I [Jeremiah] chose to pursue this project in hopes to learn more about a field within my major, robotics. Programming a robot to perform specific tasks and learning how to problem solve, real and physical problems, have always been an interest of mine that I wish to turn into a skill. This project, I believe, was a great choice for me in the sense that I am going to be able to apply my skills in programming and code to make something move in the physical world, which I find very exciting. I have also decided to take this opportunity to treat my learning environment as a work environment, so that by the end of this project I will have an idea of what it’s like to work with others in the same field, as well as other disciplinaries i.e., mechanical engineers.

**Goals/Objectives**

## Base Goals

The purpose of this project is to assist in the inspection of University of Central Florida’s storm drains. To iterate, the goal is to build a functioning robot which can accomplish a set of objectives, such as inspection, evaluation, and an integrity check of the storm drains. Being that the conditions of the storm drains are wet, made of concrete and metal, and are 20 feet below the ground, there are restrictions that need to be worked around and solved. As for must haves, the robot will be tethered, most likely by a steel cable for recovery if needed. Also, since the robot will be operating underground, a wireless connection is not possible and will instead be connected via ethernet cable. Due to the conditions that the robot will be operating in, good lighting is needed which in return will allow for good visibility for any cameras.

In order to accomplish said base goal, there are other objectives that should first be achieved. This includes the robot’s basic functions, such as vision, driving, location tracking, easy maintenance and durability. These objectives will be approached with much consideration and thought in order to deliver a high quality, practical, efficient and finished product.

In order for the robot to maneuver and drive in the storm drain, a tank drive is going to be implemented. In approaching this objective, there are other factors that are being taken into consideration, such as traction for the wheels. To successfully have the robot drive around in the conditions of a storm drain, wheels with high traction are going to be necessary. This will allow the robot to move on slippery surfaces and in mud, depending on the condition of the storm drain. In addition, the robot will not only move forward, but will also be able to maneuver in reverse as well. By accomplishing this small list of requirements, the robot would be considered as successfully driving.

The next objective is to implement a source of vision for the robot and the operator. To achieve this, a camera will be on the robot, relaying video feed to the user's computer. This will be done through the ethernet cable that is mandatory for the robot. The video feed will then both be saved on a local hard drive and streamed to a website where it will be saved on the server. Accomplishing these two tasks will make the vision objective a success.

As for location tracking, there are several methods as to how it’s going to be done. Although some methods are more advanced than others, there are some rudimentary approaches that redeem themselves through efficiency and practicality. Location tracking is considered as a success when the operator can accurately pinpoint the location of the robot from above ground, with an error margin of about five feet.

The robot will be designed for easy maintenance, allowing for the operator to easily fix or replace any parts. To accomplish this, the robot will have components that are soldered into the robot, these parts are not meant to be meddled with or replaced and are designed for longevity, such as the NVIDIA Jetson. If these components were to be replaced, opening the entire robot would have to be required. Components that would be maintained include wheels, camera, batteries, lights and the possibility of extra sensors. In addition, as requested by the stakeholder, these parts, such as lights and wheels, are to be easily replaceable and found with ease. To consider easy maintenance as a success would include the operator being capable of replacing these different components with ease.

To achieve the objective of durability, the robot will be soldered airtight, minimizing any cracks and openings within the robot’s frame and body. This will allow the robot to be submerged in water, protecting the onboard electronics and components. For durability to be considered a success, the robot will have to be water resistant, negating any water from touching any electronic components, and shockproof, having the robot be able to withstand any sudden drops or impacts.

**Requirements**

The robot will require several types of hardware such as, cameras, controls, circuit boards, lights, motors, tether, power supply and a cooling system. These will be designed and provided by the mechanical engineers who are working alongside us during the manufacturing of the robot. As for the computer scientists, several pieces of hardware will be needed in order to successfully program the robot to perform its set tasks. One of these required pieces of hardware will be the NVIDIA Jetson Nano Developer Kit, which is already equipped with computing power to run AI workloads, which can be useful in developing the final version of the robot. With the NVIDIA Jetson Nano alongside with other sensors, the robot may be able to perform its tasks more efficiently, such as detecting cracks, cave-ins, and hazard conditions, allowing users to effectively complete their tasks within the storm drain. For the mentioned sensors, laser mapping, ultrasound sensors, and light sensors may be used in order to create a more efficient robot that can detect issues that the user may not see or realize.

**Broader Impact**

With the completion of this project, its applications can help not only the university, but also cities who are prone to being rained on or flooded. The robot can be used as a replacement for a person, as conditions in storm drains can get risky and dangerous from gas leaks, cave-ins and wildlife. This robot can increase the efficiency of many inspections done by cities or construction companies, saving them time and money as well as lowering the risk of injury for laborers. The application of this project doesn’t stop with the inspection of storm drains though, the idea of completing a risky task using a controlled robot can help in other fields in the workforce i.e., construction.

The success of this project can bring to light the significance that computer scientists and engineers can be beneficial to society in a way that makes life easier and safer for everyone. People will see how the innovation of science and engineering can make a brighter future for the world, and advance society to a new level of efficiency and safety. As this innovation ripples into the future, inspired people will want to partake in the pursuit of this brighter future, creating more and intuitive engineers and scientists.

**Legal, Ethical, and Privacy Issues**

All parts and components that are to be used within the robot will be either purchased from retailers, such as the NVIDIA Jetson Nano, wheels and electronics, or manufactured, such as the robot frame and body. These components will then be tested to ensure that no misuse will occur by the operator or any user while also upholding the integrity of the product retailers.

The misuse of this robot can lead to legal issues, as it is probable that this robot will be equipped with a weapon in order to fend off any wildlife it encounters in the storm drain, and it will be equipped with cameras, lights and sensors. The misuse of the weapon can lead to serious injury or fatality. Also, if this robot were to be inserted into a storm drain in which it is not allowed to be in, it can be misused as a reconnaissance robot, where trespassing will occur, and as the robot is equipped with lights, cameras and sensors, a breach of privacy will occur as well. The issues listed from misuse can lead to legal action and should be avoided by the operator in order to uphold the integrity of the operator, and the component manufacturers.

## Sensors

There is a possibility that the robot will be built with additional sensors, some which assist in the detection of hidden objects and room mapping. This would allow the user to map out the space in front and around it, revealing details that may not be seen by the camera. The abuse of these sensors could result in breach in privacy, especially if the robot trespasses into an area that it is not permitted to be in. Privacy laws will be broken and result in legal action.

Geographical Tracking

One of the major objectives in this project is for the robot to track its position inside the storm drain, so that the surface can be marked and indicate where the said obstruction is located from the surface (Figure x.xx). There are a handful of possible ways of doing this, but most exceed the allocated budget, or are too complex.

A picture containing iPod

Description automatically generated

GPS

An idea that was brought up, but immediately rejected, was the use of a Global Positioning System (GPS). GPSs are found in many appliances and applications, such as cellular devices, computers, satellite phones, and more. Originally, the use of a GPS was for military use, but has then been extend to commercial and public use. The way a GPS works, is by taking advantage of the many satellites that orbit the Earth and use triangulation to pinpoint the location of the device using GPS (Figure x.x1). As to why this idea was rejected, it’s because for a GPS to work, it has to be within the sight of multiple satellites. Since the environment in which the robot will operating in is underground, there is no line of sight for the satellites, making this idea unsound.

A picture containing object, first-aid kit

Description automatically generated

Ultimately, this idea would have been the most efficient and practical method of tracking the robot’s position while in the storm drain. Unlike the other ideas, the complexity and implementation of this idea would have been practical, easy and useful, but because GPS requires a direct line of sight between the robot and satellites, this idea can’t be used.

It is possible to create a makeshift GPS for the robot, but that would require additional components that would have to be separate from the robot, as well as requiring financing from the budget.

Radar

The use of radar is within the scope of the project, it would allow for tracking distance traveled in an accurate fashion, as well as be used to detect any obstructions inside the storm drain. The way a radar functions is by transmitting signals with an antenna and receiving the bounced signal to determine the distance, size, and location of an object (Figure x.x2). By using radar, another object may have to be lowered in alongside the robot, to act as a starting marker for the robot, but detecting cave-ins and obstructions would be easy, and efficient on time.

A picture containing object, gauge

Description automatically generated

Radar would a good option for tracking distance traveled, but it would require more financing, and advanced code in order to properly use the radar. Additionally, more components would be needed in order to construct the radar, increasing the complexity and maintenance of the robot. In addition, there are several conditions in which the radar would not function properly, or to its full potential. The first condition would be that the storm drains aren’t always leveled, meaning that if the robot was traversing the pipe with deviations in terrain, there is a chance that the marker that was placed as a starting point, may not be detected by the radar due to difference in elevation, thus ruining the point of the starting marker. Secondly, there are slight turns within the storm drain pipes, meaning that the starting marker may be out of sight, and thus ruining the point of the starting marker again. Thirdly, any obstacles that are inside the drain can possibly prevent the sonar from functioning.

Another use of radar is Ground Penetrating Radar (GPR). This allows for finding objects located underground, from the surface. Using radar in this fashion would be practical in finding cave-ins and obstructions inside the storm drain, but it doesn’t allow for the user to evaluate the conditions inside the storm drain, along with the fact that pipes may run under buildings in which the user won’t be able to properly follow the layout of the storm drain pipes. To further reason as to why GPR isn’t a good idea is the fact that obtaining a GPR is expensive and would require funding over the designated budget.

Overall, radar is a good idea as a concept, but it adds onto complexity as well as financing. There are also many things that can inhibit the radar from functioning, meaning that the radar would need specific or perfect conditions for it to operate as expected. Furthermore, with the conditions inside the storm drains, using radar would not be the most optimal or practical option to track the robot’s traveled distance.

Sonar

A similar idea that was considered, was the use of Sonar, which is very similar to radar (Figure x.x2). Instead of radio waves, sonar uses and manipulates sound waves at a high frequency in order to locate any objects. This concept is similar to how bats use echolocation, in which a pulse of sound is released, and a receiver then calculates the position and distance of an object based on the time it takes for the echoed sound to return. Just as the radar had its specific conditions in which it would function properly, sonar has conditions that mirror the problems of radar, such as using a starting marker. Furthermore, sonar would be more complicated and complex due to the fact that sound waves will bounce off the surrounding walls more inside the storm drains, and overwhelm the receiver, thus inhibiting the sonar from operating properly.

In this case, using sonar would be inconvenient to implement, especially in the conditions that the robot would be working in; it would require additional funding and advanced math techniques to properly map out and understand the incoming data. Overall, sonar seems to be one of the less likely ideas, due to its inconvenience and complications.

Tether Length

A rudimentary idea that is taken into consideration is to just measure out the length of the tether once the robot discovers any cave-ins or obstructions inside the storm drain. In order to do this a few concepts were brought up that could help with measuring.

One concept discussed was to place the tether on a reel and calculate the length of tether used based on how many rotations the reel has gone through. One reason as to why this concept was rejected was because of its inaccuracy. By pulling the tether off the reel, the radius would constantly get smaller, thus requiring a remeasurement of the radius in order to calculate how much tether has been used.

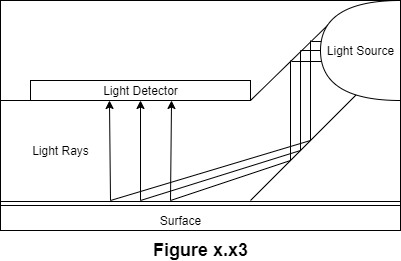
Another concept, involving the reel, would be to mark up the tether to indicate a unit of length, feet, yards, meters, etc. and then have the tether run through a sensor that would count each mark, resulting in a total distance traveled. This idea proved to be practical in the sense that it is easy to implement, cost efficient and practical. By having a marked tether run through a sensor, there is no room for error and would also be easy to program. The complexity of this concept is also very low, allowing for quick use and measurement.

A few concerns that arose for both concepts are the fact that the tether can get tangled or misaligned when reeling in said tether. Although the robot and tether system will be handled by professionals, there is still a possibility that the tether will get tangled in the reel, in which it would add more time to the operation to untangle. Another concern is the fact that the robot would need extra torque and power in order to pull the tether off the reel. Realistically, the robot should only have enough torque and power to pull itself, and not have to unwind the tether from a reel.

Overall, the second concept is effective and has a high chance of being the main method of measuring distance traveled. The pros, in this case, outweigh the cons and allow for simplicity to take place when implementing this system. It contains practicality, simplicity and efficiency when achieving the goal of measuring distance traveled, by the robot.

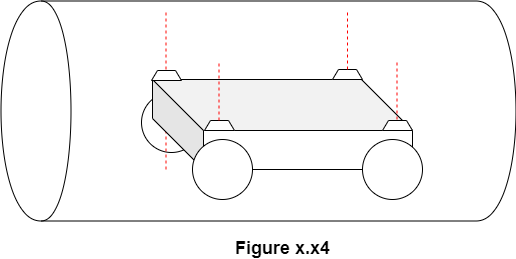
Optical Mouse Sensor

One such solution to tracking the distance traveled by the robot is the use of an optical mouse sensor. An optical mouse sensor uses a light source and a light detector in order to calculate movement in relation to the surface it’s on (Figure x.x3). Like the sensor underneath a computer mouse, the robot can use this sensor to achieve the same functionality. By using this sensor, the robot can track what direction it’s going, as well as how far it’s gone.



In order to successfully implement this idea, several optical sensors will have to be used, since just having one sensor will not be enough. These sensors will have to be placed all over the robot, locations that track the robot’s actual movement in three-dimensional space. The level of complexity to program this would be moderate since it would only involve vectors and matrices in three-dimensional space.

To properly execute this system of optical sensors, there will have to be at least eight different optical sensors (Figure x.x4), four on top of the robot, and four underneath the robot. This implementation would be effective because some sensors may be blinded by mud, or anything that can be found in the storm drains, and by having more than one sensor, there will be other optical sensors to take its place along with the fact that by having more sensors, the more accurate the calculation of distance travelled will be.



In total, using optical mouse sensors is a good idea, but there are some inconveniences that make it difficult use this idea. One such problem is the wide range of optical sensors, ranging from low quality and basic sensors to high quality and advanced sensors. Each of these tiers will of course have a different price range, making these sensors either very cheap or extremely expensive. Another problem is that most optical mouse sensors require a flat surface in order to properly function. Inside the storm drain, the surface will not be smooth, instead it will be wet, rough and made of concrete. What this means is, the light being used by the optical mouse sensor will be diffused or refracted, resulting in the sensor not being able to receive the bounced light. Depending on the quality of the sensor and its source of light, this problem can be overcome, but may be impractical and inefficient with the budget.

Wheel Rotation

An obvious idea that is highly leaned towards, is the calculation of distance travelled based on how many times the wheels on the robot have rotated. This concept is basic, practical, and not complex at all. Even though it may be the most basic idea, it redeems itself though its effectiveness and efficiency. By adding a sensor to count how many wheel rotations have occurred, a program will have to just use an equation, with wheel diameter taken into consideration, to find the total distance travelled by the robot.

Though this method has a high chance of being used, there is one major concern that is taken into consideration, the problem of wheel slippage. Wheel slippage will most likely occur within the storm drain, especially with the conditions the robot will be operating in. Slippage occurs when the wheels on the robot can’t get enough traction on the surface it’s operating on, and so rotation of the wheel occurs but no directional movement is achieved. This can be detrimental for the goal of tracking distance because, based on this method of tracking, any excess wheel rotation will contribute to the totality of wheel rotations.

By having extra wheel rotations that do not contribute to actual distance traveled, the distance traveled may be misrepresented and misinform the user. Being that this robot is being used and operated to find any cave-ins, and or obstructions in a storm drain underground, its severely important that tracking the robot’s location must be accurate so that operations above ground can take place in an efficient manner.

Methods of how to prevent slippage have been narrowed down to the type of wheel to be used. Several wheels have been discussed, and only a select few characteristics have been chosen to be mandatory.

One such characteristic is the fact that the wheel should be heavy in order to allow the robot to stay stable on the surface of the storm drain. Also, in order to make the wheels heavy, the wheels will most likely be made of a solid material, with minimal air. This is beneficial, such that by having solid wheels, they will sink in water and not float. Furthermore, by having solid and heavy wheels, the robot can remain stable when maneuvering.

Another characteristic is treading on the wheel. Treads are useful because they allow for the wheel to stay in contact with whatever surface they are operating on top of (Figure x.x5). In addition, treads are designed to repel water from the contact surface, which in this case is very beneficial for the robot and for the environment the robot will be operating in.

A close up of a logo

Description automatically generated

Overall, using wheel rotations as the method of tracking distance traveled is the most appropriate approach for this project. This method contains minimal problems, in which can easily be corrected, if errors were to occur. In addition, this method proves to be most practical, intuitive and budget efficient way of tracking distance traveled by the robot.

Communication Methods

An important component for the robot to operate properly is its method of communicating with the computer and user. There are a handful of ways that communication can be achieved, such as ethernet cable, and wireless connections. There are various types of wireless connections that can be used in order to establish a connection between the robot and computer. Some examples of these wireless connections include radar, Bluetooth, and Wi-Fi, which are all essentially radio waves, just being manipulated in various ways.

Communication: Radio

For the first method mentioned, radio waves operate by transmitting electromagnetic signals from one location to another, without the use of a direct connection, such as wires. There are two important pieces of hardware that allow radio to work, the transmitter and the receiver. The transmitter sends out the radio waves, while the receiver receives these signals (Figure x.x2).

Since radio waves fail at penetrating rock and reaching underground, a transmitter and receiver will have to be made and lowered into the storm drain along with the robot. This transmitter would then be tethered to the computer in order send and receive data from the robot. Radio waves also travel at the speed of light, meaning that the transmission of data would be fast and in real time. This would allow for real time video feedback and let the user control the robot a lot easier, without a latency in video feedback. In addition, radio waves can reach long distances, so long as the transmitter and receiver are strong enough, and therefore allow for the user to potentially drive the robot from one end of the storm drain, to the other. Overall, using radio signals to control the robot is a possibility, but implementing the hardware and programming it may be tedious.

There are some negative sides to using a wireless connection though. As previously stated, an extra piece of hardware will be required in order to properly transmit and receive data to and from the robot, which in turn may take more resources from the allocated budget. Secondly, wireless connections always have the chance of being intersected or negated, by other signals and noise. What’s meant by noise is, the irregular fluctuations that can accompany a signal and obscure it. Another negative aspect is the fact that the wireless connection can be blocked off by objects that are thick and dense enough, such as concrete. If the robot were to get behind rubble, or go through a narrow passageway, driving out of sight of the transmitter, it’s possible that the robot will not be able to send or receive any signals, thus leaving it stuck with no way of controlling the robot and immobilized. If this were to occur, a person would have to be sent into the storm drain to either retrieve the robot, or put the robot back in range of the transmitter, defeating the purpose of the robot, to inspect the storm drain in the place of a person.

Communication: Ethernet Cable

The other form of communication comes through the use of an ethernet cable to transmit data and signals, to and from the robot. There are several benefits by using an ethernet connection over a radio connection, while at the same time several deficiencies that can make the wired connection seem like a second choice.

Even though the wired connection may not be as fast as radio waves, the speed at which ethernet cables work, is enough to transmit data in a timely manner. By using a Local Area Network (LAN) connection, transmitting data can reach speeds up to 100 Megabits, or 10 Megabytes per second, which is a fast connection; one that is enough to transmit video in what can seem to be real time. LAN is a computer network that connects several computers within an area (Figure LAN), which in our case is just the robot and computer.

A screenshot of a cell phone

Description automatically generated

To list the few advantages of using an ethernet cable over a wireless connection, the physical cable should be taken into consideration. By having a wired connection, it’s less likely that noise will interrupt the video feed and cause discrepancies. This will allow for a higher quality video feed and make it easier for the user to assess the conditions inside of the storm drain. Furthermore, by having an ethernet cable, it would be difficult to lose connection to the robot in the storm drain. The only way to lose connection would be if the ethernet cable was severed or became unplugged from the robot.

On the negative side, having an ethernet cable run between the robot and computer can add some complications. One obvious example, is the fact that the cable can get tangled by debris inside the storm drain, making the operation done by the robot more difficult. Another inconvenience is the limited distance the robot will be able to travel, because of the finite length the ethernet cable has. Unlike a wireless connection, the robot will be required to drag the extra weight of the ethernet cable. This means that the robot will require extra power and will vary depending on how long the cable is. As for another problem of using a wired connection, the robot may end up being submerged in water. This would require the cable port on the robot to be waterproofed because the said port could be a breeching point for water. This is a problem that has been discussed in which several solutions have been conceived. One such solution is to add a sleeve over the ethernet cable to keep the water out and another solution would be to use rubber stoppers on the ports. These two solutions are the best proposals, based on practicality.

Overall, between wired and wireless, wired maybe the best option. Reasons to support this is the want for quality over quantity. Even though a wired connection may not be as fast as a wireless one, the quality of video will be far better than that of a wireless one. In addition, by using a wired connection, extra hardware such as a transmitter and receiver will not be required for the robot, thus saving resources in the budget.

Controller Method

In order to operate the robot, two methods have been brought up, computer keyboard and console controller. These two methods of operating the robot have been discussed and agreed that both will most likely be used interchangeably. By having two methods of controls, the user can choose how they would like to control the robot while also having a second method as a failsafe, just in case one method becomes unusable.

Controller Method: Keyboard

In comparison, the keyboard will have more buttons than that of a console controller. This difference in button count could be beneficial for the robot in respect to functionality. By having more buttons, more things can be programmed onto the robot, making it easier for the user to execute desired tasks. On the other hand, by having more buttons, the user may get overwhelmed or confused and mistake one button for another, resulting in an undesired action. Realistically, the robot won’t be programmed with many automated actions based on the requirements. Also, if using a console controller is to be included, the number of buttons being used on a keyboard should match the number of buttons on the console controller. By doing this, the functionality between keyboard and controller will be one to one and will allow for the user to swap between both methods without having to worry about extra actions and buttons.

Controller Method: Console Controller

As for using the console controller, the user may feel more comfortable operating the robot this way because it can give control over how much power is being outputted by the robot, such as movement speed. Also, console controllers are very intuitive and are widely used by many people, meaning that it could be easier for the user to operate the robot with the controller. A console controller usually comes with two joy sticks, one for directional steering, and another for camera angle. By allowing the camera on the robot to be controlled by one of the joy sticks, typically the right one, inspection of specific areas inside the storm drain can be made with ease.

Between keyboard and controller, keyboard can will be more practical if different autonomous actions were to be programmed into the robot. By having more buttons, more actions can be programmed and thus allowing for more functionality on the robot. In respect to console controllers, controlling the robot would be a lot easier because of the intuitive design. The controller comes equipped with pressure sensors, meaning that more control is granted to the user. The user would be able to control how fast the robot is moving, and control where the camera is facing, independent from the rest of the robot. This means that only the camera would have to move, instead of moving the entire robot in order to change what the camera sees. Both methods have their pros and cons, but by having both implemented as methods of operating the robot, users can choose which approach they find more comfortable.

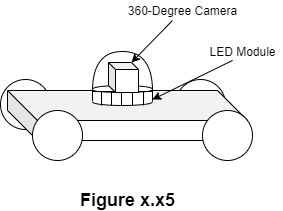
Lighting Method

For lighting, several ways to illuminate the dark conditions inside the storm drain have been thought of. Multiple methods and designs have been discussed, such as a 360-degree light around the robot, or having a light attached to the camera so that whatever the camera is pointing at is illuminated. The type of lights that are most likely to be used are Light Emitting Diodes (LED) because of how powerful they can be without the need for much power.

Compared to other light sources, LEDs are not only the most efficient, but they also emit small amounts of heat, which is optimal for the robot in a heat output aspect. As for the downside of using LEDs, the price range of these lights are a lot higher than other lighting options. The reason for this is that LEDs get more expensive depending on their power, or how many lumens they can output. A lumen is a unit of measure for how much light is emitted per second. This means that depending on the requirements of this project, if the user is to be able to see far, through the video feed, the price of lighting can get expensive. In addition, there is a possibility for multiple light sources on board the robot, meaning that in order to have acceptable lighting, more resources may have to be allocated for lights in the budget.

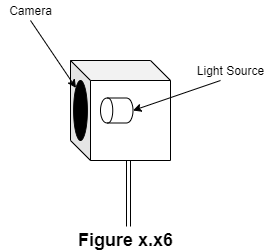
Lighting Method: 360-Degree Light

As previously mentioned, there are two approaches for equipping the robot with lights. The first idea is to put lights all around the robot, encompassing it in a 360-degree ring of lights (Figure x.x5). By doing this, the area around the robot will always be lit up, allowing for a uniform distribution of light. In addition to this idea, if the project advances with the use of a 360-degree camera, this lighting method will be useful and practical. Overall, this lighting system seems to be the most efficient method for lighting up the storm drain but may become expensive depending on how many lights are required to achieve the 360-degree area around the robot. Secondly, depending on the power of the lights, the price range will rise alongside the light’s lumen output.



Lighting Method: Light Attached to Camera

The other approach for lighting is to attach a light onto the camera. By doing this, a light will always be shining wherever the camera is looking. Taken into consideration, this can be the most cost-efficient option since only one light would be required. In addition to cost-efficiency, it may be much cheaper to get a powerful single light rather than multiple lights. There are some downsides to this approach though, such as having a singular light source. If this light source were to go out, the user would not be able to operate the robot properly because they will not be able to see what they’re doing. This occurrence would practically immobilize the robot. Secondly, by having a single powerful light, more heat may come from it, resulting in the need for heat dissipation methods.



Both methods are useful in their own respect, each having their pros and cons, but depending on what is desired on the final product, one of these methods will have to be chosen. Between both approaches, having a 360-degree light that can cover all angles of the robot would only be useful for a 360-degree camera. In addition, if one of the light modules were to go out, there would be other light sources on the robot that can illuminate the storm drain for the user. The downside to this method would be the price of all the lights, cumulatively. On the contrary, having a light source be attached to the camera would efficient in the sense of simplicity, but since it’s the only source of light it should have a high lumen output.

Cameras

On the topic of cameras, there are several types of cameras that can be used in order to capture video and stream it to the user. There are only several cameras that can operate in the conditions the robot will be going into. Some examples are Go Pros, Canon IVY, Olympus Tough TG, Nikon Coolpix and Kaiser Baas. These are a handful of cameras the robot can operate with, but depending on the minimal requirements, some options will be more practical than others.

One brand of cameras that are being looked at are Go Pros. These cameras are well known for their durability and quality. They are waterproof, meaning that submerging this camera won’t be a problem. In addition, Go Pros are also shock proof, which means that they can be dropped from high heights or absorb a large amount of impact and not get damaged. These two qualities are great for the conditions in which the robot will be operating in and are practical.

Along with Canon, there are other previously named brands that have developed their own waterproof cameras. Although this is the case, the go to for effective waterproof cameras would have to be Go Pros, because they are known to be taken into water along with other extreme conditions. In the market, Go Pro seems to take control because of their quality and durability.

There is one setback to using Go Pro as the camera, it costs much more than the other options. As of right now the GoPro Hero7 is at $200, whereas its competitors such as Canon, range from $80 to $100. This means that if multiple cameras are desired for the robot, more resources will have to be allocated for the cameras within the budget, especially if the option of Go Pro is selected.

Implementation: Single Camera

As for implementing cameras, there are a several approaches on how these cameras will be used. One such method is the use of one camera, which can be swiveled and turned by motors on the robot (Figure x.x8). By having this option for the camera, the user can easily look around and inspect anything they want within the storm drain by controlling the swivel and or axle.

A picture containing text

Description automatically generated

There are several negative sides to using a singular camera though. One being that the user would have to manually turn the camera and adjust the axle and swivel individually in order to look at what is desired. This could be time consuming for the user and slow down the operation, or even be frustrating for the user. Overall, this concept is rudimentary and works, it’s simple and practical, but there are other approaches that can fix these small problems for the user.

Implementation: Multiple Cameras

One other implementation of cameras is the use of more than one camera. This concept requires at least two cameras where one camera would be facing the front and the other facing the rear. This allows the user to easily control and know what is around the robot. To increase this effect, more cameras would be required in order to see in all cardinal directions. By having more cameras facing in multiple directions, the user won’t have to adjust the robot in order to face anything in the storm drain, instead the user would have to just adjust that specific camera.

This concept has its pros, such as seeing more and having multiple backup cameras, but also has its cons. One such con would be the implementation of these cameras, having the processor communicate with all of them at the same time. In addition, programming all the cameras would be a hassle and could lead to complications. Furthermore, depending on the type of camera that will be used, the price of all multiple cameras can severely raise the cost of the robot, meaning more money would have to be allocated to the cameras.

Implementation: 360 Degree Camera

One big solution to the problems made by two previous ideas would be the use of a 360-degree camera. This camera would allow for the use of only one camera module, while also encompassing a 360-degree field of view for the user. This takes both previous methods into one because the user wouldn’t have to turn the robot or camera in order to view something, and the camera would already provide a 360-degree view, meaning that there would be no necessity for more than one camera.

Streaming and Saving Video Feed

It is crucial that the robot has a camera, for the user to see what is inside the storm drain. In order to achieve this task of having the user to see inside the storm drain, camera feed will have to be fed from the onboard camera, to the robot operator’s computer via ethernet cable, then to the online server via Wi-Fi. This task is straight forward but has its complications as well.

Implementing a streaming functionality will be done by having the operator’s computer connect to a wireless source then relay the video feed to the website’s server. In result, other computers who have access to the website will be able to view what is being fed to the server and receive the stream as well.

Streaming and Saving Video Feed: Issues

There are several problems in streaming from the robot operator’s computer via Wi-Fi and streaming in general. The stream will only work on the conditions of a solid and strong internet connection, if there is no internet connection then the stream won’t work. If the internet connection is weak and flimsy, some data and packets might get lost in the streaming process. There may even be a case where the operator’s computer may not be in range of any Wi-Fi hotspot and thus inhibiting the operator’s computer from streaming at all. Then there’s the issue of video feed quality. If high definition video feed is going to be uploaded to the server, a good and stable connection to the Wi-Fi is a must in order to maintain a high quality stream, otherwise the next best option would be to stream a low quality video feed to the server, which might not be an option since whoever is viewing the stream needs to be able to see properly, thus requiring a high definition video feed.

Streaming and Saving Video Feed: Solutions

In order to resolve these issues, there are several solutions that can be pursued. One such solution is to implement a Wi-Fi range extender. What this does is, it relays the Wi-Fi signal from the wireless source, thus extending the range in which the Wi-Fi source can reach. This application can then be amplified by having multiple Wi-Fi range extenders that relay the wireless connection from one module, to the next, until a connection between the wireless source and computer can be made. By doing this, the operator’s computer will be able to connect to the UCF Wi-Fi from a reasonable distance, so long as the operations are taking place on campus or if the operations are in range of an accessible wireless hotspot. Though this option is viable, it does incur a fee to the total budget of the project, and to get a high-quality range extender for the operator’s computer, it would mean spending about one hundred dollars or more per unit.

Another option is not so much a solution, but an alternative, with this alternative being that the video feed is saved on a local hard drive then uploaded to the server later when a stable connection can be made. By having an onboard hard drive that saves the video feed while also streaming, the integrity of the captured data will stay intact and also retain a high quality for later viewing. For example, if the connection to the Wi-Fi were weak, there may be some packets in the stream that don’t get sent out to the server, thus resulting in a patchy or cut up video stream, but by having an on board hard drive via computer or robot, the integrity of the video will be saved and later uploaded to the server. It may not be live on the server at that point of recording, but it does redeem the problem of a faulty wireless connection, if conditions were to be that way.

Electrical

There will be multiple components on the robot that will require a source of power, such as the camera, lights, motors, NVIDIA Jetson Nano and possibly peripheral sensors. In order to properly supply power to these different components, an onboard battery will be installed instead of using a power cable. The reason for using a battery over a power cable is due to the required voltage needed to power the robot. In order to power the robot via cable, the required voltage would need to be dangerously high for electricity to reach the robot, especially at lengths of 250 feet.

Electrical: Issues

In addition, if a power line were to be used alongside the ethernet cable, there is a possibility that the electromagnetic field could interrupt the data that’s being sent from the robot to the computer, or even inhibit the ethernet cable entirely. This would mean that no data would be sent from the robot to the computer, and vice versa.

Furthermore, by using a high voltage cable, there is the possibility that the cable can wear and tear, resulting in an exposed wire, with high voltages. This is extremely dangerous as the conditions in which the cable will be exposed to are wet, meaning that electricity will be able to travel from the source, so long as it is in contact with a wet surface. This could also result in the destruction of the robot, costing a lot of money and resources, along with creating a problem in recovering the robot itself. Overall, by using a cable that would run a high voltage within it, many possible problems arise, some that can be fatal.

Electrical: Benefits

By installing a battery on the robot, there are several advantages. One obvious advantage is the lack of a 250-foot power line, running from a power source to the robot. This will allow for a safer environment for the operator since there will be no high voltages that could potentially harm the operator, and equipment. A battery also allows for easy maintenance upon the robot because the battery will be able to be swapped out by another, and then recharged for later use. Overall, the use of a battery, rather than a power line, makes the whole operation more efficient and effective in respect to safety, which is something that should be prioritized when developing this project.

Additional Components

As for the several electrical components on the robot, each will require a certain amount of power in order to operate properly. These different components include the motors, lights, cameras, and sensors, which can drain the battery if the required power input isn’t calculated properly. In addition, the robot will require a decent amount of run time in order to complete its varying tasks, this means that the battery should have enough output to supply power to all the different components while also being efficient in its output. The different components themselves should also be efficient when it comes to using battery power in order to extend the run time of the robot.

Application vs Website

In order to view the stored data and footage on the servers, there are two distinct ways in which that can be achieved. One being a web-based application, and the other being a website. There are some key differences between these two methods of viewing server data. One major difference is the fact that a web-based application allows for a user to manipulate data, such as adding, editing and deleting data, whereas a website only allows for a user to view and read the stored data. In addition, a web-based application allows the user to access hidden data.

Website

Using a website to store data is very basic but can be effective if used properly; it mostly depends on how the data is going to be used, and what it’s going to be used for. In example, if the video is going to be used for assessing the internals of a storm drain, as well as logging the condition and integrity inside of it, there would be no need to make any deletions on the server, as it would be important to keep old data to predict what faults might occur in the future.

Website: Benefits

In addition, websites would be easier to develop and deploy compared to a web-based application because, the amount of work and scope of a web-based application is broader than that of a website. In developing a website, there would be no need to implement any additional features, such as editing or hiding any data, instead users who use the website will be able to view, and only view, whatever is on the website. Also, if an update were to be made on the website, small changes in the source code would only be needed, making it fast and convenient to update.

Website: Issues

In using a website, it of course has its downsides. One of these downsides is the fact that all data is most likely to be presented on a single page, and the fact that a website can crash means that there’s a possibility that all data on the website won’t be available if a crash were to occur. Also, the fact that websites are static means that if a minor change to any data, such as changing a name was needed, altercation to the source code would be required; doing this for every minor change could become tedious and inefficient in large quantities of changes because it wouldn’t make sense for every new update to require editing the source code again. Having this occur could be considered one major problem of using a website to present and store data.

Web-based Application

To use a web-based application may be the most optimal approach for the storm drain robot project. Reasons as to why this may be the case is because, users will need to be able to access data from the servers, then should also have the ability to add, edit and delete any data, deemed necessary. In addition, since users will have the capability to manipulate data stored in the server, it is mandatory that a two-step authentication process be implemented; this will secure any data that shouldn’t be accessed by anyone who isn’t a part of the staff.

Web-based Application: Benefits

Furthermore, by using a web-based application to store and present data, more features will be available. This would include the basic editing of a file and logs, while also extending to more advanced features such as organizing the data, deleting data and gaining access to hidden data, or hiding data. In addition, anyone who is using the web-based application and authorized, will be the only ones able to manipulate the data, meaning that there is a low risk of any data being tampered with. Finally, one key characteristic that gives we-based applications a huge advantage, compared to websites, is their capability of scaling. This means that the website can be altered in such a way that more data can be put onto the servers as well as be dynamic.

Web-based Application: Issues

Just like its counterpart, a web-based application also has its downsides. Several of these downsides include the fact that security isn’t always one hundred percent guaranteed, meaning that there is a chance that someone may gain unauthorized access to secure data, if the website isn’t developed properly. Another problem that can occur, if not carefully constructed, is the possibility of the web-based application only working on a specific web browser, rather than all of them. This can be an issue, as it should be readily available and accessible without having to take into consideration the type of browser being used.

Conclusion

Between these two options, its most likely that the project will proceed with the use of a Web-based application because of several reasons. One key reason is the fact that it would push for the development of the website to be more secure, as it would be handling important information and data. Secondly, the end users should be able to interact with the data, such as, freely editing any names of any files or logs, without having to edit the source code. Having this feature would benefit the user and anyone else who has access to this data because it would save them time and increase the operations time efficiency.

Cloud vs Physical Storage

As for storing video footage, there are two distinct methods on how that can be achieved. The first method would be storing the footage on a cloud, where it can be saved and downloaded from another computer. The other method is to save footage on a hard drive that would be installed on either the computer or robot. Given the scope of the project, these two methods are the only viable options to achieve storing data.

Cloud

As previously discussed, storing the data and video footage on the cloud is very convenient and practical. It allows other computers and machines to download and view the data remotely, without having to transfer any data physically. There are a handful of benefits of using cloud storage, but it also has its faults.

Cloud: Benefits

Cloud storage has many strengths, some that make it the best choice in respect to the task that is being done. For example, cloud storage is strong when it comes to security because an authentication process could be implemented onto it, allowing for authorized users to access the data only. Secondly, cloud storage can be accessed remotely, meaning that authorized users can gain access to data from anywhere. This is heavily convenient for workflow and efficiency in a company or project. Another strength that cloud storage has, is its cost effectiveness; depending on the size of the project, it would be optimal to use cloud storage, as it would allow for larger storage space per dollar.

Cloud: Issues

There are some downsides of course, using cloud storage. One such major disadvantage is that the computer or machine, would always need to be connected to the internet in order to properly send packets to the server for storage. If the internet connection were to be interrupted at all, the file could be corrupt or fail to upload. Secondly, downloading from the cloud could be slow depending on internet speeds and signal strength. This extends to an issue that the computer or machine would need an internet connection in order to view the stored data. Then, there’s a major risk of losing all data collected and stored because if an authorized person were to gain access to this cloud, they could potentially remove and delete everything saved on the cloud, resulting in the loss of valuable information and data. Overall, these disadvantages don’t completely weigh down this option, as it heavily depends only on a solid and secure connection to the internet; so, if this can be maintained and achieved, the process of storing and downloading data would be practical and effective.

Physical Storage

Compared to the other method mentioned, this method can be seen or used as an alternative, or to be more precise, a backup plan just in case the primary method doesn’t work. Through the use of physical storage, such as a hard drive, there are some benefits to it, compared to using a cloud storage; then there are some faults to using it as well.

Physical Storage: Benefits

Some benefits to using a physical storage to keep data and footage is the fact that it is a lot easier to store the data, without the risk of losing any footage, pieces of it or entirely. The data would be writing directly to a storage device rather than sending packets over an internet connection, wireless or wired. This option would also be most optimal for higher quality video, as it wouldn’t require the transfer of large packets to the server in order to achieve a higher video quality. In addition, having the data stored in a physical device allows for data to be separated, and not be collected in one location. This means that the loss of losing all footage and collected data over time will can be minimized.

Physical Storage: Issues

Some negatives as to using a physical storage is its lack in convenience and access. Due to the fact that the data would be stored in a physical storage device, it wouldn’t be readily accessible or available right away; instead it would have to be manually uploaded to the server, which could take some time depending on the length and size of the video. Secondly, it would take up time, making it less time efficient compared to saving the video to the cloud directly.

Conclusion

Overall, both of these options have their benefits and downsides, but the option of uploading directly to the cloud seems to be the most optimal and practical. Even though using a physical storage device would be much easier, it just isn’t as convenient as using a cloud-based storage. The risks that stem from using a cloud-based storage is its dependency on a stable internet connection, which can be worked around. Also, it would allow for one less item to take into consideration when taking inventory for the user. In addition to this, it would be secure so that no one can just access this data.

Process for Deploying Software and Updates

When creating software for a computer or machine, there arises the issue and concern of how to update the said software. In this project there are several methods in which updating the software can occur. Such methods include updating via downloading from an online source, such as a cloud, and the other would be directly updating the software via uploading to the processor via ethernet cable or a drive. These two options are the most well-known methods of achieving an update, as they are the most basic and practical way to update a piece of software.

Updating from the Cloud: Benefits

When updating software, it’s most convenient when its accessible online, via the cloud. Compared to its counterpart, which would be updating via drive, this method is the best with its ease in accessibility and convenience. It doesn’t require a physical drive to be passed around in order to update all machines using the software, instead all the machines can just access this update through the cloud from their location. This ultimately saves time and compared to a physical update, is most efficient. Furthermore, the fact that the update would be in cloud allows for scalability. This means that if more machines were to be made and used, such as from one robot to twenty, all twenty could connect to the cloud at the same time and receive the update together; there would be no queue for the update.

Updating from the Cloud: Issues

There are a few minor issues that arise when pushing updates from the cloud. One such issue would be the fact that the machine must connect to the internet in order to receive the update. This just means that the machine or computer must be connected to the internet first in order to update, then be used for operations after. Another minor problem, depending on the conditions of the internet and its accessibility, is its speed. Sometimes download speeds from an online source can be extremely slow, basically wasting time and becoming impractical. If conditions were to be like this, then it would be more efficient to just use a physical drive to update the software.

Updating from Drive: Benefits

Compared to updating from the cloud, there are some areas in which a physical drive has the upper hand. Some of these areas include download speeds; when a computer downloads from a physical drive, it is typically faster in comparison to downloading from an online source. This is attributed to the fact that the drive is put into the computer and can immediately upload the update. Another attribute that physical drives have is the fact that updates can be applied on the go. As long as the drive contains the updated software, it can be uploaded onto the computer or machine while in transit. This attribute is only beneficial in certain situations that inhibit the computer or machine from connecting to the internet.

Updating from Drive: Issues

As already pointed out previously, there are of course a few issues with updating via physical drive in comparison to downloading from the cloud. One such issue is as previously mentioned; it would take time to update each machine one by one via physical drive. Thus, resulting in inefficiency and wasting of time. Instead it would be better if all the machines could just download from the source of the update, at the same time. So, if the computers or machines were to be placed in a queue in order to receive the update, it would make the physical drive impractical. In addition to this, there’s the situation of having multiple machines, which then would require someone to deliver the physical drive for another machine to use; this in result just wastes time and energy for everyone.

Conclusion

In conclusion, between both of these methods, using a cloud to deploy software updates seems to be the most practical solution; especially if multiple machines were in need of it. The fact that an update can be deployed to many machines via cloud, gives this solution a valuable attribute and upper hand, scalability. By allowing multiple machines to download the update concurrently, time is saved, and efficiency is increased for the company and users.

Even though using the cloud to deploy software updates is the best option, there are some attributes from physical drive updates that could be used. One such attribute is the fact that the machine can update without access to internet. This is conditional of course, but if such a case did occur, having a physical drive with the update would be beneficial.

Ultimately, it would be best for the robot to have both as an option and be used interchangeably. Both solutions have their strengths and weaknesses, but if looked at closely, both complement each other. So, if possible, both options should be viable for the robot, and not be restricted to only one.

Robotics Simulator Tools (Extended)

Simulation tools are essential and very useful when it comes to testing out new code, especially before deploying any updates or fixes. Reasons as to why simulation tools are essential is because of how effective they can be in the development and production of the project. Not only would it allow the team to test their code and programs, it also saves a lot of time in the project by allowing flaws in logic and issues to show up within the simulation, rather than in the real world, after the robot has been uploaded with the code; which may have taken time to do.

Even though simulation tools are effective in what they do, for the project; there are a handful of different simulation tools to choose from. To choose the right one for the project is crucial as it will save time and increase efficiency in the deployment of the project and updates. To name a few of these simulation tools, there are V-REP, Webots, anyKode Marilou Robotics Studio and Gazebo. These are just a few simulator tools that can be used when developing software and code for robots. Each simulation software will be somewhat similar to another, but each have their own strengths and weakness that make each of them practical in respect to their scope of features. For example, its possible that one tool will have more data and information about a component, while the other one lacks this information. This issue can be found among all of the simulation tools, each lacking in something another software may have.

V-REP

Looking at V-REP (Virtual Robot Experimentation Platform), it is a simulation software that has the feature and power to allow the user to model, edit and program a virtual robot; essentially allowing the user to simulate their robot. In respect to the project, this tool is viable and would allow the team to benefit from running experimental code and builds in a simulation, rather than wasting time deploying new code to the physical robot.

There are also additional features that V-REP holds that make it more viable and appealing. One such feature is the fact that V-REP not only allows you to replicate your robot and code virtually, but V-REP also offers the simulation of sensors and any mechanisms your robot may have. This means that users can be expected to test their entire robotic system and run code against it. In addition, V-REP comes equipped with ready to use robots; this means that more testing can be done with the already verified code and allow for troubleshooting against new code. Just having these few components from V-REP would be beneficial and allow for the project to move forward at an appropriate pace.

Furthermore, V-REP is compliant with languages such as Python, which is going to be used in the development of this project. Secondly, V-REP is compatible with multiple platforms such as, Windows, macOS and Linux. Having this feature within the simulation tool can be considered essential because the team is not running the same platform. Additionally, V-REP has the option of two engines, Bullet or ODE, allowing for users to choose whichever most fits their requirements. The last feature that V-REP has is something that can benefit the development team if used properly, and that is the feature of allowing users to create and modify interfaces for the robot. In respect to the project, this would be very convenient in saving the team some time and resources.

Webots

Webots is another robot simulator tool that has been looked at. As previously stated, Webots is like the other simulation tools in respect to the fact that it provides the user to simulate a robot. The user can model and program robots and run the code virtually, allowing for a real time simulation to take place. Looking at Webots for its functionality and use, it’s one of the simulator tools that are being looked at. In addition, Webots was designed for professional use, along with education and research.

Furthermore, taking a closer look at Webots, it too has a wide scope of features that can be used by the team, in order to successfully test and build code. Webots features a library that includes code for their robots, sensors, actuators and objects. In addition to this, Webots allows for the user to upload their own CAD models. This means that the user can design their robot in a different piece of software, such as Blender, and then upload it to Webots for testing and use.

Webots is also capable of simulating a various number of different robots. These include robots ranging from robots with two wheels, to robots with multiple legs, or robots that can fly, or be submerged and more. In addition, Webots also has a few environments in which the virtual robot can interact with. This means that the virtual robots and code can be tested in a simulated environment as well, allowing users to edit and adjust components as seen fit, as if it were deployed into the real world. Having this small feature could benefit the team in respect to saving time and efficiency with testing in the real world.

There are a few components that make up Webots. These components include a GUI that’s powered by Qt, a physics engine that’s powered by ODE fork, and an open sourced rendering engine powered by OpenGL. Webots is also compatible with Windows and Linux, along with macOS. With Webots having the ability to be run on any platform, it benefits many users including this project, as the whole will not be using the same platform due to resource constrictions. Webots is also capable of using multiple coding languages such as C, C++, Python, Java, MATLAB and ROS (Robot Operating System). This is convenient because majority of the languages listed, are languages that the team is familiar with. Additionally, it is most likely that the team will be using python for the source code of the robot.

Another feature that Webots offers is the option to export the simulations as movies or interactive scenes on HTML. In addition to this, the simulation can also be streamed through the use of WebGL. In respect to the project, this feature doesn’t seem to be important for the final product, but if needed the option is available.

anyKode Marilou Robotics Studio  
 For simulating any robots and code, anyKode Marilou Robotics Studio is another piece of software that is used for simulating a robot and its functionality. It contains the capability of modeling robots and environments in respect to the laws of physics. Due to these key features, anyKode Marilou Robotics Studio easily becomes a candidate for simulating code and the robot.

Some other features that this piece of software has is its ability to help the user create models. This could be beneficial and increase efficiency for quick prototyping and testing. Also, anyKode Marilou Robotics Studio includes surface textures and properties, such as friction, reflection and many other factors that make up the real world.

Furthermore, anyKode Marilou Robotics Studio comes equipped with libraries that include data for motors, servos, and other mechanical components. These libraries also include data for other peripherals, such as sensors. In addition, there are virtual robots that anyKode Marilou Robotics Studio has already created, which allows the user to test their code against. In addition to this, anyKode Marilou Robotics Studio has documentation, allowing for new users to learn more about how the software works. Overall, these are extremely useful features because it allows the user to test their code and experimental builds with, while also allowing for quick prototyping.

anyKode Marilou Robotics Studio is compatible with a handful of languages which include, C, C++, VB#, J#, C#, and CLI. In respect to the project, this may not be the optimal choice for a simulation tool because as discussed prior, Python will most likely be the desired language for the source code. Furthermore, anyKode Marilou Robotics Studio works on Windows and Linux, but not macOS. Overall, because the list of supported languages does not include Python, it is most likely that this simulator tool will be excluded or overlooked as an option.

Gazebo

Gazebo is another robot simulation tool that can be used for the project, it is easy to setup, meaning that users can easily get into designing their robots and testing their code. Secondly, it comes with tutorials and examples which allow the user to learn basic concepts and put what they’ve learned to use. In addition to learning from examples and tutorials, Gazebo has a community where other users share what they’ve done and learned, through experimentation and sharing of concepts and ideas. Gazebo, with respect to the project, is already a compelling simulator that can be used due to its simplicity and user-friendly aspects, such as tutorials and community.

In regard to the different features, Gazebo has access to multiple physics engines allowing for a variety of different simulations and testing. Also, through the use of OGRE, Gazebo is capable of simulating realistic environments and high-quality textures and lighting. Another feature that Gazebo has to offer is the ability to add and create data from sensors, such as lasers, cameras and more. This is useful in respect to the project because not only will the robot be equipped with cameras and lights, but the option of adding additional sensors and peripherals are being taken into consideration for the final product. Additionally, Gazebo has another feature that allows users to use plugins for a variety of different things; such things include plugins for sensors and environmental control. Having the ability to use plugins as well as making custom plugins, may benefit the project in a sense that the team can create different plugins to test and experiment with. In addition to this, there may already be plugins that can benefit the team, in which the team won’t have to use time and resources making them.

Furthermore, Gazebo comes with a handful of robot models already provided. This means that users can start testing code against these models right away, before building their own model and robot. This is useful for the project because it allows the development of the robot to be as smooth as possible without wasting much time on deploying and redeploying the source code onto the real robot. Experiments and tests can be conducted on these included robot models allowing for users to be intuitive on how their robot may react with their robot. Overall, this feature is very useful and is a must when using a simulation tool.

Another useful feature is Gazebo’s ability to use CloudSim, which allows for users to interact and use Gazebo through a browser. This feature, in respect to the project, could be useful because multiple people will be able to develop one simulation from multiple locations and computers, as well as spectate and view the work that has been done. This in regard to efficiency and productivity is useful and would allow the project to progress in a steady manner.

Gazebo supports multiple physics engines such as ODE, Bullet, Simbody and DART. With multiple options for what physics engine is to be used, users are given more options and flexibility with how they would want to simulate their robot and environment. In regard to the project, this gives the team more options to work with, and allows for more testing and experimenting to find which physics engine is the most practical and useful.

Finally, because Gazebo can work with ROS, it’s most likely that this simulator tool will be the desired and used software for the project. This is because the team has decided to develop the robot through the use of ROS.

Conclusion

As mentioned, there are several types of simulation software that can be used, but due to the restrictions and confinements of the project, there are only a select few that can be used. These include V-REP, Webots and Gazebo, because from these options, each are using the language in which the team has agreed to use for the source code, Python. In addition to this each have their own strengths and weaknesses, with some overlapping. These overlaps occur in the fact that all of them have the feature of simulating the robot working with the source code, but some standout with the feature of adding and editing an environment for the virtual robot to work and move within. Overall, all of these simulator tools are effective in the aspect of simulating a robot and the source code, but it is most likely that Gazebo will be used for this project because of its easy to use interface and documentation, along with a community to learn from.