B.M.S COLLEGE OF ENGINEERING

Bull Temple Road, Basavanagudi, Bangalore - 560 019, Karnataka, India

MAJOR PROJECT

SYNOPSIS

ON

AUTONOMOUS ROBOT DEVELOPMENT OPEN SOURCE PLATFORM - THIRD GENERATION (ARDOP 3.0)

Submitted in partial fulfilment of the requirement for completion of

MAJOR PROJECT

[16EC8DCMPJ]

Submitted by

NAME	USN
SUDARSHAN S HARITHAS	1BM16EC109
V AMRUTH	1BM16EC115

2019 – 2020

Under the guidance of

Harish V. Mekali

Assistant Professor, ECE Dept, BMSCE

Department of Electronics and Communication Engineering

CONTENTS

1. Introduction

The Project ARDOP 3.0 deals with Design and Development of a BMS Humanoid Robot. This is the 3rd version of the humanoid that is under progress. Robots are becoming prevalent in the world today as they are being used in a multitude of different sectors such as surgery, autonomous driving, etc. Intelligent robots that are capable of performing the required tasks are built using a modular approach and by integrating the various subsystems. ARDOP is an open source humanoid robot that would serve as an extensive platform for the development of robotic applications. The aim is to make a full-fledged integrated hardware and software platform so that it can be used for research purposes to accelerate the growth of robotics. The intention is to build an intelligent autonomous robot that would be used to forward research in the fields of computer vision, control, kinematics, motion planning and machine learning. ARDOP would serve as a platform for students and researchers to implement their ideas and algorithms. One of the primary aspects of ARDOP is its ability to perceive and interact with objects in its surroundings. Such a task is achieved by the seamless integration of various domains of robotics, electronics and computer science such as kinematics, computer vision, machine learning, CNN (convolutional neural networks), mapping, localization and navigation. ARDOP consists of two 6DOF robotic arms, a Kinect camera for vision and a four-wheel Mobile base for locomotion.

Fig1: Ardop3.0

2. AREA OF APPLICATION / WORK/ DOMAIN

The main intention of this project is to build an intelligent autonomous robot that would be used to make acquaintance, develop physical models for implementation and forward research in the fields of computer vision, kinematics, motion planning and machine learning. The integration of these concepts is essential for any modern intelligent robot that wishes to make a mark in the rapidly advancing field of robotics. ARDOP 3.0 would serve as a platform for students and researchers to implement their ideas and algorithms in this direction. They will be able to use pre-made algorithms, modify them and execute them with ease, or develop new ones and test out their executions. The different abstraction layers in the project will enable people with all levels of technical expertise to build and develop algorithms and functionality in the robot. Being economical, the platform would help make robotics research more feasible.

- Robotics Research (Humanoid Robots)
- Computer Vision and Perception
- Kinematics and Manipulation
- Machine Learning and Deep Learning

3. Literature Survey

3.1 Vision and Perception

The object recognition problem is solved in a plethora of ways today. Computer vision-based methods are moving closer into the field of machine learning in order to get more accurate results. Classifiers such as neural networks and support vector machines are widely used to classify images and they tend to work well on images containing objects. Another popular method to find shapes in an image is the Hough transform. This method has been extended in many ways in order to improve performance.

The main drawback in using a traditional machine learning approach is that the features have to be manually selected. Very often, researchers spend a major portion of their time finding appropriate features that describe the dataset. The performance of the system depends heavily on the features used as the input and the model would have to be implemented and trained to evaluate the contribution of the features in producing accurate results. This is very cumbersome and time-consuming process and it significantly reduces productivity. Hence, computer vision algorithms are employed in order to extract features in the images which then can be fed into a classifier. A typical image classification process is shown in

Computer vision libraries such as OpenCV have inbuilt code written in order to implement these methods 5 quickly and they provide a real-time recognition application.

Although these algorithms work very fast and can provide high frame rates, they don't perform well on certain data sets. Although first invented in the 1960s, Convolutional neural networks have made their way back into the modern scene as computational power has increased enormously in recent times. CNNs eliminate the problem of manual feature selection by learning appropriate filters which extract the most influential features for the given data.

3.2 Kinematics

Kinematics is a branch of mechanics that deals with the motion of objects without taking into account the forces that act upon them. Hence it is specifically used for pose estimation of objects- to find out their positions and orientations in three-dimensional space. Kinematics is used to study the motion of complex link-joint structures where each joint actuates a particular link system. By employing kinematic equations, it becomes possible to find out the joint parameters that would make the structure move in a way that is optimal for the given task. The end-effector of a robotic arm is the part that comprises of a gripper or some other manipulating structure which can be used to pick up objects, etc. The goal of the kinematic model is to enable the end-effector to reach the desired position to allow for interaction with real world objects. The solution obtained from the kinematic equations are the angle values for each of the joints that would make the end-effector

move to the required position with the required orientation.
High DOF Robotic arms are typically developed based on inverse kinematics models so that the arm can reach any position or orientation in structured environments. The forward kinematic model is predicated on Denavit Hartenberg (DH) parametric scheme of robot arm position placement. Given the desired position and orientation of the robot end-effector, the realized inverse kinematics model provides the required corresponding joint angles.
Solving an Inverse Kinematics problem analytically means to compute the joint angles by manipulation of the Forward Kinematics equation. An iterative method computes the joint angles by changing the joint angles by small amounts every iteration. At one point, this process would converge and the joint angles would remain constant. The values at this point are taken to be the solution of the Inverse Kinematic problem.

3.3ARDOP 1.0

In addition, since we are taking forward the ARDOP 1.0, project and research into it is imperative. Convolutional Neural Networks (using CAFFE) is used to implement object recognition using the stereo depth map. The stereo camera used is DUO MLX, however this has a limitation that it provides only grayscale images. In order to use it with the CNNs for object recognition, RGB images are required which will improve the classification and localization accuracy. One solution would be to use additional RGB camera and requires us to solve the registration problem in order to synchronize the RGB and Depth maps. Another drawback is that the DUO MLX APIs are not open source and the camera itself is very expensive. Trajectory planning involved the computation of joint or end-effector trajectories to obtain required arm trajectories. Servo motors are controlled using I2C controller. Kalman Filter is used to eliminate distance measurement errors. ARDOP1.0 is shown below.

3.4 ARDOP2.0

ARDOP 2.0 initiated the use of ROS and built a Moveit Package to solve the inverse kinematics in simulation. They also integrated the gazebo simulation with Rviz solution. But their design of the arm was bulky and had unintended offsets at the wrist joints this decreased the accuracy of the kinematics and increased the required torque of the motors hence increase in the amount of power sourced. They also replaced the DUO MLX with Kinect this helped in easy integration with ROS and also active use of drivers such as Openni. The motors were driven using buck and boost converters and an SMPS was used as the main power supply, this also limited the mobility of the robot. They initiated the use of Darknet and YOLO V3 for object recognition which we plan to continue for ARDOP3.0.

4. PROBLEM DEFINITION

On the tree of robotic life, humanlike robots play a particularly valuable role. It makes sense. Humans are brilliant, beautiful, compassionate, loveable, and capable of love, so why shouldn't we aspire to make robots humanlike in these ways? Don't we want robots to have such marvellous capabilities?

Certainly, robots don't have these capacities yet, but only by striving towards such goals do we stand a chance of achieving them. In designing human-inspired robotics, we hold our machines to the highest standards we know—humanlike robots being the apex of bio-inspired engineering.

The problem statement includes:

4.1 Mobile manipulation

The robot has to move from the starting point to the source zone where it has to pick up an object and move towards the destination zone and place it. The working methodology can be described as shown in the block diagram, the 3D camera (Kinect) identifies the object and provides that is to be picked and placed, the object is picked by the arm by performing inverse kinematics, and the process of localization and navigation is performed using SLAM (simultaneous localization and Mapping).

4.2 Human Robot Interaction (HRI)

To develop capabilities for ARDOP to Interact and Engage with Humans in a Game of Tic-Tac-Toe

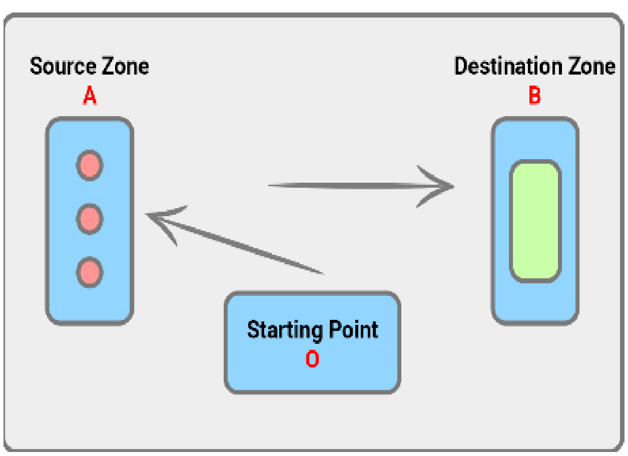


Fig2: Mobile manipulation

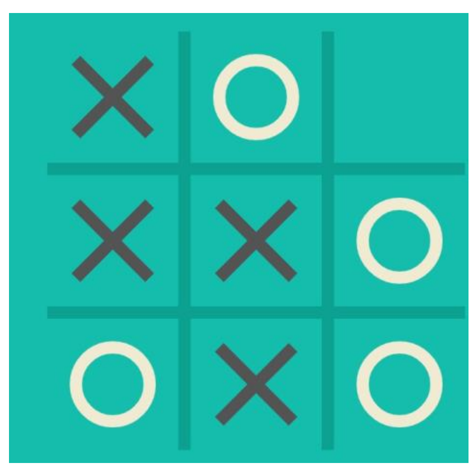


Fig3: Tic-Tae-toe game (HRI)

5. PROPOSED SOLUTION

This project covers Design decisions, Development process, and Test case selection in the making of the novel, custom-designed humanoid platform named ARDOP (Autonomous Robot Development Open-source Platform). ARDOP is an integration of 3 major functional units Computer vision, Manipulation, Navigation. Computer vision is built around Microsoft Kinect and You Only Look Once (YOLO) V3. Manipulation consists of two 6 Degree of Freedom (DOF) arms built using servo motors and iterative Jacobian pseudo inverse method. Navigation system is built using hub motors, Visual inertial odometer and Simultaneous Localization and Mapping (SLAM). 3D camera sensor data is processed using laptops, other sensors data and user data is processed on the Atmega2560 controller board. ARDOP is powered with two independent power supply units, one for computer vision & manipulation and other for a navigation system. The strong, stable but simple mechanical structure is designed and fabricated using aluminium and then insulated with powder coating. Before the actual development, the complete robot has been modelled in Solid works and simulated in ROS (Robotic Operating System) using MoveIt and Gazebo. Finally, custom-designed test cases for testing the Accuracy, using Pick & Place and Point to Point navigation and testing Logical ability, using Tic-Tac-Toe game have been developed and adopted to validate the ARDOP's functionality.

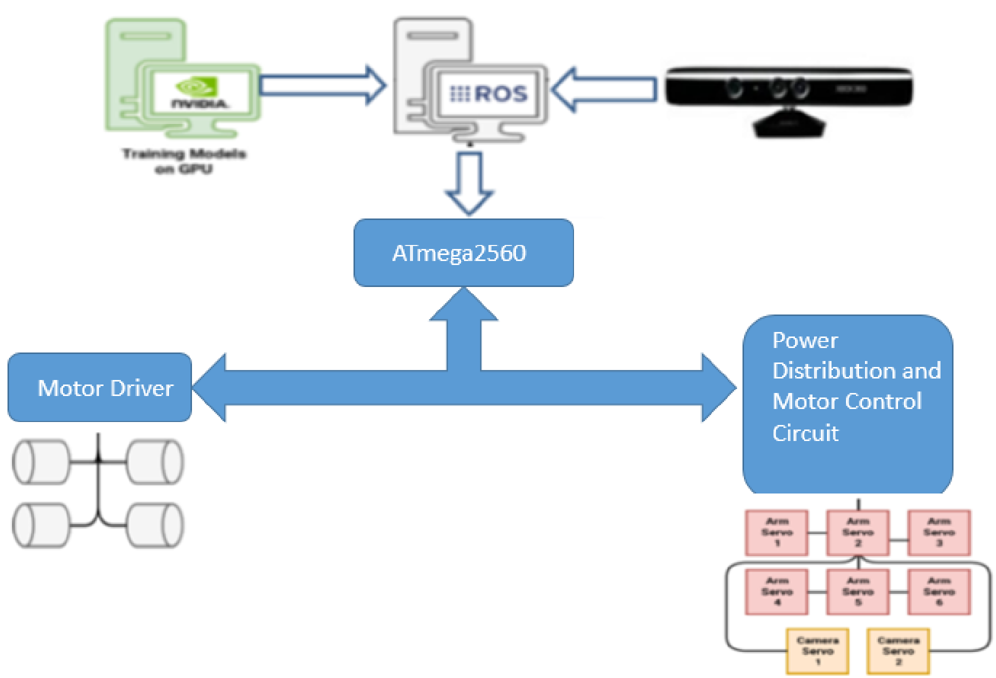


Fig4: Block Diagram overview

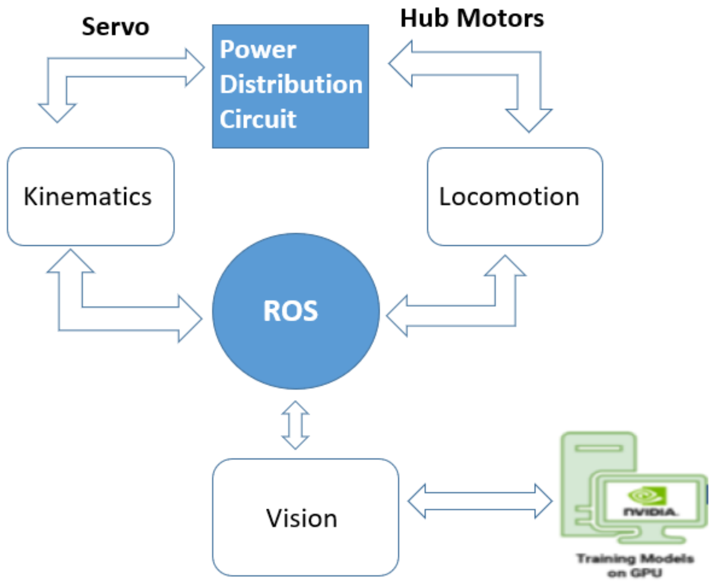


Fig5: System integration

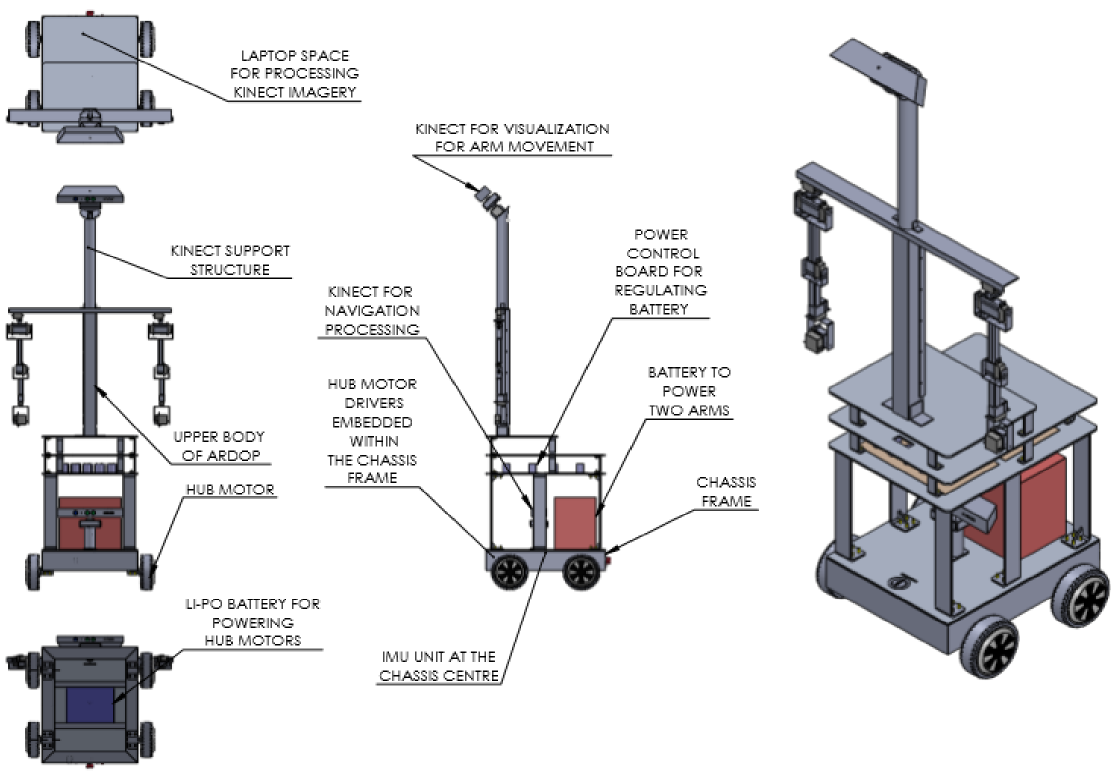


Fig6: Design of ARDOP3.0 with Mobile Base

6. IMPLEMENTATION STAGE – 1

6.1 Kinematic system:

A robotic arm with 6 degree of freedom to pick up the object of interest identified with the vision system without effecting the other objects in the work area. The following diagram describes the working of the kinematic system in simulation and in hardware. A solidworks file is converted to URDF and proceeded for simulation using ROS, Moveit. It generates files such as SRDF, Config and launch files which are further used by KDL to obtain the kinematic solution.

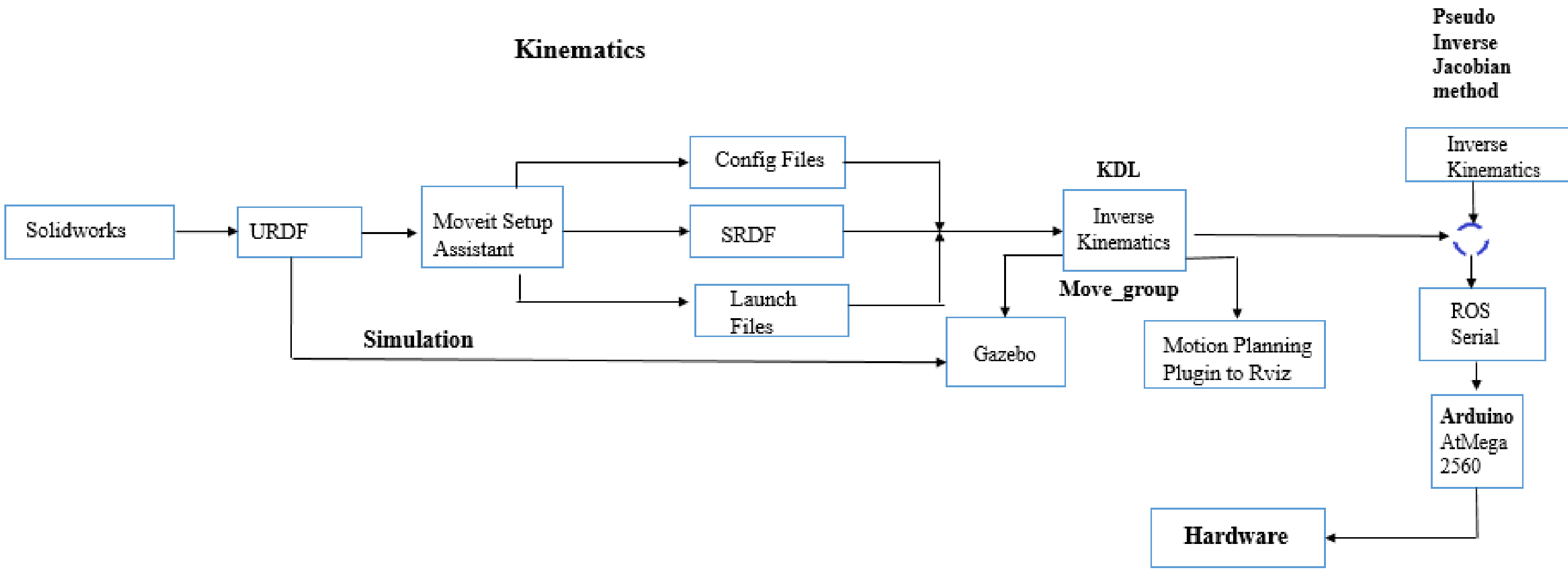


Fig7: Kinematics flowchart

6.2 Vision System:

3D camera sensor that provides surrounding objects features with depth information i.e., RGBD information to a processing unit that precisely identifies the object of interest from the field of view and localizes it in the 3D space. These values are further transformed into the camera base frame from where the kinematics system would take over to generate the solution.

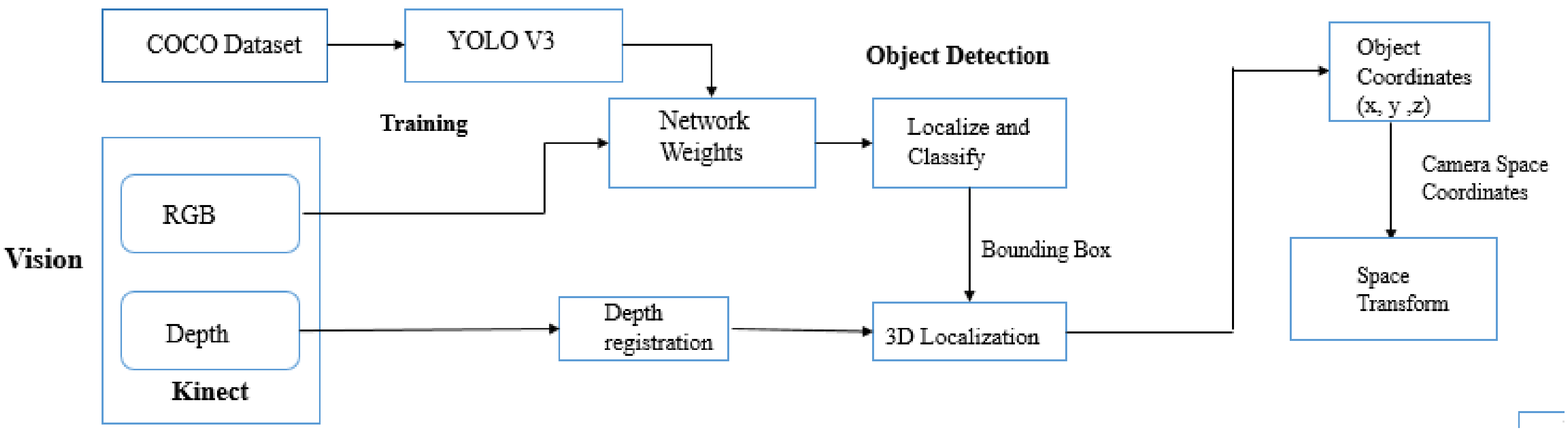
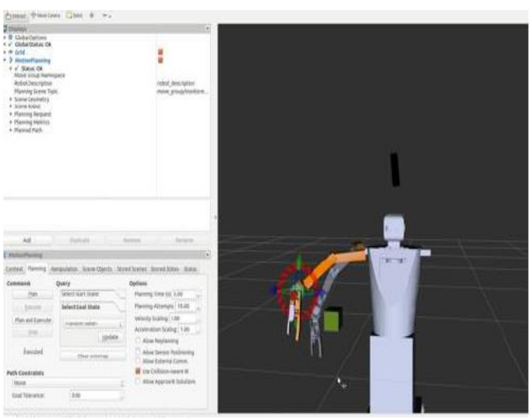


Fig8: Vision system flow chart

6.3Implementation Results:

1. Manipulation System: Simulation Results

Simulations have also been conducted and the results are presented below. It can also be noticed that an obstacle aware IK has been implemented during simulation.



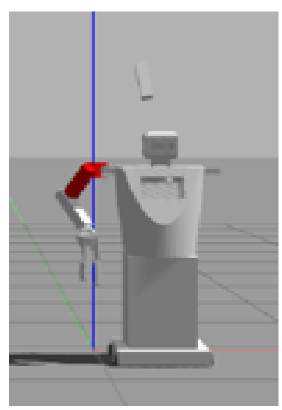


Fig9: Rviz and Gazebo simulation results

2. Vision System:

YOLO V3 is used for object detection and the results are shown below. Multiple objects are recognized in a single frame and the target object (Ball and Cup) can be localized within.

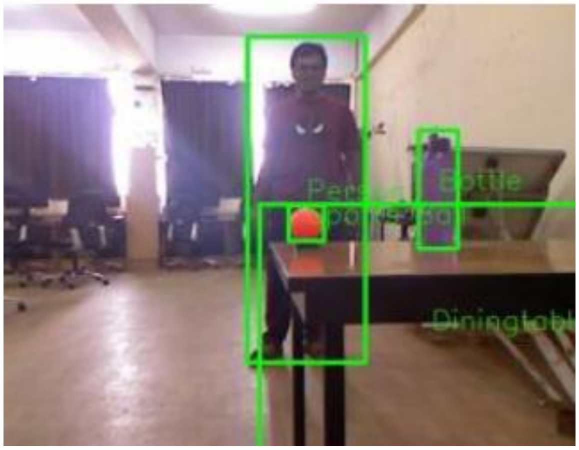


Fig10: YOLO V3 Implementation

3. Object Manipulation:

The above figure shows the integration of the camera and the kinematics of the arm, the object (ball) is located and is picked. The coordinates of the cup has also been determined in the same image and the picked object is dropped into the cup, by obtaining the kinematic solution for the cup.

Fig11: Object Manipulation

7. IMPLEMENTATION STAGE – 2

In stage 2, our work has been focused on the development of the Tic-Tac-Toe game for human Robot Interaction and we have simultaneously developed tools such as the "ARDOP kinematics Simulator" and the "ardop arm" library, we would like to open source in Stage III. The following section gives a brief description of the progress in stage 2.

7.1 Tic-Tac-Toe

Tic-tac-toe is a **2-player** combinatorial game, where the players X and O take turns filling a 3 times 3×3 grid. Whoever places three respective marks in a vertical, horizontal, or diagonal manner will be the winner of the game. When all 9 squares are full, the game is over. If no player has 3 marks in a row, the game ends in a tie and nobody wins.

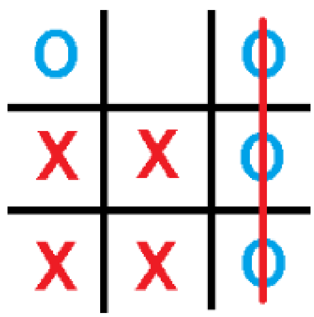


Fig12: Tic-Tac-Toe

7.2 Strategy

On playing tic-tac-toe, tactics and strategy are a must in order to win. The simplest tactic is to complete a three-in-a-row: if you have two of your symbols lined up in a row (either horizontal, vertical, or diagonal) and the remaining square is empty, giving you the win. If you have two of your symbols lined up, and the last square is empty, this is called a **threat**. The above tactic gives rise to **blocking**: if the opponent has two of their symbols lined up and the remaining square is empty, you must block it by playing on the last remaining square. Given the blocking tactic, it's pretty natural to consider creating two different threats so that the opponent cannot block both at the same time. This is called a **fork** (as it has two (or more) directions, like a dining fork splitting from one handle to several times; also compare with fork in chess, which is to make a threat in multiple directions). On the defensive side we must prevent fork moves from the opponent (Human). For a fork action: It is necessary that the center of the grid and either of the corners are filled with the same symbols, and an edge is occupied by the other symbol. Hence for defense we (computer) check if there are any vacancies in the corner once the center is filled (by human) and block the corner.

```
Algorithm 1: Tick Tack Toe Logic
Result: ARDOP responds to Human Move
Board - initialization;
while True do
    Subscribe to Location-index and enter CallBack;
    CallBack(Location-index);
    if ARDOP-Won == True then
        Exit();
    end
    if space-is-free(Location-index) == True then
        insert-letter ('O' , Location-index );
        if not - Human-won then
            ARDOP-Move()
        end
    end
    ARDOP-Move();
      Possible-moves = empty-index(Board);
      move =0;
      for let in [ 'X' , 'O];
        for i in Possible-moves ;
          Board-copy = board[:];
          Board-copy = let;
          if winner-detected(board, let) == True then
              move = i ;
              return move ;
    end
    corners-array = corners(Possible-moves);
    edges-array = edges(Possible-moves);
    if corners-available(corners) == True then
        move = select-random-in(corners-array);
        return move;
    end
    if middle-cell-available(corners) == True then
        move = 5;
        return move;
    end
    if edges-available(edges) == True then
        move = select-random-in(edges-array);
        return move;
    end
    r
end
```

Algorithm 1: Logic of the Tic-Tac-Toe solution

7.3 Tic–Tac–Toe Game with Move Recognition and Logical Response:

For Human Robot Interaction, it is necessary to recognize the move the human and logically respond to it, this task has been accomplished using ROS, Opencv and device drives such as Openni. Here the Human is plays “O” and the Computer Plays “X”. Hough Circles is used to detect the move made by human and the location on the board is determined by the Board co-ordinate system. Every cell is assigned an index and the index is further published as a ROS topic, the Tic-Tac-Toe logic program provides a response to this which is updated in the board table. Now the program waits for the next move from the user.

Algorithm 1: Human Move Detection

```
Result: The Human Move is detected and the location is Published as a ROS Topic
initialization;
Location-array = Board-points;
while True do
    Subscribe to Image;
    Enter CallBack();
    Callback(Image);
      gray1 = convert-to-grayscale(Image);
      gray2 = Filter-Image(gray);
      center-x , center-y , radius = Hough-Circles(gray2);
      Detect-Location(center-x , center-y);
    Detect-Location(a,b);
    while i < 9 do
        if center-x-y within upper and Lower bounds of index i then
            Human-move = i;
            location-index = Human-move;
            publish: location-index;
            return location-index;
            i++;
        end
    end
end
```

Algorithm 2: Human Move Detection

7.4 Development of ARDOP Open Source Tools:

The Kinematics simulator and the “ardop arm” library are the tools that we plan to open source. The kinematics simulator is capable of generating the inverse kinematics solution for any ARDOP like robot and also generate trajectory plans using the cubic trajectory planning. A GUI (Graphical User Interface) has been developed where the user can input the dimension of his robot , and the end location in the camera frame. Once he submits these values, it will forward to the trajectory planning window where the user has to input the start and end time for the velocity analysis. The simulation window is shown in Fig13.

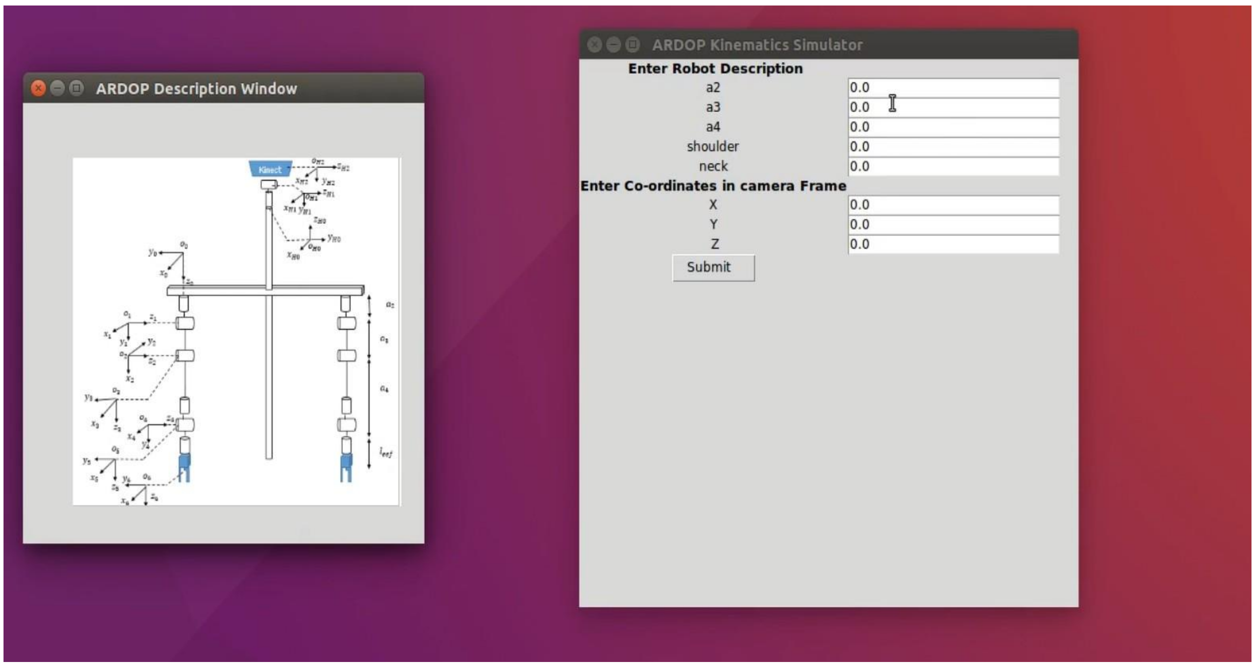


Fig13: ARDOP kinematics Simulator

7.5 Development of the Power Distribution Board:

A Power Distribution Board (PDB) distributes the power to the servos present in the arm according to the requirements. PDB is a circuit that connects all the ground connectors to one another and all the positive connectors together allowing the battery to power all the servos, as per the set requirements. The safety circuit is connected to the PDB, which monitors the present level of power and switches the circuit off during overload. The PDB comprises of Buck and Boost Converters.

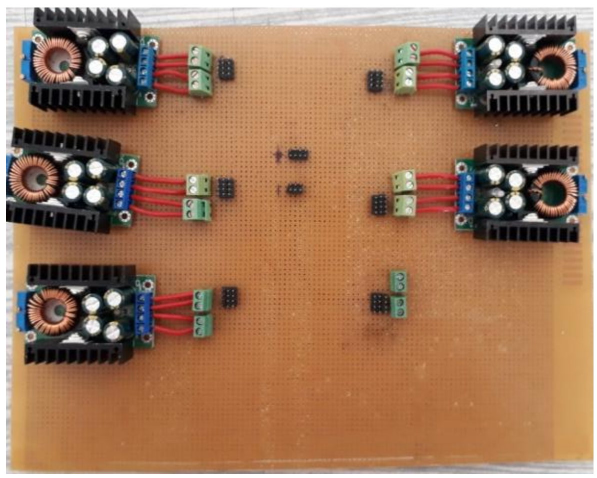


Fig14: The Power Distribution Board

7.6 Implementation Results: Stage II

1. Tic-Tac-Toe (Logic)

In the below figure (Fig 15) the program accepts the input from the keyboard and responds to it. Here the symbol of Human is X, and Computer symbol is O.

```
===========================
Select a number between 1 to 9 to place X5
===========================
   |   |
   |   | O
   |   |
-----------
   |   |
   | X |
   |   |
-----------
   |   |
   |   |
   |   |
===========================
===========================
   |   |
   |   | O
   |   |
-----------
   |   |
   | X |
   |   |
-----------
   |   |
 O |   |
   |   |
===========================
Select a number between 1 to 9 to place X
4
```

Fig15: Logical implementation of Tic-Tac-Toe game.

2. Tic–Tac–Toe Game with Move Recognition and Logical Response

In the below figure (Fig 16) the program accepts the input from the human, detects the circle, identifies the cell and logically responds to it. Here the symbol of Human is O and Computer symbol is X.

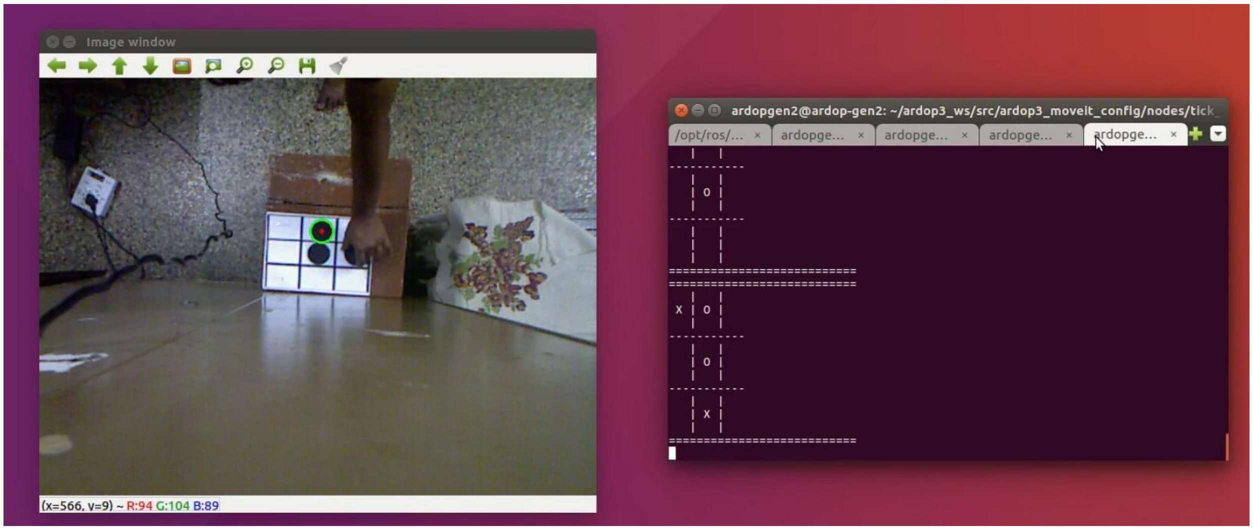


Fig16: Move Detection and Logical Response

3. ARDOP Kinematics Simulator:

The working demonstration of the kinematics simulator is show in Fig17, the user has entered the input dimensions and the co-ordinates in the camera frame the resulting solution is also shown.

Fig18, demonstrates the trajectory planning simulator here the user enters the start and stop time and the resulting cubic trajectory is shown in Fig18.

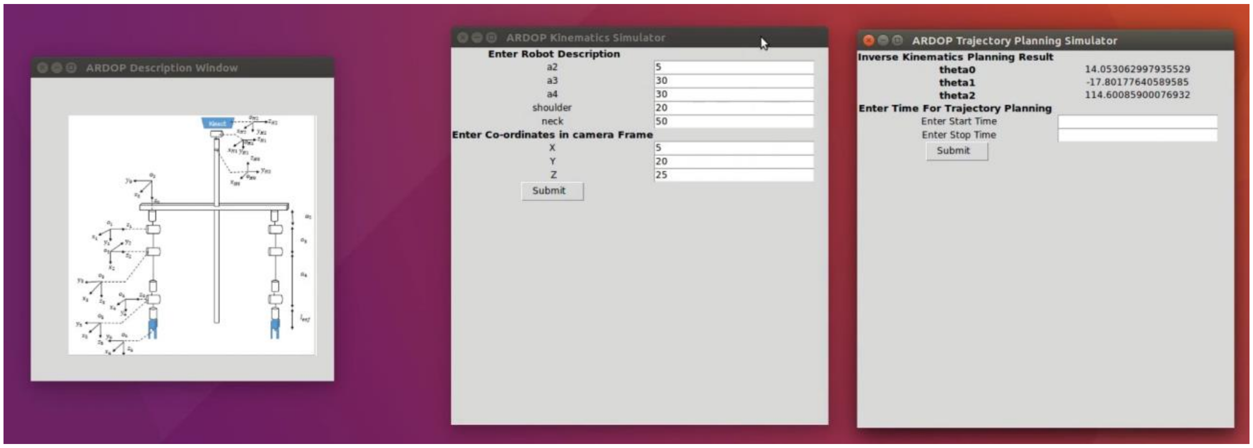


Fig17: Kinematic Solution

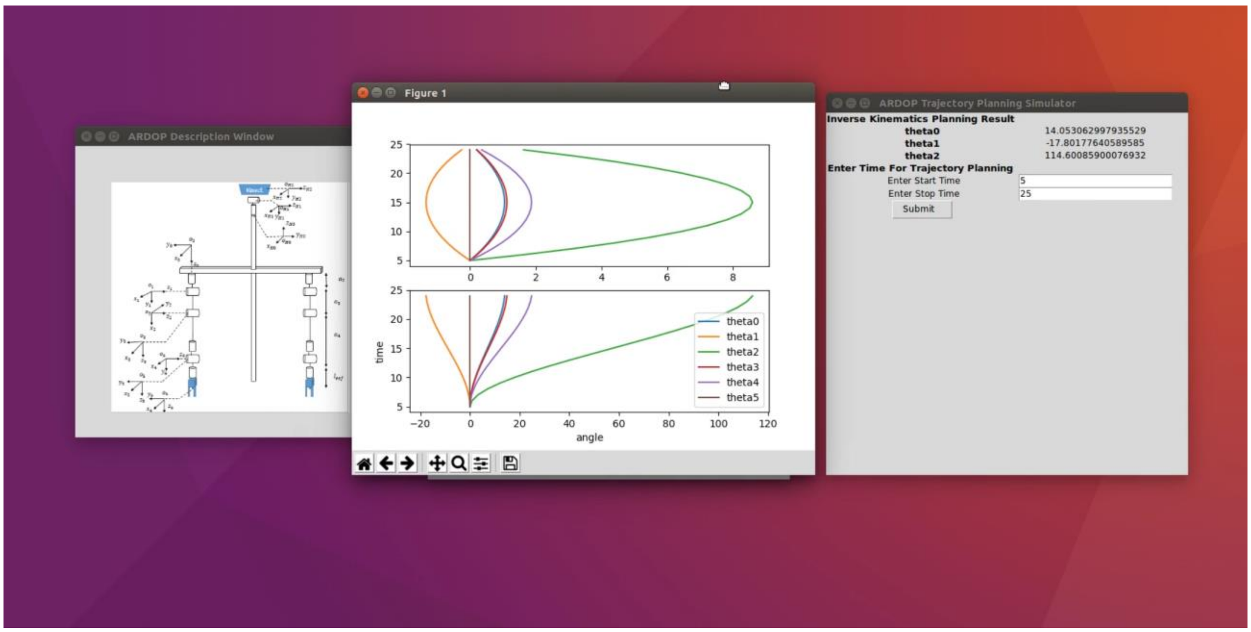


Fig17: Cubic Trajectory Plan Solution

8. IMPLEMENTATION STAGE – 3 (Future Work)

Stage III consists of the work that is to be accomplished in the near future, the stage III description is shown in the Fig 18.

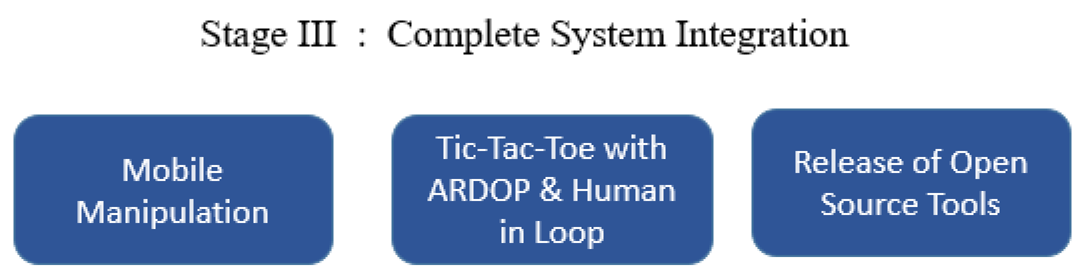


Fig18: Stage III - Implementation

In stage III, we plan to integrate various systems that have been developed till date. The navigation and manipulation system will form the mobile manipulation system, the tic-tac-toe game is to be integrated with the hardware robot for the execution of ARDOP and Human in the loop, Human Robot interaction. Finally, we would open source our designs, programs and development tools.

9. References:

1. Staff (Sandia National Labs) (August 16, 2012), Lifelike, cost effective robotic hand can disable IEDs, R and D Magazine, rdmag.com, retrieved September 13, 2012

2. IEEE Xplore: The Canadarm grasps this boom and can position it in the necessary positions to permit a complete inspection.

3. John Illingworth and Josef Kittler. A survey of the hough transform. Computer vision, graphics, and image processing, 44(1):87116, 1988.

4. David G Lowe. Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 11501157. Ieee, 1999.

5. Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886893. IEEE, 2005.

6. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. Computer visionECCV 2006, pages 404417, 2006

7. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 10971105, 2012.

8. Karun Warrior, Shrenik Muralidhar, Autonomous Robot Development Open Source Platform, BMS College of Engineering, 2017.