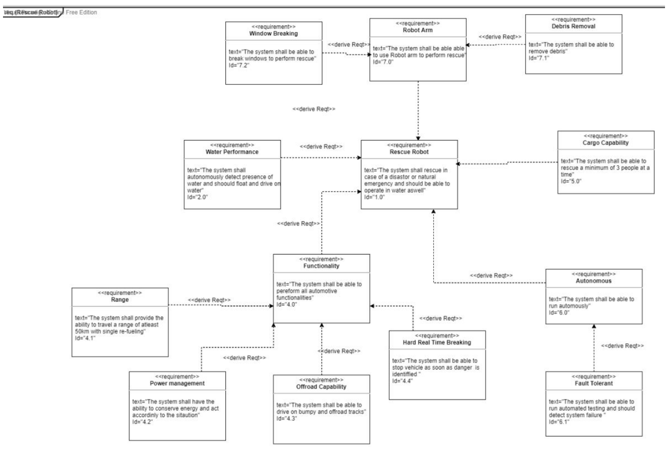


Fig. 4.

The requirement diagram of our rescue robot is quite self-explanatory. Every requirement in the diagram includes a text that explicitly explains the requirements that are need to be fulfilled in our rescue robot. All of the requirements have *¡¡derive Reqt¿¿* relation with each other and finally they merge in the main requirement named as *rescue robot*. The requirement diagram includes a functionality requirement, which further has four sub-functionalities such as, *range*, *power management*, *off-road capability*, *and hard real time braking*, that combines with the *rescue robot requirement*. A subsequent requirement can be shown if we start looking through *fault tolerant requirement* which merges into *autonomous requirement* and finally it merges to *rescue robot*. One of the important requirements is the *robot arm* which includes *debris removal* and *window breaking* and finally combines with the *rescue robot* requirement. The explanation of all these requirements can be read inside the requirement boxes.

D. Constraint Diagram

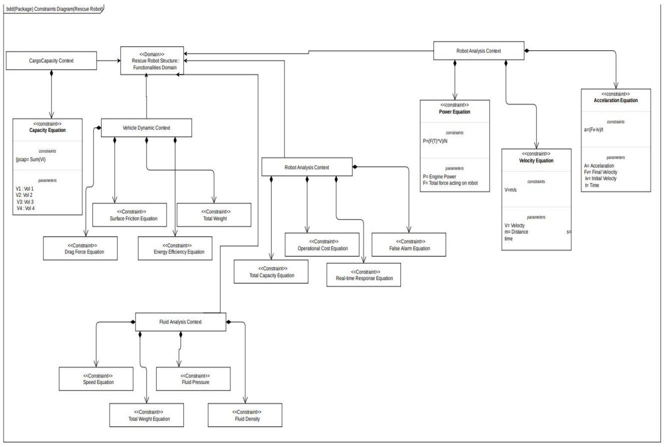


Fig. 5.

This diagram of our rescue robot explains many constraints that can be attached with our rescue robot. It has a *capacity equation* which can be seen under the *cargo capacity context* at the top-left corner. The *¡¡Domain¿¿*, which is shown next to *cargo capacity context*, named as *rescue robot structure* that include *functionalities domain*. All the constraints merge into their respective contexts that finally combines with the *¡¡Domain¿¿*. The first context is named as *vehicle dynamic context* comprise of four constraints such as, *drag force equation*, *surface friction equation*, *energy efficiency equation* and *total weight*. The second context is named as *Fluid analysis context* that also includes four constraints such as, *speed equation*, *total weight equation*, *fluid pressure* and *fluid density*. The third context is named as *Robot analysis context* that also mentions four constraints such as, *total capacity equation*, *operational cost equation*, *real-time response equation* and *false alarm equation*. The last context is named as *robot analysis context* comprise of three constraints such as, *power equation*, *velocity equation* and *acceleration equation*. The constraints in the last context are further explained by including equations and variables.

E. Block Diagram

This is a block diagram of our rescue robot that shows several paramount components inside our robot. On the left-hand side, you can see some actors such as *passengers*, *controller* and *vehicle occupant*. The external components,

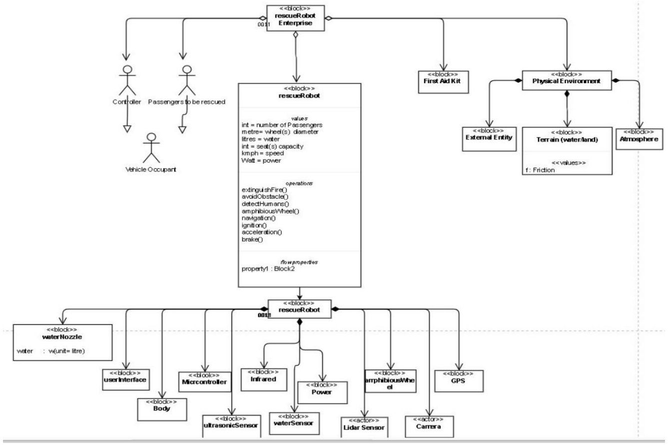


Fig. 6.

which is not the main part of the rescue robot, can be attached to our rescue robot that is shown on the right-hand side of the diagram. In the center values and operations are highlight that gives an idea of the functionality of the robot.

Our recue robot has several values that include number of passengers that can be entered in the rescue robot. This is defined as an integer data type to store only number for this variable. Secondly, the diameter of the wheel has been shown to give an idea that what is the suitable size of wheels should be used. Thirdly, water in liters is mentioned to tell the capacity that how much water the robot can store in it. Water can be helpful in extinguishing fire, so it is one of the important information. Fourthly, the speed of the rescue robot is included to show that how faster or slower it can travel along a rescue journey. Lastly, the power of the rescue robot in Watt is added to give an understanding that how much time and difficulty our robot is going to withstand.

The designed robot comprises of various functions that include *extinguishFire* to give an order to the robot to extinguish fire wherever the fire erupts. Secondly, it mentions the function of *avoidObstacle* that informs the robot to avoid any hurdle that comes in the rescue journey. This can be done either by destroying the obstacle or making a way through it. Thirdly, it has a function of *detectHumans* that will identify humans and hence, rescue them. Fourthly, *amphibiousWheel* function, which is shown in the diagram, detects the water on the ground along the rescue journey. The purpose of this function is to open the amphibious wheel to pass the water terrain feasibly by floating on the water like a boat. Fifthly, it has an ignition function the purpose of which is to turn on the vehicle. Lastly, it has an *acceleration* function together with the *brake* function which will drive our rescue robot along the rescue journey.

As we move down into the block diagram, the diagram explains some of the components that can be classified as the building blocks of our rescue robot. It has a *water nozzle* that sprinkles the water in the place of fire together with the water sensor. In addition, it has a *user interface* that displays the functionality of the rescue robot and to control it. Moreover, it has a *body*, which is indeed an essential component, that must be fault tolerant. Furthermore, to control the rescue robot, it has *microcontroller* which is connected to all other sensors such as *lidar sensor*, *infrared sensor*, *water sensor* and *ultrasonic sensor*. It has also a *camera* to record and watch the rescue journey and *GPS* that helps the rescue robot to navigate. This diagram includes all the necessities based on an ideals scenario that are required for a rescue robot to work efficiently which might not appear hundred percent when implementing it practically.

IV. Prototype Design Phase

An iterative method Was followed along the Prototype Design Phase, this was largely due to the fact that it was really hard to get every thing right in the first attempt or in a single go. Solid Works was used for the modeling phase, since all the team members were comfortable and familiar with the software and its working.The following subsections give a brief picture of the entire journey that TEAM ENTROPY went through when we were designing the Robot. To simplify the process that we went through, Team Entropy has divided the entire modeling period in four phases.

A. Phase I (Requirements and Inspiration)

As briefly discussed in the system engineering section, our robot was supposed to have a robot arm, a ladder and a water canon. Moreover the Wheels were supposed to be amphibious, that helps it to drive on land and water.

Robot arm would have helped the rescue mission by breaking windows and by removing Debris. Water Canon was an integral tool for fire extinguishing and it could also remove the debris with high water pressure.

To fulfill all the requirements, Team Entropy came to a conclusion to build something that is huge, and is able to rescue a minimum of three people. The motivation initially came from the TESLA CYBER TRUCK. Cyber Truck is a compact and powerful machine, We believed that it would be the best fit for the given scenario.

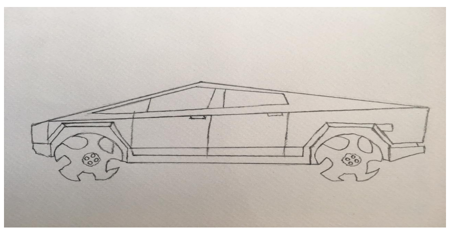


Fig. 7. SIDE VIEW

Figure 7, shows the initial sketch of the Robot. This is basically a side angle that helps us visual how the wheels would look like.

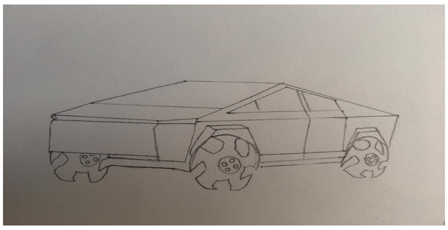


Fig. 8. tilted side angle of the vehicle

As mentioned earlier, We wanted to include a Robot Arm in our prototype, that was to remove debris and to break windows. Our team worked really hard to design a simpler model of robot arm, that can be easily integrated and designed.

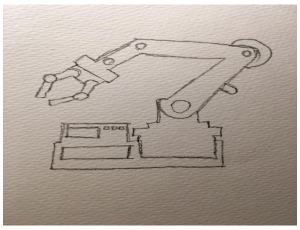


Fig. 9. Robot arm 1

Figure 9 and 10, were the two sketches that our team finalized. Figure 9 was simpler to build as compared to figure 10, but it seemed like that it would spoil the entire image of the vehicle, as it was not as impressive as we wanted to build. The main concern was that Figure 4 was really complicated to build on Solid works;nevertheless, we decieded to proceed with it. Initially, the water canon was not included, it was due to the fact that TEAM ENTROPY was a little skeptical on its inclusion, as all the heavy duty tasks, were handled by the robot arm.

Another important aspect of the vehicle was the tires. There were many possible solutions to drive a vehicle on land and water, we proceeded with the most easiest and simplest model.

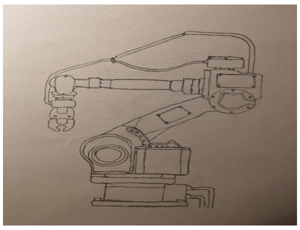


Fig. 10. tilted side angle of the vehicle 2

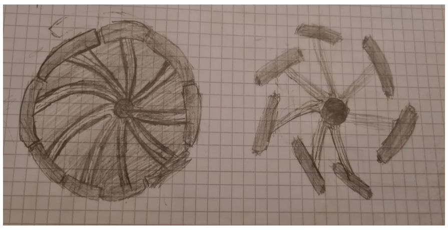


Fig. 11. Wheels of the vehicle 1

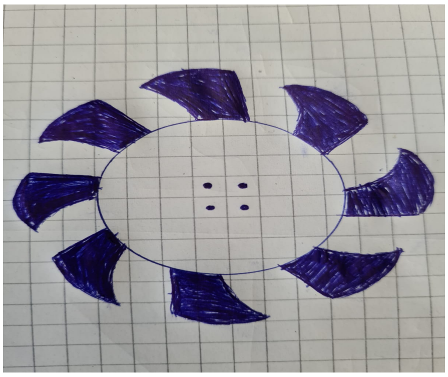


Fig. 12. Wheels of the vehicle 2

Figure 11 and Figure 12 shows the two tires prototypes sketches that could be used in the vehicle. Figure 11 was much more complicated, as it changed its structure depending on if its in water or on land. In figure 11, The sketch on the left shows how the wheels would have been when in water, The rotation of the tires would help the vehicle in moving forward. Figure 12 was something very simple, there was no need to change the structure, regardless of when in water or land. The fins of the wheels were designed with such an angle, when the wheel rotated in water, the vehicle would go forward. This prints the parameter passed inside the braces on the left side of the footer.

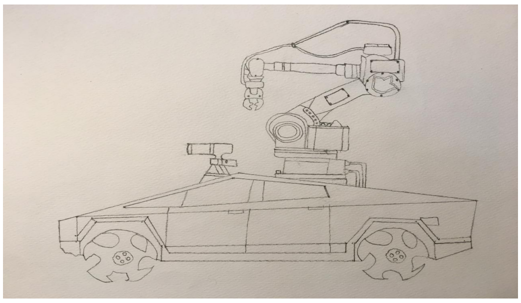


Fig. 13. Final sketch result

Figure 13 shows the end result of PHASE I. It clearly shows that a small water canon was placed on the top. Either a large water canon should have been part of the vehicle, or nothing at all.

B. PHASE II(Initial Prototype design on Solid works and complications)

The first step for designing a prototype on Solid works calls for chalking out the outline of the main body with proper dimensions. The dimensions were inspired from a modern day truck and the dimensions were scaled down to 10 folds. This is by far the most important step when designing on Solid works, in case there is a fault with the main body outline, the entire structure deteriorates and you start moving away from the main goal.

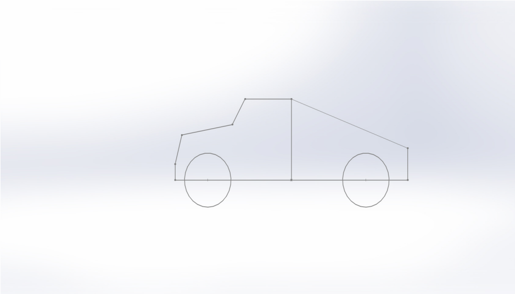


Fig. 14. Outline for Solidworks

Figure 14. gives a rough overview of the side view of the vehicle. It is clearly visible that all the aspects of the vehicle are kept in a good proportion, and it is a good starting point for further developments.

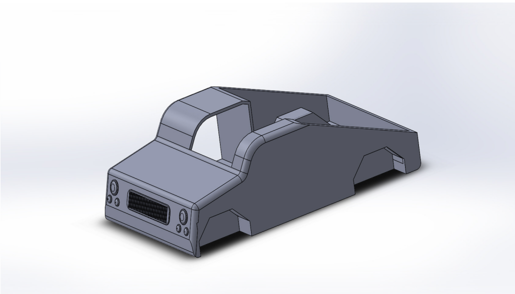


Fig. 15. Main body Initial Design

Figure 15. shows the next step in the designing phase. The face front was not as impressive as Team Entropy would have liked. The vehicle was made open, considering the fact that this vehicle would be performing a rescue operation. Adding additional doors would be a barrier to our rescue operation, in other words, it would have slowed down the rescue process to a very big extent. There was initially only one entry point, but this issue was later addressed in the steps to come.

Figure.16, shows a very important angle of the vehicle, the bottom side of the vehicle was initially not made flat. That meant that water could have collected under the vehicle, hence making it difficult to function in water.

It was assumed that the body of the vehicle would be made out of Aluminium, as it is one of the strongest and light weight metal that exist. Moreover, the fact that aluminium is used to manufacture Air crafts lends credibility to our assumption.

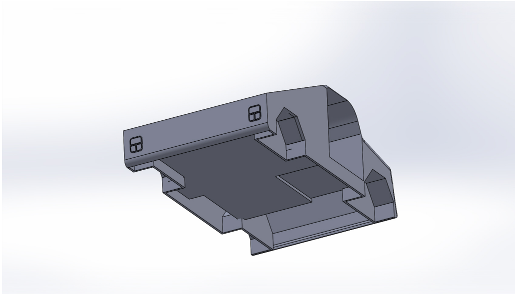


Fig. 16. Bottom Vehicle View

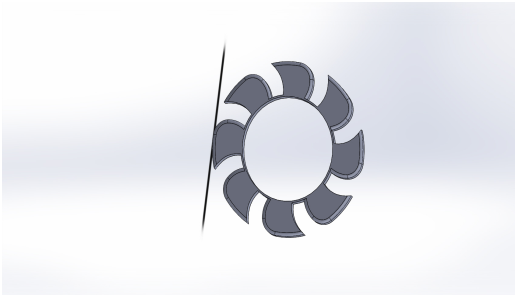


Fig. 17. Wheel Design in Solidworks

Figure 17. shows the first prototype design of the tire or the wheels of the vehicle. As mentioned earlier, the tires should be able to operate on land as well as water. The angle of the slight length of the tires were kept such that, it can operate on water and land as well. Choosing the right numbers of paddles per tire was also a very important task. Theoretically speaking, in case the number of paddles per tire was kept to four, that would mean that on land, the vehicle would not have a smooth ride, that after every 0.25 revolution of the tire, the vehicle would face a bump. The ideal number of paddles per tire was eight, that meant that both halves of the tire were equally weighted.

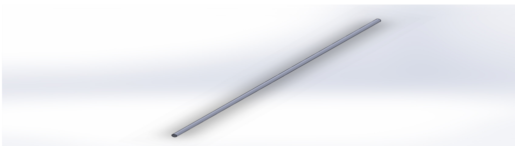


Fig. 18. Wheel axle

Figure 18. shows the illustration of the wheel axle. This component would be plugged into both the wheel hubs, in order to make the wheels of vehicle functional.

Figure 19 shows the prototype model of the wheel hub. As mentioned earlier, combination of wheel axle, tire and hub would make a complete set of wheels.

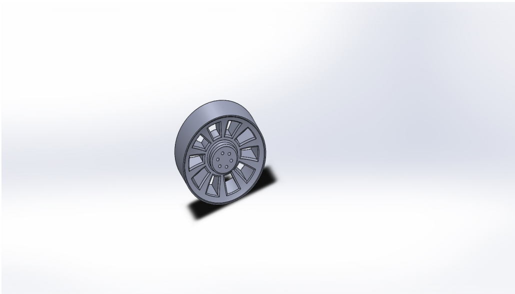


Fig. 19. Wheel hub

This subsection demonstrates the major outcomes of the phase II of the designing phase. Assembly was planned to be completed in Phase IV, as the Team Entropy was still unsure if many of these components were really required. The designing of the water canon and the robot arm was still pending in this phase and was to be completed in the stages to come.

C. PHASE III(Electronic Components integration and vehicle movement)

Two of the most important parts of the project included the integration of the electronic components and vehicle movement. A logical solution was to be presented that shows the vehicle is able to steer as well, rather than just moving forward and backward. A potential solution to this problem was addition of a steering gear, that helps to change the direction of the vehicle. That was a good solution but designing it on Solid works was a long and tedious task. Nevertheless, Team Entropy came up with an effective yet simple solution that you ease the task and would help to achieve the objective at the same time.

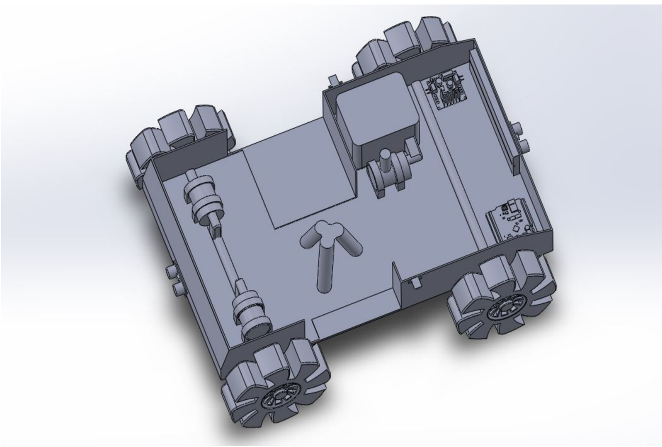


Fig. 20. Chassis of the vehicle

Figure 20. demonstrates that the rear wheels of the vehicle has to additional motors, that would be driving the vehicle. That meant that both the rear wheels were not interconnected and the wheel axle that was designed in Phase II was no longer required. Maneuvering the wheel was achieved by installing two motor in the rear. The working is really simple, for instance, if the wheel is to be steer to the left, the left wheel would stop and the right wheel will rotate in clockwise direction. That would help to change the direction of the vehicle. In case, the vehicle needs to turn right, the same logic is followed, but vice versa.

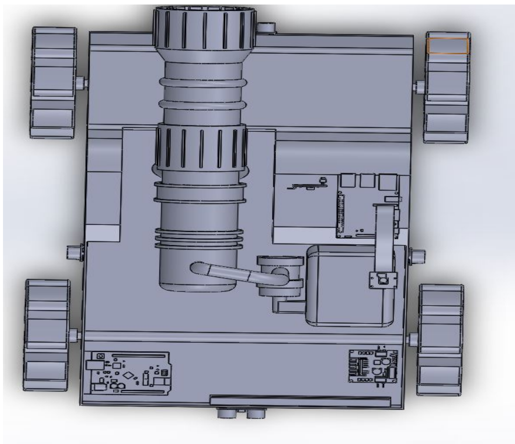


Fig. 21. Top view of vehicle

Figure 21. is extracted from the results that were achieved after Phase IV(briefly explained in following subsection). Figure 21. shows the integration of the main electronic components, including, Adruino UNO, Motor driver and Rasberry Pi. The reason was placing them on the back of the vehicle was that our team were unable to fix these components in the rear, mainly due to unavailability of space. PHASE III ended up producing the best results but there were a lot of tasks that were to be completed, including the assembly of the vehicle and addition of a robot arm and water canon.

D. PHASE IV(Final Phase)

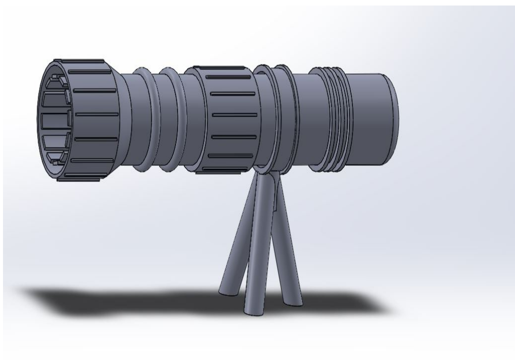


Fig. 22. Water canon

The first barrier that was addressed by TEAM Entropy was to decide, if a water canon or robot arm was to be included. As mentioned earlier, Robot arm is something that is really complex and was hard to design. Water Canon was opted for the desired functioning of the vehicle, but it was to be ensured that water canon was powerful enough to remove debris and extinguish fire. Moreover, a water tank was to be added, that you supply the water canon with water and an additional motor was required to drive the water canon with maximum power.

Figure 22. shows the design of Water Canon. The whole machinery stands on a tripod which is to be later on installed into the vehicle. The water canon looks simple, yet very effective and powerful.

Another main task that had to be addressed in this phase was to prevent the water from flowing into the vehicle. Water could have entered the vehicle from the wheel hub or the main door area. Addition of ball bearing could have solved the problem. However, another approach was adopted, that was the addition of a floater under the vehicle. The floater would stop the tire hub from dipping into the water, it will only allow the paddles of the tire to submerge into the water.

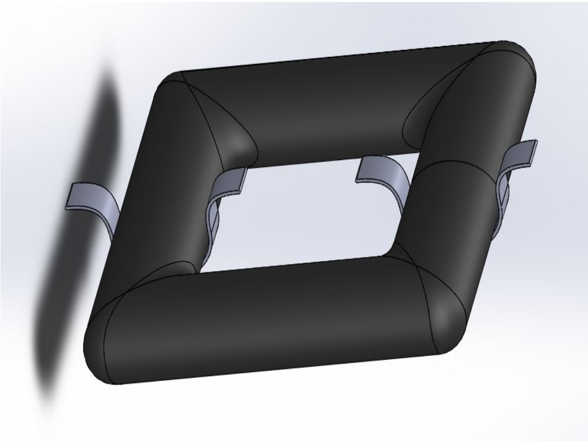


Fig. 23. Floater

Figure 23. shows the floater that is to be installed under the vehicle. That will ensure that the wheel hub remains above the water level and that water does not enter inside the vehicle. The floater is assumed to be made out of rubber and is inflated with air.

Fig. 24. Water Pipe for joining water canon to water tank

Figure 24. shows the model of water pipe that would connected the water tank to the motor and motor to the water canon. The water tank was designed with optimal size, it was ensured that it does not occupy a lot of space, or else there would not be any space to rescue three survivors.

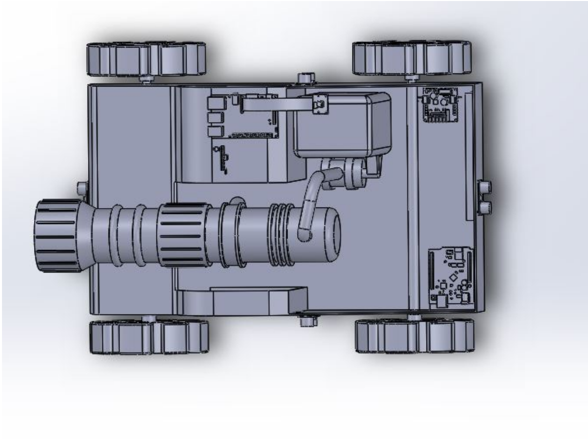


Fig. 25. Top view of vehicle

Figure 25. shows the top view of the vehicle. It shows the rearrangement of the electronic components that were used. The connection between the water tank, motor and water canon can also be easily located. An important addition that was previously missing in PHASE III, was the ultra sonic sensors. Ultra sound sensors are needed for the vehicle to compute the distance between its self and the surrounding. The vehicle drives autonomously, so that was really important to install the sensors in all four sides of the vehicle. The top view also shows that there is enough space to rescue to maximum of three people, moreover, all the desired tasks may be fulfilled using the features that this vehicle contains.

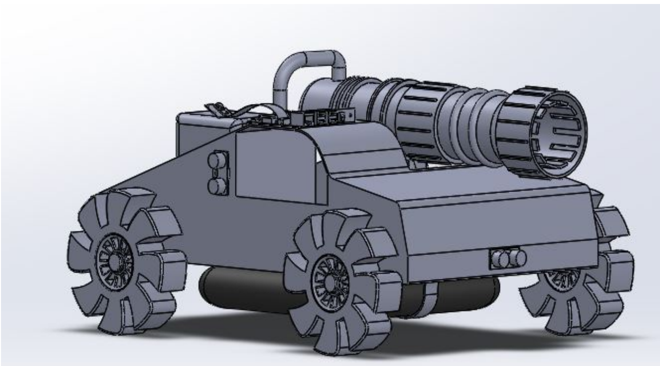


Fig. 26. Final Robot Model

Figure 26. shows a good angle of the final vehicle. Ultra sonic sensors can be seen on the front and the right side view of the vehicle. The wall on the door side of the vehicle was increased to prevent the water from entering. It was suggested that when the vehicle enters the water for the first time, the

vehicle would be tilted and the water may enter from the door side into the vehicle.The main problem associated with water entering the vehicle was that, most the electronic components were installed at ground level. Water might result in major malfunctioning and would remove the purpose of the vehicle.

Figure 26. Furthermore, allows us to make an educated assumption, the water would not be able to reach the center of the tires, that is mainly due to the large diameter of the tires and the floater that was installed under the vehicle.

The designed was initially inspired by TESLA Cyber Truck, but the outcome looks completely different, that was achieved on purpose, and the designing was done to achieve the required task. The front face that was made as a result of Design PHASE II, was changed completely as well, that is due to the reason that it did not synchronize well with the vehicle. Moreover, it gave the vehicle a funny look and our aim was to make something that is compact and powerful.

V. Implementation Phase

A. Project Requirements / Task Objectives:

In the implementation phase, four tasks were assigned as part of the virtual simulation of the robot in various conditions these four tasks were a part of a single big task that involved the navigation of the robot in a 2D array map with various conditions and various outcomes depending upon those conditions. The task complexity was increased with each consecutive task. Initially, the task 1 was assigned in which the map consisted of a wall , a robot R, an empty space O and a target T as shown in Figure 27.

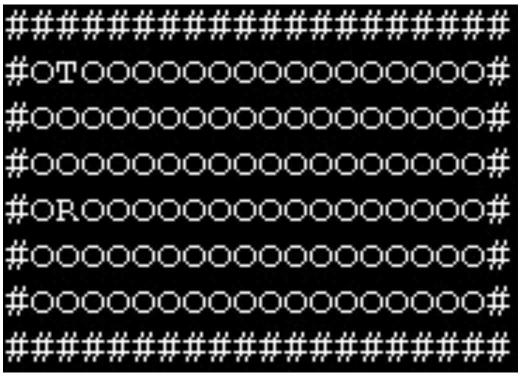


Fig. 27.

The objective in this map was to move the robot R in the map without hitting a wall and to reach the target T within 200 steps. Also, this was to be done using functions that returned values 1, 2, 3 and 4 for directions North, East, South and West respectively.

The second task consisted of the same objects in the map but with an addition of water represented by . The objective for the second map was that the robot R should reach the target T without hitting a wall and incase the robot R encounters water in its path then it should toggle to water mode by returning a value 5 from the function and it should switch between water mode or land mode by returning 5 depending upon the situation it is facing. This task too was meant to be completed within 200 steps.

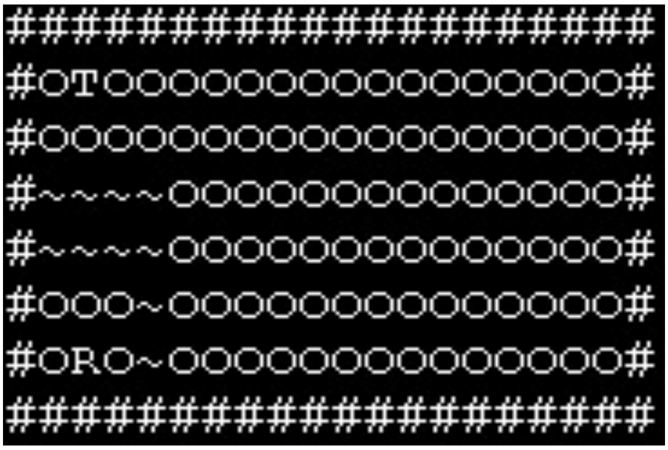


Fig. 28.

The third task was the same as the second task but with the map containing additional features such as home X and obstacles $*$ and the target T was now assigned as a land target. The concept of energy points was also introduced in this task such that the robot consumes 10 energy points per each movement and 30 energy points per each toggle between the land or water mode. The objective of the task 3 was to navigate the robot R to the land target T by avoiding obstacles $*$, walls , switching between water/land mode depending upon the situation and then returning to the point where the robot R initially started which is called home and is labeled as X in the map. The objective of task 3 was supposed to be done by consuming as less energy points as possible.

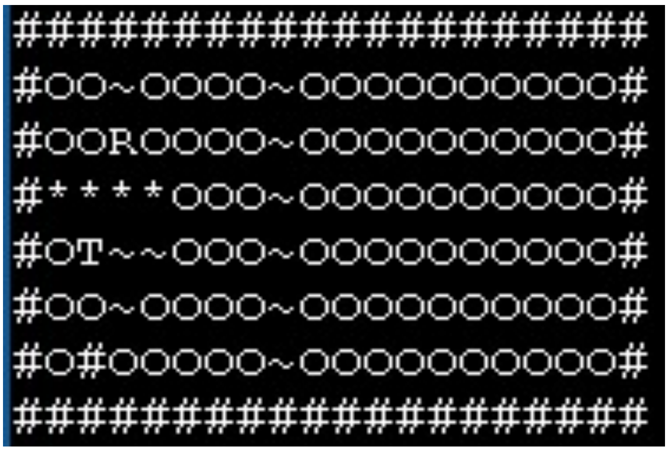


Fig. 29.

The fourth task consisted of an additional logic that the obstacles $*$ could now be destroyed and in order to destroy the obstacles $*$ the robot R should take up 70 energy points. At this point the function(s) returned values 1,2,3,4,5,6,7,8,9 for movement north, movement east, movement south, movement west, toggle land/water, destroy obstacle north, destroy obstacle east, destroy obstacle south and destroy obstacle west respectively. Each successive task had incremental objectives meaning that each new task had the requirements of the previous task plus additional requirements as part of the new task. This meant that in order to complete the third task, the second task had to be completed.

```
####################
#~t~*OOOOOOOOOOOOOO#
#~~~*OOOOOOOOOOOOOO#
#****OOOOOOOOOOOOOO#
#OOOOOROOOOOOOOOOOO#
#OOOOOOOOOOOOOOOOOO#
#OOOOOXOOOOOOOOOOOO#
####################
```

Fig. 30.

B. Code Explanation

The find_path and $move$ functions are declared in the header file and the rest of the code is in the $.c$ file. The code starts by including the libraries and the header files then the important integer and character data types such as char coming_from = ‘X’ are defined.

The r_x, t_x, r_y and t_y are robot and target indexes mapped to x and y coordinates respectively this way they can be used easily in the remaining part of the code.

```
r_x = robot_index / raw_count;
t_x = target_index / raw_count;
r_y = robot_index % raw_count;
t_y = target_index % raw_count;
```

Fig. 31.

A function world[robot_index] = ’V’ is also defined so the robot does not visit the same place where there was a wall again. The navigational function will check recursively from the robot coordinates to the target coordinates. In each step robot moves in the map, robot will go to the target and verify the target is reachable. But at the end it will return where should the robot go to in order to achieve the target. In this operation if we want to change the traveling mode we will return 5 instead of the direction value. In next iteration it will go to the next tile. The function ”find_path” is used to develop the program in clear manner and efficiently. It will return 0 if we cannot reach the target. So if we check four paths from the current location, we can get a clear idea whether or not we can reach the target. after we reach the target, we should have a way to know it. So we can write values to the variable “facing” so we can use that for checking purposes afterwards. The values from the navigational functions (that are responsible for detection of wall , obstacle $*$, water   and the directions in which they are) are returned in the form of values 1-9 depending upon the situation then these values are used by the int update_world(int movement, char *world, int robot_index, int width) function a sample is shown in the figure where if the navigational function returns the value 5 then the energy becomes equal to ENERGY_TOGGLE which was declared to be 30 at the start of the code. It also switches between the land and the water mode depending upon the mode it was previously in..

```
else if(movement == 5) {
    energy += ENERGY_TOGGLE;
    printf("Toggling mode!\n");
    target_index = robot_index;
    if(driving_mode == 0) { driving_mode = 1; }
    else { driving_mode = 0; }
```

Fig. 32.

A piece of code for distinguishing between the type of target water target or land target is also present. This small code first checks if world[target_index] is T or t. If world[target_index] = T then coming_from becomes O otherwise if becomes . After doing this it just replaces the world[target_index] with R (the character for robot) and world[robot_index] with coming_from meaning that depending upon T or t after reaching the targets the robot will replace them with O or   respectively.

VI. Conclusion

In this paper, we proposed the design of our new rescue robot. It is designed to drive in complicated debris environments, drive in water and to present useful images for the scenario. Sensor fusion of sensory input allows autonomous system to bring together inputs from multiple ultrasonic sensors, lidars to form a single model or image of the environment around a vehicle. Future work focuses on development of the method that can steer Robot model in 3D rubble environment, verification of its steering in different scenarios, and development of convenient user interface for rescue teams.

```
else if(world[target_index] == 'T' || world[target_index] == 't') {
    if(world[target_index] == 'T') {
        coming_from = 'O';
    } else {
        coming_from = '~';
    }
    world[target_index] = 'R';
    world[robot_index] = coming_from;
    rescued++;
    return target_index;
```

Fig. 33.

Furthermore, we are also considering the packaging of the system so that it is portable and simply transportable to the disaster location.

REFERENCES

[1] Sara R Hedberg. Ijcai-03 conference highlights. *AI Magazine*, 24(4):9–9, 2003.

[2] Robin Murphy, Satoshi Tadokoro, Daniele Nardi, Adam Jacoff, Paolo Fiorini, Howie Choset, and Aydan Erkmen. *Search and Rescue Robotics*, pages 1151–1173. 01 2008.

[3] Hannah Ritchie and Max Roser. Natural disasters, Jun 2014.

APPENDIX

It must be noted that all team members worked equally in the project as well as the documentation. Our group initially had 6 group members but Syed Aizaz Ali Shah and Rohail Usman left the group. Our group was comprised of 4 group members. The following is the workload distribution for documentation:

- **Dawar Zaman:** Implementation Phase, Conclusion,Design Phase III
- **Abdullah Zafar:** Design Phase I, II, IV,Requirement Diagram, Code explanation
- **Syed Muhammad Saim:** Abstract, Introduction, Applications of Rescue Robot, Use Case Diagram, Activity Diagram
- **Mustafa Touqir:** Introduction, Requirement Diagram, Constraint Diagram, Block Diagram, Design Phase I.

Every group member wrote 2.5 pages and was involved in other sections as well.

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Zaman, Dawar	Hamm 16.07.2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Touqir, Mustafa	Hamm, 16.07.2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

ZAFAR, ABDULLAH	HAMM, 16.07.2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

Saim, Syed Muhammad	Hamm, 16.07.2021	
Name, Vorname	Ort, Datum	Unterschrift
Last Name, First Name	Location, Date	Signature