TEAM 4

OSCAR

Introduction

During the last 25 years, rapid uncontrolled urbanization has taken place in India. A substantial increase in areas has taken place due to insensitive developments of areas by private land developers and real estate business. The impacts of such unplanned development have its toll on environment, socio-economic and environmental requirements. One major impact is the sewage and drainage system and its management within areas of India.

Due to the absence of proper sewage facilities in real estate projects, water logging is a major problem. Waterlogging is created due to the accumulation of solid waste and excess silt deposition. Due to this silt deposition, it promotes the growth of aquatic plants which hampers the natural flow of stormwater and becomes a major environmental hazard as it serves as a breeding ground for mosquitoes and foul-smelling place. Therefore, in most of the area in India, solid waste has become a serious problem with health and hygiene consequences for the dwellers. Hence, our purpose is to build a robot system which would effectively clean the sewage canals and get rid of solid wastes like plastic and polythenes to prevent waterlogging and flooding and hence maintaining a free-flowing sewage line without any obstruction.

OSCAR (Open Sewer Cleaning Autonomous Robot)

OSCAR has one basic function and different modules designed for different tasks. In the demonstration bot, we have one basic function of autonomous sewer cleaning which includes waste collection and its dumping. OSCAR interacts in a multi-robot system to optimize all the sewer cleaning operations. This multi-robot system includes one or more robots and a central dock system where the robot dumps the solid waste collected. The central dock also serves as a charging station for the OSCAR. The robot performs the instructed operation autonomously with the help of GPS and IMU based localization and imagery input from the onboard camera.

Different Modules of OSCAR:

Floating Waste Collection

The head of the conveyor belt of the bot will be submerged in water. As the bot moves, the floating waste will be carried up to the bin by the belt. The belt is ‘cleated’ making it easier to carry the waste. The pan attached at the front too can assist this function.

Silt Removal

The claw attached at the tail of the bot will drop down into the water. It will reach the bottom of the ‘sewage canal’ and collect the deposited silt. It will carry the silt up, traverse to the top of the bin, and drop the silt there.

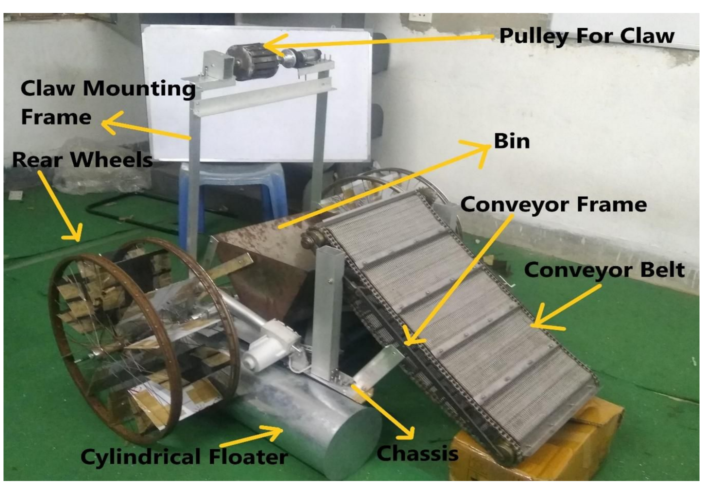
Mosquito Larvicide Sprinkler System

There’s a sprinkler attached at the rear end of the bot, which will sprinkle ‘larvicide’ (VTI) as the bot moves around in the canal. The ‘larvicide’ kills mosquitoes in their ‘larvae stage’ itself.

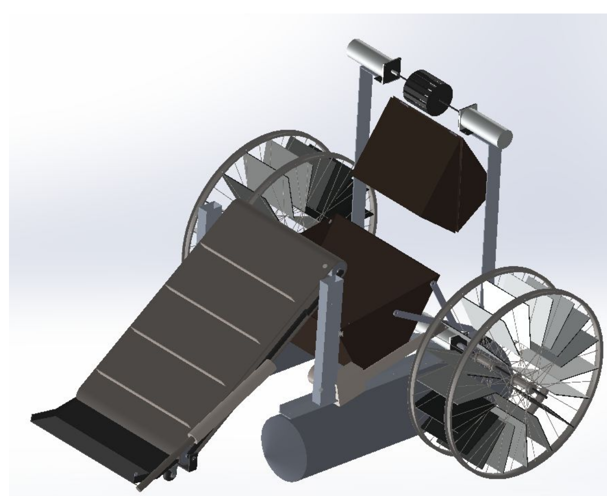
Aquatic Plants Removal

The bot is also equipped with two small rotating blades at the front end of the pan which can cut off the aquatic plants on the water surface. The floating detached parts of the plants can then be easily carried to the bin by the belt.

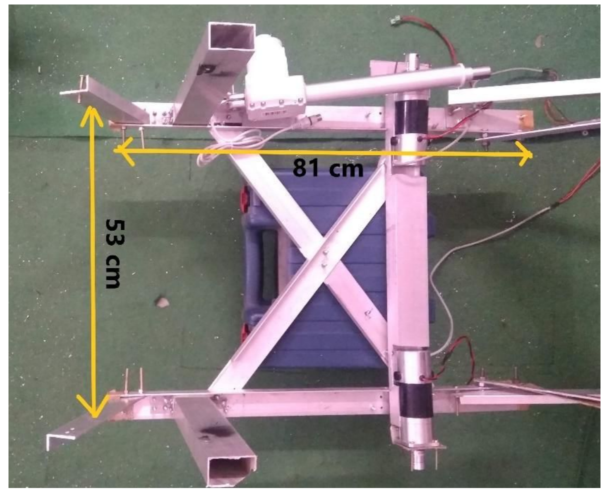
Mechanical Design:



CAD Model of Oscar:



Mechanical Design



The chassis is designed to support the Floating Waste Collection mechanism which is the basic module and additional modules such as Silt Removal and Mosquito Larvicide Sprinkler System. The chassis is equipped with a four-bar linkage mechanism to provide the lifting mechanism of the bin. The chassis is made so as to distribute the weight evenly over the surface of the cylindrical floaters. The chassis is made out of L aluminium bars in order to provide bending strength and reduction of overall weight simultaneously. The chassis along with mechanism of the bin also supports the conveyor belt at one end and the silt removal mechanism at the other end. The dimensions of the chassis are as shown in the image.

Material Consideration

This section contains the materials we have used to build our robotic system. The variety of components were chosen based on their reliability, availability and also economic feasibility.

Chassis

Material: Aluminium L bars

Properties:

- Well finished
- Lightweight and durable, thereby reducing an effective portion of the weight of the bot
- Anti-corrosion
- L shape bar imparts bending strength to the chassis

Bin, Silt Claw, Gears

Material: Mild Steel

Properties:

- High Strength due to the low amount of carbon
- High Weldability
- Can be used as a structural steel

- Cost-effective

Cylindrical Floaters

2 Floaters of Material: Stainless Steel (Coding)

Properties:

- High workability and ductility
- Improved Weldability
- High corrosion resisting properties

2 Floaters of Material: Polyvinyl Chloride(PVC)

Properties:

- Readily available and cheap
- High hardness and strength
- Lightweight plastic

Conveyor Belt

Material: AISI 304 Stainless Steel

Properties:

- Lower susceptibility to intergranular corrosion
- High ductility
- Resistance to the acidic environment
- High thermal resistance

Pan

Material: Cast Iron

Properties:

- High Compressive Strength
- High Resistance to deformation
- Excellent Machinability
- Resistant to destruction and weakening by oxidation

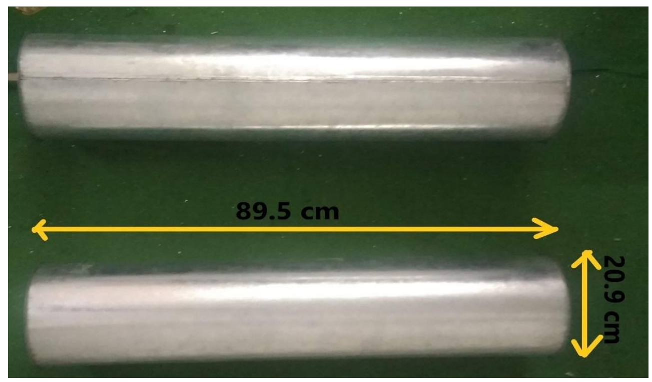
Drive Mechanism

Drive mechanism was designed keeping in mind the following objectives:

- Implementation of the differential drive to allow the robot to turn in the desired direction both on land and in water.
- Independent driving motors to avoid mechanical complexity in the differential drive.
- Specially designed wheels to ensure mobility on land and water using the same motors that simplify control.

Floaters

Four hollow cylinders made up of stainless steel are used in the bot for its floating characteristics. They displace a large volume of water resulting in a large buoyant force of the water which in turn balances the weight of the bot and enables it to float on the surface of the water. The cylindrical shape is preferred for these buoyant as this shape experiences minimum drag in the water and provides good support to the bot so that it cannot topple.

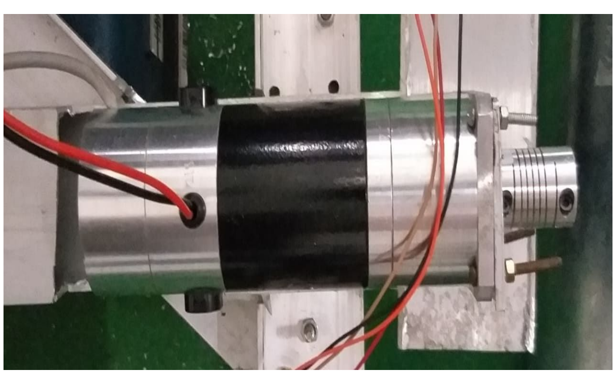


Specifications of large cylinder:
Length = 89.5 cm
Diameter = 20.9 cm
Thickness = 0.5 cm

Specifications of small cylinder:
Length = 26 cm
Diameter = 16 cm
Thickness = 0.4 cm

Calculations:
Volume of 2 large cylinder = $2\pi r^2 h$ = 0.0614 m^3
Volume of 2 small cylinder = $2\pi r^2 h$ = 0.0105 m^3
Total Volume = 0.0719 m^3
Average Weight of the robot = 50 kg
Safety Factor = 1.2
Buoyant Force needed = F_b = (50*1.2)*9.81 = 588.6N
Volume displacement required = $F_b/\rho g$ = 0.06 m^3
Percentage of cylinders dipped in water = (0.06/0.0719)*100 = 83.4%
These values are in the very safe range for the bot to float over the surface of water.

Driving Motors



The differential drive consists of the following:

Operating Voltage: 24V DC

Rated Torque: 142 kg.cm

Stall Torque: 344 kg.cm

No/Full Load Current: 1.12 amp/30 amp

Motor Selection

Taking into consideration the speed constraints, manoeuvrability constraints, loading constraints and other dynamical constraints, the power, torque and speed requirements were derived. The electrical power requirements depend on the choice of the actuators, hence the actuators were chosen to economize the expense on power components and actuators, but satisfied the power, torque and speed requirements.

Linear Actuator

Load Capacity : 6000N Load

Input Voltage : 24VDC Input

No Load Current: <1.5amp
amp

Speed: 4mm/sec

Linear Actuator

Capacity: 750N

Voltage : 24V DC

No Load Current: <1

Speed: 10 mm/sec

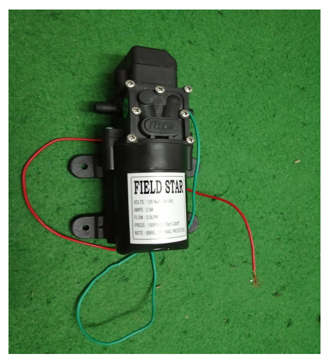
Chain Motor
Motor

Power Rating: 250 Watt Flow: 5.0

RPM: 300 rpm

Operating Voltage: 24V AMPs:

Driving Sprocket Pitch: ½ inch



Larvicide Sprinkler

LPM

Input Voltage: 12V

~2.8 amp

Pressure: 100 PSI

Motor Clamps

Motor clamps are used to clamp the motor to the chassis using bolts.

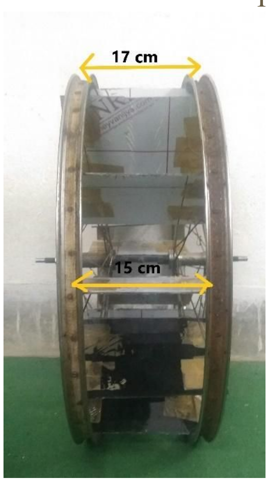
Shaft Coupler

Shaft coupler connects the wheel shaft and the motor output shaft which are of different diameters.

Rear Wheels

The wheels are designed with flaps with an intention that the flaps will push water behind and in turn, will help our bot to move forward and provide it with differential motion so that it can move to do the designated task. There are two motored wheels in our bot. Wheels are made by welding the axles of two-cycle wheels while aligning them parallelly together. The flaps are then joined with simultaneously aligned spokes of the wheels. While wheels are rotated these flaps will act as oars and push water behind and thus making our bot to move forward. We designed flaps such that only three flaps will be inside water at a time when the bot is moving. The force calculations were made by assuming that the water velocity is zero and the torque provided by the motor will help us provide sufficient change in momentum of water thus creating sufficient force which helps us make the bot move forward.

The flaps are made of hard plastic.



Dimensions of the flaps are 15*10 cm
Distance between the centre's of two-cycle wheels is 17 cm
Diameter of cycle wheels is 60 cm

Front Wheels

Swivel Caster wheels were used as the front wheels to balance robot on the ground. Caster wheels allow the robot to take a zero radius turn by rotating them in opposite directions because the caster wheels follow the direction in which rear drive wheels pull them.


Floating Waste Collection Mechanism

The floating waste collection mechanism consists of various integrated systems as listed below:

Pan

The 'bot' has a pan attached to its front which will be used to pick up the waste floating on the water surface or garbage on the ground. The pan will dump the picked-up waste and drop it onto the conveyor belt. The belt will carry it to the bin.
The pan is made up of cast iron painted by spray paint to avoid corrosion. It is driven by an actuator. A small binary link is attached at the tip of the pan enhances the pulling capacity of the actuator.
Maximum force the actuator can exert = 6000 N
Considering a safety factor of 5, Maximum Force = 6000/5 = 1200 N
Maximum length of the actuator = 50 cm
Minimum length of the actuator = 32 cm
At the lowermost position of the pan (when the actuator will have to exert its maximum force):

$l = l_{max} = 50$ cm

Approximate angle of the actuator from the horizontal in this position = 31.37°. The centre of gravity of the waste is calculated at 12 cm from the point of connection of the pan with the bot assuming equal distribution of garbage, we get W_{max} (where W: weight of the waste) to be 1637N. This means that ideally, the pan can lift a maximum garbage/waste weight of 167 kg.

Conveyor Mechanism

Conveyor Mechanism was designed keeping in mind the following objectives:

- To effectively collect the floating waste and debris from the water
- To minimize the power required to drive the system
- Easy to control and helpful for automation

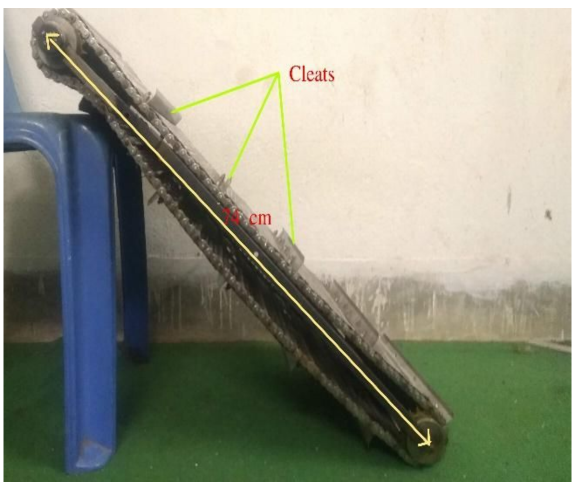
The conveyor mechanism mainly consists of two main subcomponents:

Conveyor Belt

Description*:* It is a compound balanced belt with ½'' pitch AISI 304 Chain & 5 mm diameter rod placed at every 3'' pitch with angle 25 x 25 mm placed at every 150mm
The dimension of the conveyor: EFF Width 390m x 1.750m LONG
Reasons for choosing the given materials:

- The pitch of the conveyor was decided based on the ease of the availability of gears of the same pitch in the market.
- Material of the belt was chosen because of its property to resist the corrosion
- Angles were strategically placed to keep the material collected on the conveyor from falling

All the design considerations were made and optimized to reduce the power requirement of the conveyor drive while not compromising either with safety or features of the material.

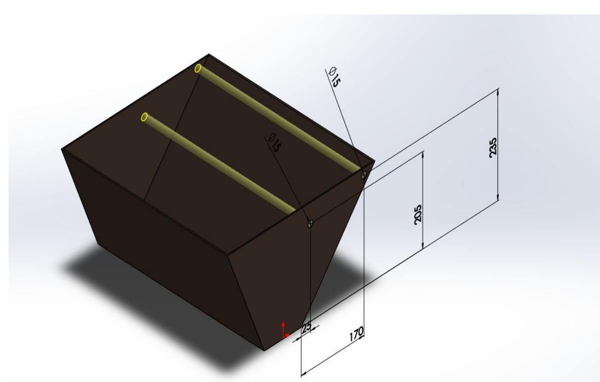


Conveyor Mount

The mount was specifically designed considering all the major forces and moments acting on it. The analysis of the mount was done in Ansys and then the dimensions were fixed.
The dimension of the conveyor mount is 74cm x 47cm.

Bin

The bin is designed to collect waste transported by the conveyor belt and is used as a temporary storage container. The bin is made to rest on the chassis just below the edge of the conveyor belt. The back face of the bin is maintained at a higher obtuse angle than the front face, so as to easily dump the waste in the waste collector.

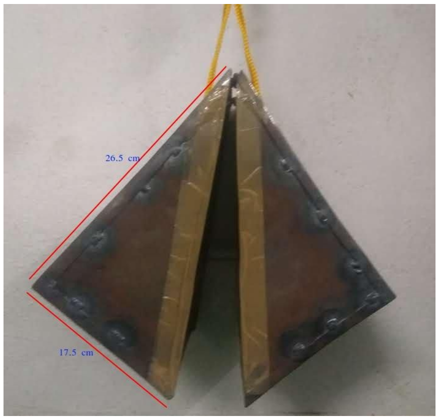


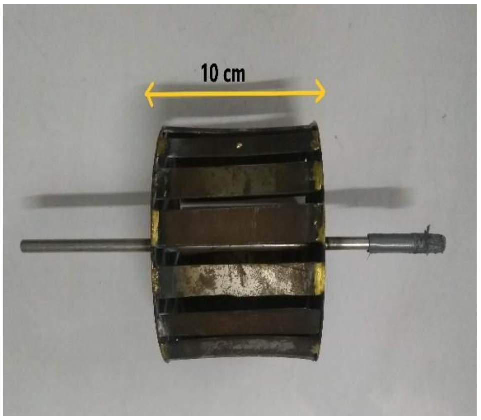
The bin is made out of mild steel sheets of thickness 0.75mm, welded together. The dimensions of the bin are shown in the image.

Silt Removal Mechanism

Silt deposited in open sewage is a major problem which labourers have to remove manually using the spade. The design and mechanism are so chosen such that it can pick up silt from the bed and it can dump it in the bin. The whole mechanism consists of 1) Pulley 2) Aluminium linkages 3) Motor 4) 2-Actuators 5) Nylon rope 6) Claw collector

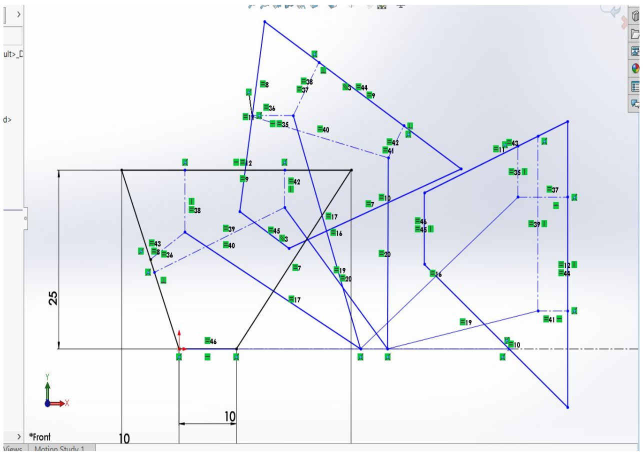
The actuator attached to the claw helps in collecting the silt while the other actuator attached to the link helps position the claw while dumping as well as while collecting the silt. The pulley is fixed with the motor so that it can length of the rope can be varied so that hand can be placed at varying bed to bot length. The actuator used is for the hand is IP44 waterproof. The hand garbage capacity of hand is 28.11 litres.





Waste Bin lifting Mechanism

A four-bar mechanism is designed to lift the bin from the chassis and dump the waste in the waste dock. The linkage lengths and origins are decided based on the three-position synthesis technique. Position 1 is as shown in the figure is that of the bin rested on the chassis of the robot. Position 2 is to maintain a height of about 25 cm from the base frame so as to occupy electronics and spraying mechanisms of the bot. Position 3 is chosen to dump the waste



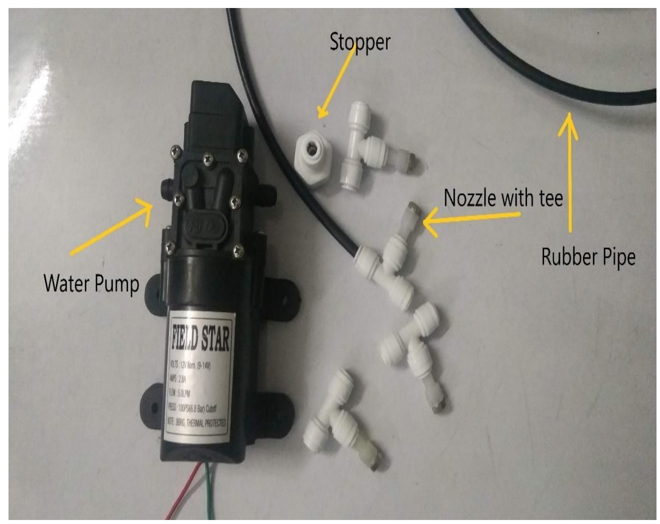
in the waste dock when the bot would approach it from the rear end.

Larvicide Sprinkler System

Mosquitos spend around 4 -14 days of their life span in the larvae stage. To eliminate the mosquitoes at this stage from the sewage, the robot is equipped with a larvicide (BTI) misting nozzle sprinkler system. The sprinkler system is attached to the bot at its rear end and consists of a storage tank of capacity 3 litres, water pump of 100 psi, pressure, misting tees, nozzles and rubber pipes. The nozzles are arranged so as to create a thin film of the chemical BTI over the water surface.

Why is BTI chosen?

Bti is used to kill developing mosquito larvae by being applied to standing water where those larvae are found. Bti has been shown to be effective in reducing mosquito larval populations and could be effective in controlling mosquitos carrying Zika, dengue, and chikungunya. No special precautions are needed for applying Bti. Bti has no toxicity to people and is approved for use for pest control in organic farming operations. Bti is also approved for aerial spraying. Bti can be sprayed over water bodies such as ponds, lakes, rivers, and streams.



Solar Panels

The solar panels used is rated at 3W and 9VDC per module. The material used in the solar cell is monocrystalline silicate which provides higher efficiency than amorphous silicate. Several modules are used to charge the battery which drives the robot.

Electronic Architecture

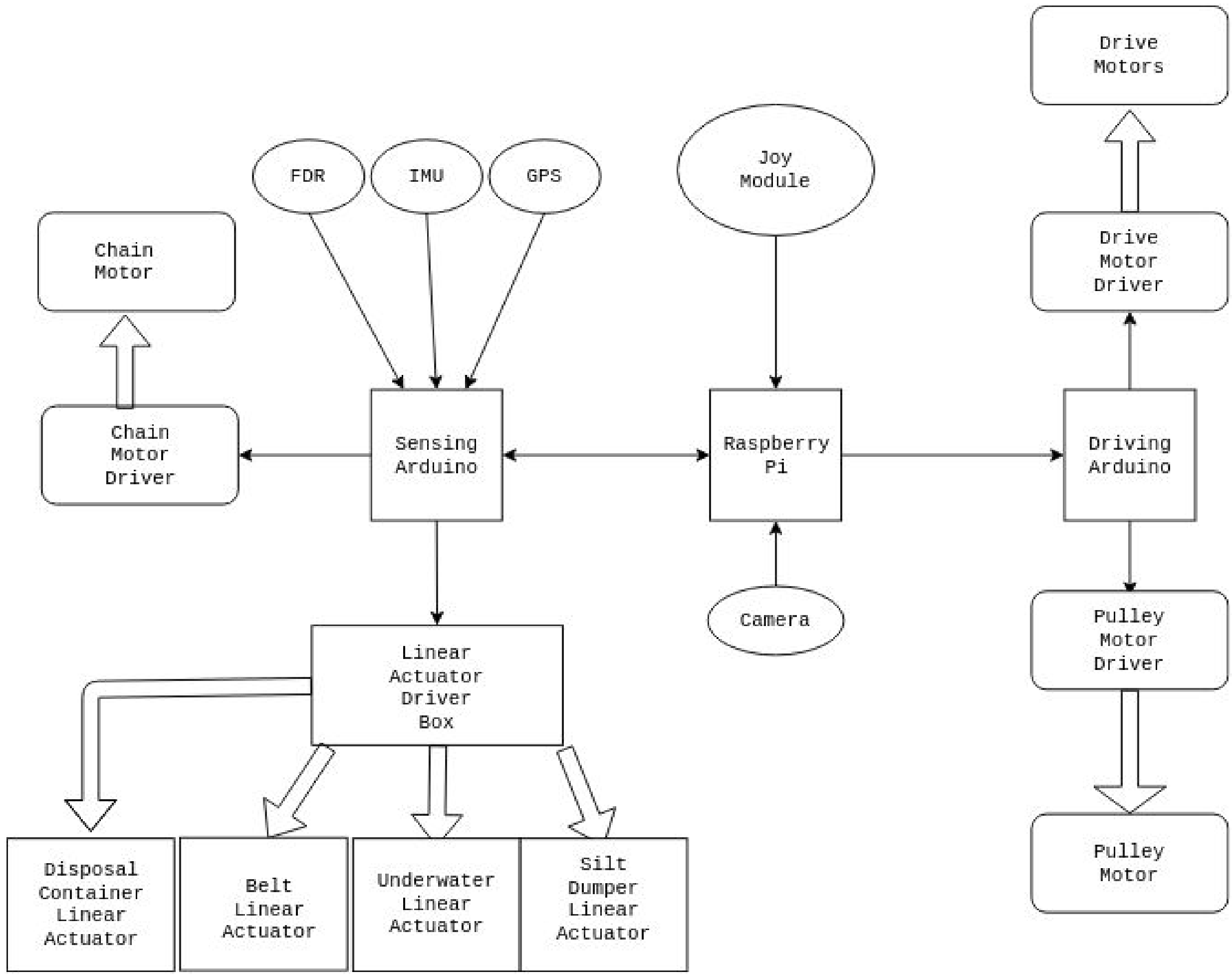
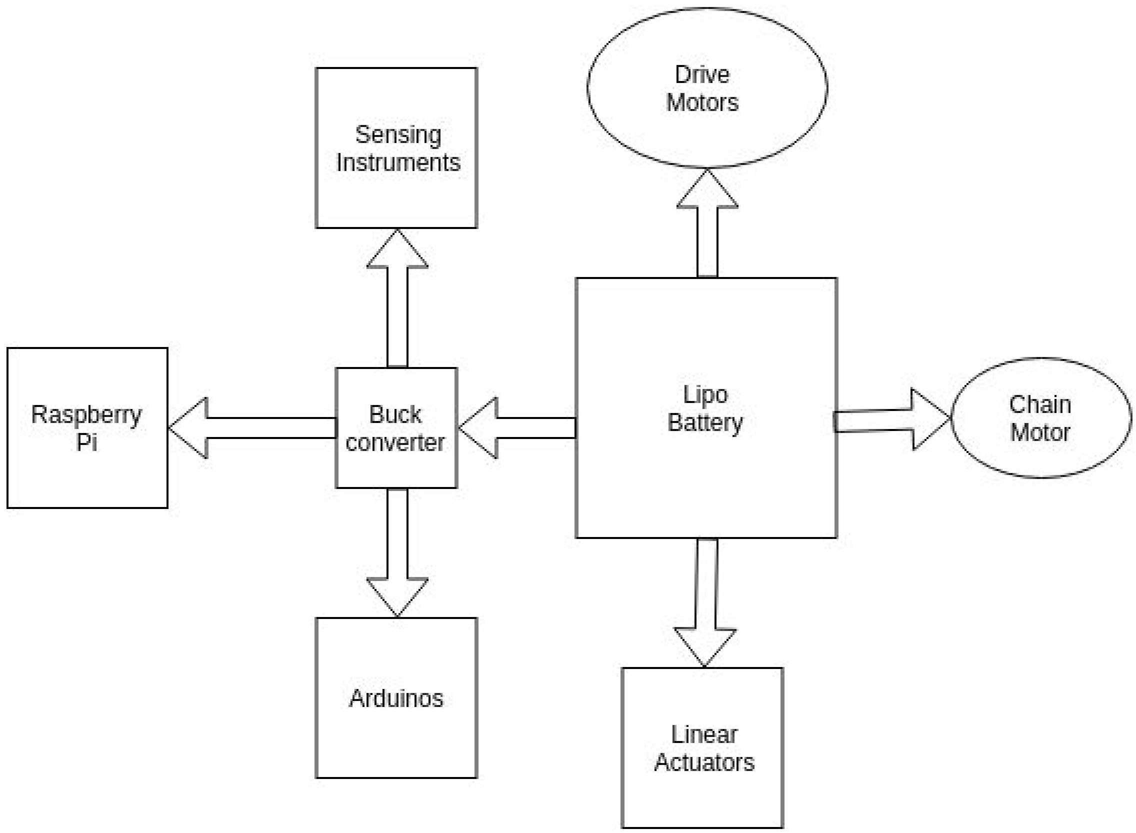


Fig: Embedded system flowchart

Power Subsystem

The robot is primarily powered by LiPo batteries. Different systems require different voltages and currents. So, voltage converter ICs and buck converters have been used in the circuits. The chain motor needs high power input, and hence it is powered by a separate LiPo battery.

All the other sensors, actuators and motors are powered by the other LiPo battery. As the motors and linear actuators can still extract high currents from the batteries, safety fuses have been used.



Sensing

The automation of the bot is implemented through positioning provided by the satellites i.e. GPS. To enhance the usual 10-meter accuracy provided by the GPS system IMU (Inertial Measurement Unit) is appropriated. The autonomous mowing and sowing operation is achieved by constantly tracking the robot's position with respect to the frame of reference fixed to the work field after it has been mapped. The sensors work in tandem to track the bot position as follows:

USB/UART GPS Module [RKI-1610]

After trilateration, the module provided with patch antenna on top outputs NMEA $GPGGA data stream which is parsed into Navigational Coordinates - Latitude and Longitude employed in robot's localization.

Specifications:

- Low Power Consumption : 55mA @ acquisition, 40mA @ tracking

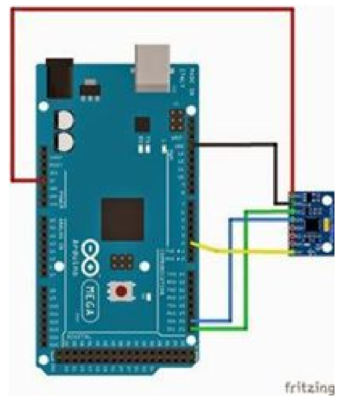
- Position Accuracy Without aid: 3m 2D-RMS
- DGPS (RTM, SBAS (WAAS, EGNOS, MASA)):2.5m 2D-RMS
- Data output Baud rate：9600 bps
- Max. Update Rate：5Hz (Default: 1 Hz)

MPU6050: 6 Axis MEMS Accelerometer + MEMS Gyro meter

The inertial data i.e. linear accelerations and the angular velocities along the axis obtained from the sensor is fed into an orientation filter to extract the data in Quaternion orientation form employed in active localization.

Specification:

- Accelerometer ranges: ±2, ±4, ±8, ±16g
- Gyroscope ranges: ± 250, 500, 1000, 2000 °/s
- Voltage range: 3.3V - 5V (the module include a low drop-out voltage regulator)



Interfacing

It comprises the methods by which all the sensors and actuators communicate with the central controller. Since the embedded architecture on OSCAR is based on the distributed processing of the varied data, it is required to have a proper communication scheme at each level. The central controller we have used here is Raspberry Pi2. It communicates with the three Arduino MEGA boards and also receives the control commands for the actuators like motors and servos. On each Arduino, an interfacing code is developed individually and uniquely.

We have two major types of interfaces in the OSCAR.

Sensors-Arduino Interface: - I2C and Serial UART

The raw data received from the sensors is processed and packed at the Arduino board. The process of receiving the data from the sensors is done with the required communication protocol. Inertial Measurement Unit (IMU) sensor and the GPS module are interfaced with the Arduino via the I2C and Serial UART Communication protocol respectively.

I2C is the two wire interface where one wire is for the Serial Data signal and other for the Serial Clock signal. Arduino Board as the master device shares its clock signal and a single data line is used for the transfer of data between the master and the slave as per the request for data generated from the master.

Serial UART Communication with GPS is achieved through Receiver and Transmitter signal lines. The GPS module sends the NMEA data streams which are then parsed in Arduino into Latitude and Longitude.

Arduino-Raspberry Pi 2 Interface: - Serial

The processed data at the Arduino boards are sent to the Raspberry Pi 2 for the localization with the use of serial communication. We have used the Robot Operating System (ROS) to integrate all the intercommunicating nodes at one place. With the use of ROS serial library, it has become a lot handier to communicate between Arduino and RPi.

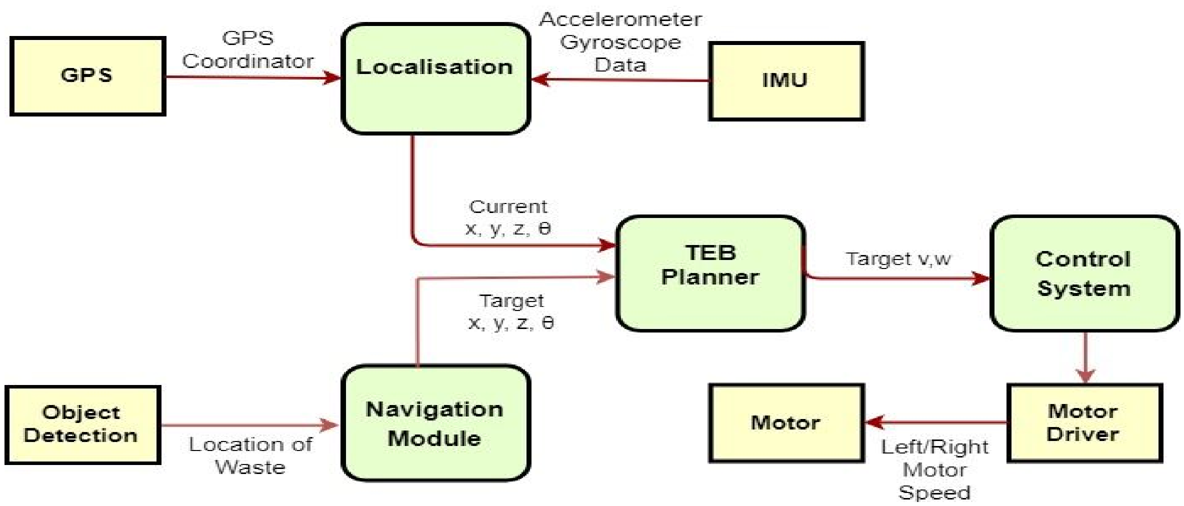
Arduino Boards sends the following data to the Raspberry Pi:

- Quaternions
- Angular Velocities
- Linear Acceleration
- Parsed GPS data
- Processed data of Encoders

Simultaneously Raspberry Pi also sends the control commands to the Arduino which eventually actuates the motors and servos.

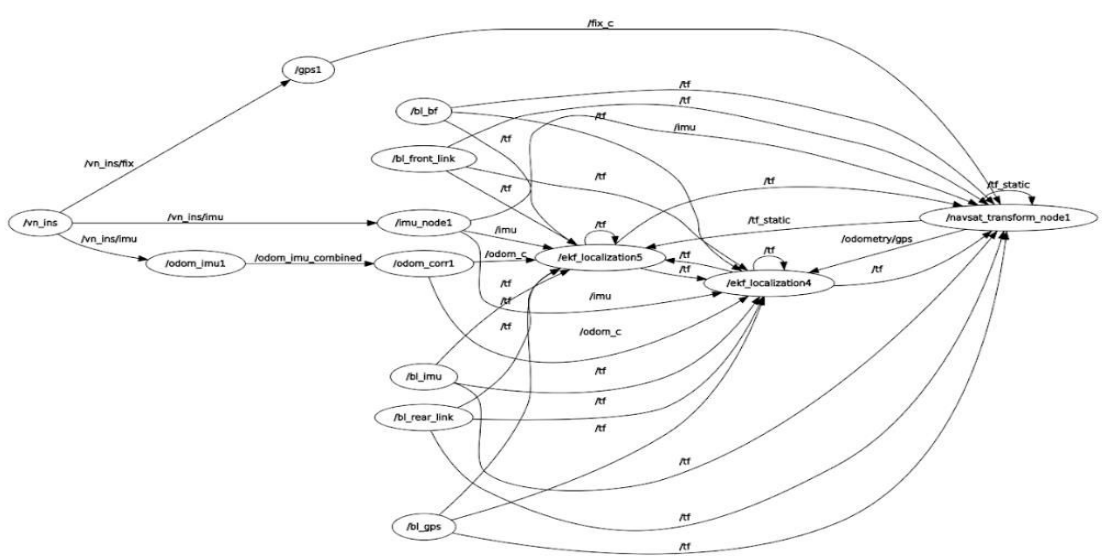
Software Architecture

The software architecture of the robot consists of modules like Localisation, Path Planning, Object Detection.



Localisation

The backbone of spatial awareness of a robot is localization. Proprioceptive sensors used are IMUs and Motor Encoders. The environmental sensor used is GPS. Data from the proprioceptive and environmental sensors are fed into a Localization package. Localization package consists of extended Kalman filter (EKF) which is used to filter the sensory data and then get pose estimation from the filtered sensory data. Mapping is used to constantly update the cut and uncut area. Mapping is done by using a flag which is toggled when the robot visits the corresponding grid. We are using ROS to integrate the system of Kalman filter and pose estimation. The following image explains ROS architecture.



Object Detection

Object detection is a basic skill for a robot to perform tasks in human environments. In order to build a good object classifier, a large training set of labelled images is required; this is typically collected and labelled (often painstakingly) by a human. This method is not scalable and therefore limits the robot's detection performance. We propose an algorithm for a robot to collect more data in the environment during its training phase so that in the future it could detect objects more reliably. The first step is to plan a path for collecting additional training images, which is hard because a previously visited location affects the decision for future locations. One key component of our work is path planning by building a sparse graph that captures these dependencies. The other key component is our learning algorithm that weighs the errors made in the robot's data collection process while updating the classifier. Our work is motivated by active learning, where the algorithm selects the next data-point to be labelled and uses that for training. Our proposed algorithm also selects a path where the robot takes the new views, and then uses these views for training the classifiers.

For a robot to perform object detection in sewer environments, existing online datasets are not good enough because of two reasons. First, they may not have enough data points for that object type or that specific object. Second, online images may be available for some canonical views, but in practice, a robot may see the object from any arbitrary view. Objects can appear drastically different from different viewing angles. Therefore, we believe that by actively collecting more data in its environment and learning how to use it, a robot can significantly improve the detection performance. Two questions arise while actively building the object classifier: where to take the images and how to update the object model. For the first question, we note that not all additional views are equally useful. New training images are dependent on each other, and observing from one view may make future observations from other views redundant. To address this, we build a sparse graph that represents these dependencies and presents an efficient algorithm to plan a path. Also, we consider additional challenges such as the robot may be able to observe multiple objects in the same view, some obstacles may prohibit the robot from moving into certain areas, and other obstacles may block the view. The second question of "how" to update the classifier is hard because, in order to use the collected data, a robot first needs to separate the object from the background. To label the object in the new training images, we use a Hidden Markov Model (HMM) to combine the evidence from tracking subsequent views with the beliefs from the existing classifiers. These labels may still be noisy, and therefore they are used with a grain of salt in the re-training phase these labels are weighted to represent how good the algorithm thinks the new data-points are.

Our approach consists of the following three steps:

- Model the importance of different views by a "utility" function. We want to take training images from locations that efficiently cover different views of the object.
- Plan a path for the robot to take new training images. New images are dependent on each other for re-training the classifier. We model this dependency through a sparse graph. We then plan a path for the robot that maximizes the total utility gain while keeping the total length of path small and also avoiding obstacles.
- Use the collected images to update the classifier. We first generate the labels using an HMM that combines tracking and detection beliefs. We then use this labelled data for training our soft-margin classifier, which takes this noise in the labelling into account.

Waypoints detection

For any mobile device, the ability to navigate in its environment is important. Avoiding dangerous situations such as collisions and unsafe conditions comes first, but if the robot has a purpose that relates to specific places in the robot environment, it must find those places. For OSCAR, these specific places refer to the regions of waste.

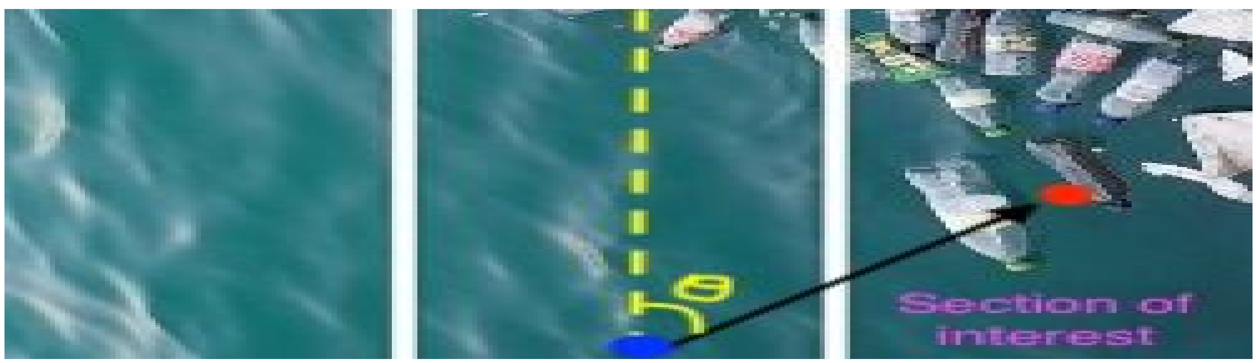
Keeping in mind the onboard computational capabilities and limitations, we are averse to use modern machine learning and deep learning models for the following reasons -

- An inherent pitfall of these methods is that their runtime performance is capped by the quality of data used at train time. And since the visual nature of water is dependent on a variety of factors such as the environment, time of day, temperature, depth, texture (presence or absence of silt), local flora and fauna, and a multitude of other factors, it is virtually impossible to collect sufficient data that would promise good run-time performance.
- Since OSCAR is solar powered, we are restrained to use low energy consumption resources, which have lesser computational power but is cost-effective. The convolution operation used in convolutional neural networks (CNNs) is very expensive but the domain demands fast real-time performance and so usage of CNNs is not feasible on the hardware used.

All things considered, we propose the following vision based pipeline for navigation which successfully overcomes above-stated drawbacks and moreover, adapts to the current environment dynamically.

Every raw camera image (front view) is converted to the bird's eye view (commonly known as the top view) using a perspective transform by using a homography matrix. For a specific set of camera parameters, the homography matrix is only calculated once. We maintain a weighted rolling average of the colour distribution of the water.

W



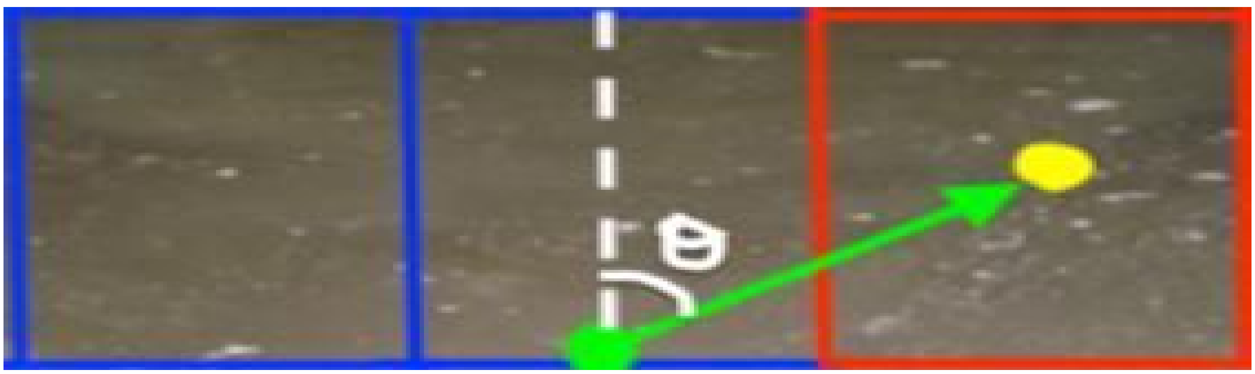
- Section the image into 3 vertical segments to isolate the garbage section-wise and to give one of 3 direction headings - left, right or centre. The section which has a colour distribution most different (difference must be above a chosen threshold) from the tracked rolling average of water is chosen as the segment that contains the waste and is the segment to move towards.
- If neither of the 3 segments has a colour distribution that is significantly different from the water distribution, then the bot is commanded to move straight.
- The rolling average is updated using those segments that have a colour distribution that is similar (difference must be below a chosen threshold) to the colour distribution of water.

For navigation, a waypoint is to be passed to the planner so that a local trajectory may be generated and appropriate commands can be sent to the motors by the control systems in place.

The centre of the bottommost row is the location of the camera with respect to the image. (shown as the blue dot in the above figure)

We give the centre of the section of interest as the coordinates of the waypoint, taking the bot as the origin. The x and y coordinates in the image coordinate system are multiplied with the calculated constant of metres per pixel so that they are transformed into the real world coordinate system, with actual distance from the bots. (shown with the black line and the red dot in the above figure)

The angle for the waypoint is the angle of the straight line joining the bot to the centre of the chosen section. (shown as θ in the above figure)



REFERENCES

- https://www.arduino.cc/
- http://wiki.ros.org/dwa_local_planner
- http://docs.ros.org/lunar/api/robot_localization/html/index.html
- http://ieeexplore.ieee.org/document/7331052/
- Shigley's Mechanical Engineering Design - Book by J. Keith Nisbett and Richard G. Budynas
- https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/
- Theory of Mechanisms and Machines – Book by Amitabha Ghosh and A K Mallik
- Image processing library (OpenCV) documentation