e-Yantra Robotics Competition

<Team ID>

Team leader name	**Syed Mohammed Yousuff Hussain**
College	**National Institute of Technology, Karnataka, Surathkal**
Email	**yousuff145@gmail.com**
Theme assigned	**Cargo-alignment.**
Date	**18/12/2014**

Scope (5)

State the scope of the theme assigned to you.

Theme:

1. Cargo alignment.
2. Cubes are covered with papers, black on one side and white on the other. Keeping this as reference, we have to design an automation that aligns all cubes as specified, without misplacing the cubes. Effort will be made to realize the simplest and most efficient automation that would take minimum amount of time and be as robust as possible.
3. It is a highly simplified version of the actual process involved in warehouses and cargo industries where a lot of factors such as handling requirements, storage life, weight, available space, speed of access, availability of power, speed of operation required, etc. need to be accounted.

Purpose of the theme:

1. Crucial process, storage, inspection and retrieval become easy when the cargo is properly aligned.
2. Loss due to damage is minimized.
3. Automation saves human labor and reduces human errors.

Building Modules (5)

Identify the major components in your robotic system provided required for designing a solution to the theme assigned.

Electronic components:

1. The microcontroller, ***Atmega2560*** will be used as the brain of the entire robot. It will be used to sense the environment, compute relevant details and monitor the motion of the bot and its activities. The slave microcontroller ***Atmega8*** will be used to read from sensors that are less important, like the rear IR proximity sensors. It will in turn feed this data to the Atmega2560 whenever needed through the SPI bus. We may not use Atmega8 in addressing the given theme unless it becomes extremely necessary.

2. White line sensors: These sensors will be used extensively throughout the process to sense the black line (or the surface below the bot) and guide the robot through the arena as specified in the algorithm.

3. IR proximity sensors: These sensors will be used to guide the bot along the wall while the bot traverses from one portion of the arena to the other. One of these sensors i.e. the front sensor will also be used to detect the alignment of the blocks in some cases.

4. IR sharp sensor: This sensor in the front will be used to
   1. Detect the cube at a node from a distance.
   2. Position the bot at a precise predetermined distance from the cube. This is an important step in the entire process. If this step is not accomplished in a desired manner, then the position and the alignment of the cube after the process would be seriously affected. (This is further explained under environment sensing section)

5. Extra three IR sensors: These sensors will be placed appropriately on the bot to determine the alignment of the cubes.

6. Extra IR sharp sensor: This sensor will be placed on the right side of the bot. It will be used to detect the entry gate into division D2.

7. Optical Encoders: The data from the two encoders will be used to move the bot to a specified distance or to turn the bot to a particular angle. Both of these will be useful while taking 90 degree turns on the arena. For ex, if a node is detected by the white line sensors, the bot will be

moved forward for precisely a distance (9.5 cm in our case) so that the node lies exactly below the point of rotation of the bot. The bot is then made to turn 90 degrees in the desired direction. This will ensure that the bot stays on the line when taking 90 degree turns. Similar principle can be applied to take U-turns whenever required.

8. AVR ISP Programmer: This device would be used to burn the boot loader on the microcontroller chips. If boot loader is not required, this device would be used to in-system program the microcontrollers using the ISP header on the adapter board.

9. LCD display: This will be used to display relevant details like sensor reading, variable values, etc. while testing and debugging the program.

10. LED bar graph display: Even this module will be used to monitor certain data during testing and debugging.

11. RS232 onboard USB to UART Bridge: This IC will be used while directly connecting the bot to PC via a USB cable. This is very useful while burning programs with the boot loader.

12. Buzzer: This will be used to produce required beeps at particular events as the task requires.

13. USB cable and ISP connector: Will be used while burning programs to the chips and also to serially read data from the microcontroller to the PC for debugging purposes.

Actuators:

1. The two 75 rpm motors: These will be used as the primary actuators in the bot to help in locomotion. Each of these motors is directly attached to a wheel in the rear. A differential drive system will be adapted.

2. Two standard size servos: These two servos will be used to actuate the arm mechanism. In addition, one extra servo (not provided in the pack) may also be used. The three servos serve the following functions:
    1. One servo to actuate (open and close) the gripper.(servo 2)
    2. One servo to rotate the gripper in a horizontal plane.(servo 1)
    3. One servo to rotate the entire arm (with the gripper at the end) in a vertical plane.(servo 3)

    See figure below.

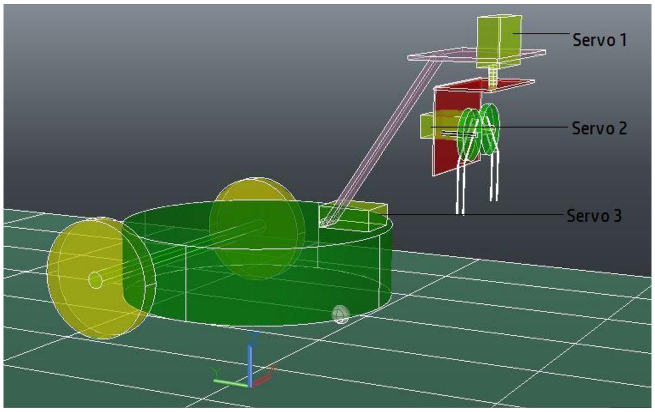

Fig 1.0

Mechanical components:

1. Robot chassis: The chassis is used to hold all the components together.

2. Wheels: The two wheels on the rear will be used for locomotion.

3. Castor wheel: The castor wheel in the front acts as a dummy wheel and support the weight of the bot in the front.

4. Acrylic plate: The transparent acrylic plate will be used as a platform to house all the extra mechanisms on the bot.

5. Spacers: These will be used at appropriate places to flexibly arrange different components on the bot.

6. Apart from what is provided in the kit, we require *spur gears*, aluminum brackets and acrylic plates to build the arm.

Power unit:

1. The main onboard battery will be used to power all the components on the bot including all the motors.

2. The AC-DC adapter will be used for two purposes:
    1. To recharge the battery.
    2. To run the bot on auxiliary power.

Actuators (10)

List all the actuators present on Fire Bird V robot. Besides the existing actuators, please mention the additional actuators that may be required for designing the robotic system in your theme if any.

1. Two 75 rpm motors for the wheels present on the robot.
2. Two standard servos along the bot in the kit.
3. Apart from these actuators, we may use another standard size servo to increase the degrees of freedom of the arm.

Explain the mechanism for controlling the actuators on your robot.

1. 75 rpm motors: The two 75 rpm motors are already interfaced with the robot. To control the direction of these motors, we appropriately change the logic fed into the L293D motor driver from PORTA. The speed of the motors is controlled by PWM technique using the timer 5 in 8 bit fast PWM mode. OCR5A and OCR5B pins are used as the PWM outputs and are connected to the enable pins of the two motors on the driver IC.

2. Servo motors: To interface the servo motors, we use the already provided servo headers on the firebird bot. See fig 1.1 below. The PWM signal required to control the position of the servos is generated from timer 1. The OC1A, OC1B and OC1C pins from the Atmega2560 are connected to the signal pins of the servo headers. All three servos can be connected using the headers onboard.

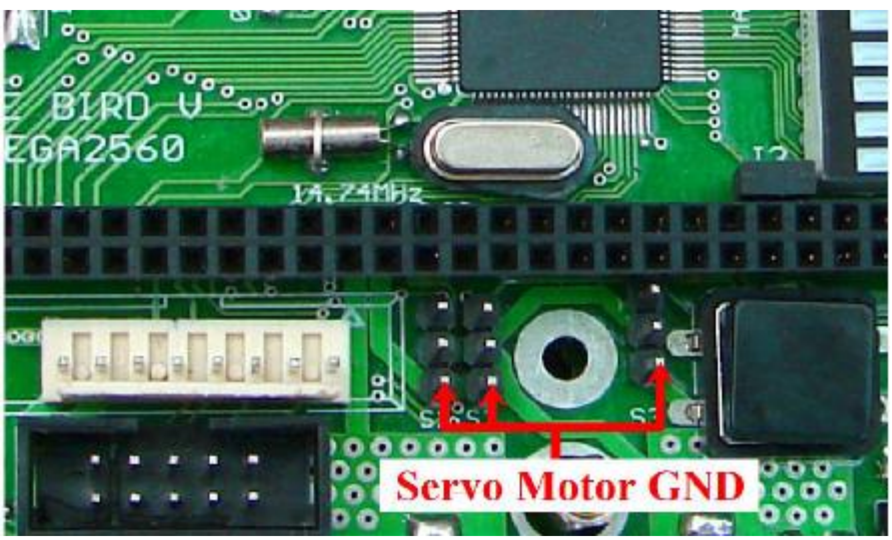

Fig1.1 Servo headers on firebird.

Environment sensing (10)

Explain the functioning of environment sensing technique used by Fire Bird V robot in your theme.

Our robotic system requires sensing the following,

1. The line below to traverse the arena.
2. The wall to move from division D1 to D2.
3. The presence of the blocks.
4. The alignment of the blocks when they are detected.
5. The entry gate into division D2.

The following sensors will be used to sense the things discussed above.

1. White line sensors: These sensors are critical as they help the bot sense the track below. The data from these sensors is used to traverse the arena in a desired pattern.

2. IR proximity sensors: These sensors' data is used to detect and avoid obstacles, mainly while following the wall. Along with following the wall, the front most IR proximity sensor is also used to detect the alignment of the blocks in some cases.

3. IR sharp sensor: This sensor is used to detect the presence of a cube. It is also used to position the bot in front of the cube at an exact distance. If this step is not performed properly, the entire placement of the cube would be spoiled. If the cube is found to be in PI state, the bot will have to turn it by 180 degrees. The rotation of the cube has to be done about a vertical axis passing through the center of the cube. If this axis is shifted away from the center, then after rotation, the cube will be linearly displaced by a distance which is undesirable. So, the placement of the bot in front of the cube should be as precise as possible so that the arm turns the cube about a vertical axis that is as close to the center of the cube as possible.

4. The extra three sensors: These sensors are mainly used to detect the alignment of the cubes. These sensors will be placed on the gripper at an appropriate position to help in detecting the alignment of the blocks.

5. The extra IR sharp sensor: This sensor will be placed on the right side of the bot above IR proximity sensor-5. As the bot follows wall D2, this sensor will detect a missing wall and the bot will turn into D2.

Power Management (5)

Explain the power management system required for a robot in general and for Fire Bird V robot in particular.

Any robot can have a wide range of components with different voltage and current requirement. So, the power management for a robot should make sure that each component is supplied with enough power and required voltage and current to function properly.

Firebird V is powered by 9.6 V NiMH battery that can source 2A of current. It can also be auxiliary powered but the current sourcing is limited to 1A in that case. The power rail is divided into two –

V batt: supplying to all the electronics on the bot. This supply is stable.
V Mot: supplying to the motors that require more current. This rail has extra current sourcing capacity. Also, due to inductive loads, this rail is highly noisy and less stable. The motors take a maximum current of 0.6A each.

There are about 6 voltage regulators, either 5 or 3.3 V onboard, each supplying for different types of components like microcontrollers, sensors, servos, battery monitoring system, Xbee module, etc.

The two motors together may take up to 1.2A. Each servo can have a stall current of up to 1A. So, to limit maximum current drawn from the battery, we try to avoid using the wheel motors and the servo's simultaneously. The arm is also designed in a way that the servo stalls are avoided and the servo draws only quiescent current at rest.

We prefer to use battery power due to

1. High current requirement.
2. Increase flexibility of motion for the bot.

Navigation Scheme (10)

Explain in brief the basic navigation technique for path traversal in the arena. Explain the concept and list the components required for basic navigation.

Division D1

There can be two cases:

1. Block is found on column D or B.
2. Block not found on column D or B.

In case1, align the cube, come back to column C, go to the next row and reach the end. Here, if a cube is present, align it and return to C. If not, then turn left since there may be a cube there also. Align the cube if present, and then continue back to column C. See figure 1.2.

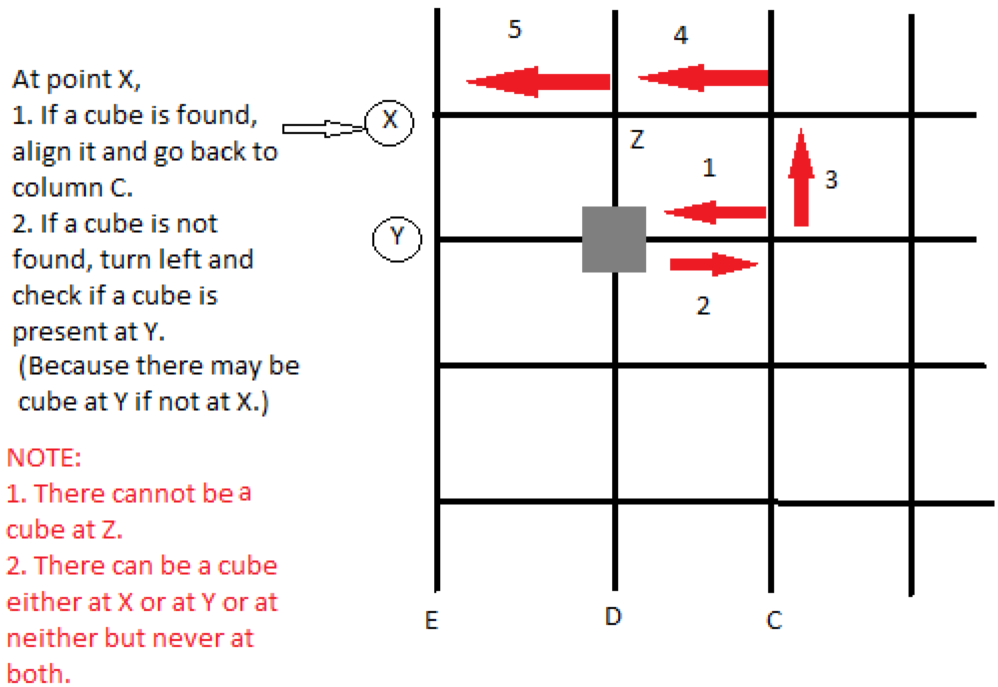

Fig. 1.2

Alternatively, the bot may lift the cube on column D or B and check if a cube is present behind it from there itself. The exact method will be decided during testing. If the bot is able to detect the cube behind by lifting the cube, then this method will be employed. Otherwise, the above discussed scheme will be used.

In case 2, there is nothing to worry about. Just align the cube and return back to column C. See fig 1.3

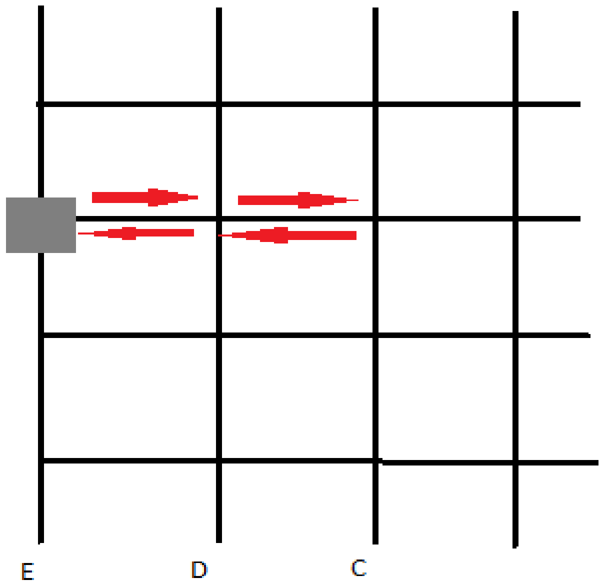

Fig 1.3

Division D2

Navigation in Division D2 would be simple. Keeping column F as the base, the bot will first move to row 6 along column F, then turn right, align blocks on row 6 if any, come back to column F, go to row 5, and continue in the same way until it reaches row 1.The bot will also memorize the positions of the cube on its way. If a cube is not found in row-1, then the bot can easily make its way to the end point, (1, I). Otherwise, the bot will create a path from memory, along which it can move to reach the end point (1, I).

The components required for navigation:

1. Wheels: The two wheels on the rear will be used for locomotion.
2. Castor wheel: The castor wheel in the front acts as a dummy wheel and support the weight of the bot in the front.

Testing your knowledge (Based on theme and rulebook) (10)

Explain in your own words how you will follow the wall and list the components required for wall following.

We use the IR proximity sensors to follow the wall. The first five sensors are used. If a right wall is being followed, sensor-5, sensor-4 and sensor-6 readings are taken and the wheel speeds are altered accordingly as indicated by figures fig 2.1 to fig 2.6. Similarly, if a left wall is being followed, sensor-1, sensor-2 and sensor-8 readings can be used.

Sensor-3 reading is used to detect the obstacle ahead and take turns.

On exiting from D1, the bot will move directly to wall2 and take a left turn and start following the wall. Using the sharp sensor mounted above sensor-5, the entry gate into D2 is detected. On detecting the gate, the bot turns right and continues its way in D2.

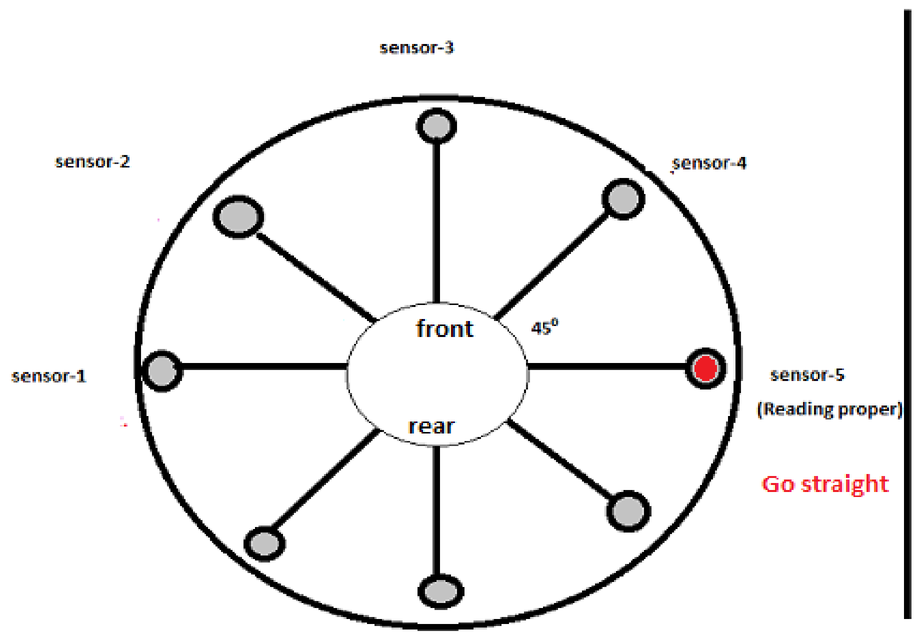

Fig 2.1

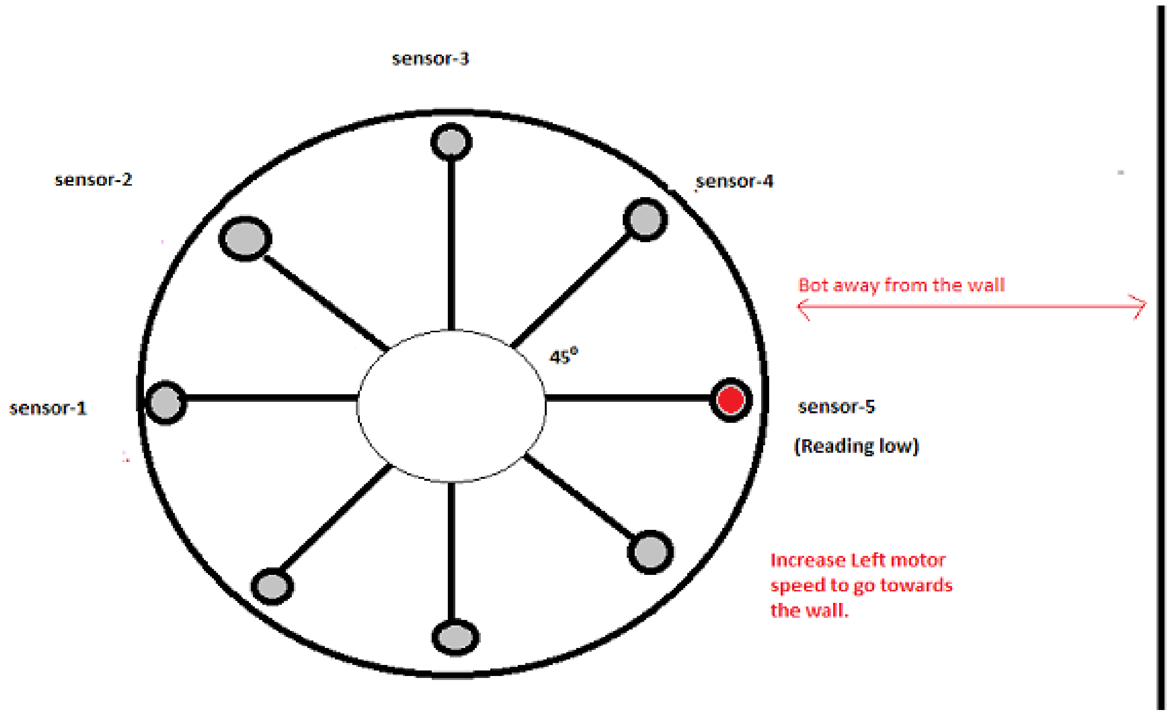
sensor-3
sensor-2
sensor-4
Bot away from the wall
45°
sensor-1
sensor-5
(Reading low)
Increase Left motor speed to go towards the wall.

Fig 2.2

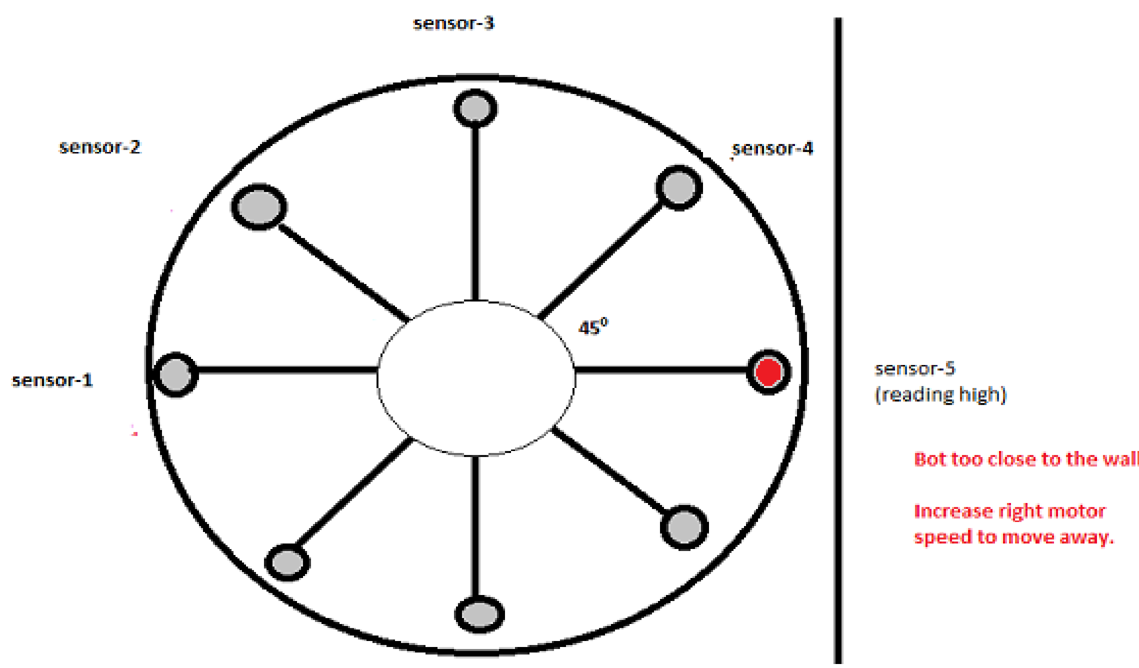
sensor-3
sensor-2
sensor-4
45°
sensor-1
sensor-5
(reading high)
Bot too close to the wall
Increase right motor speed to move away.

Fig 2.3

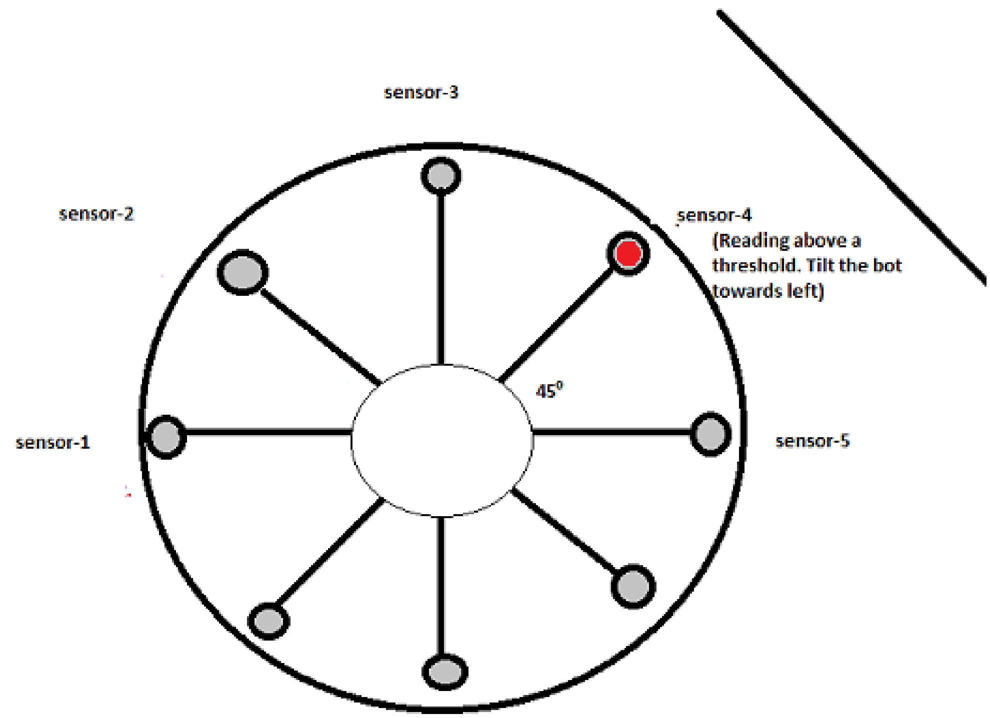
sensor-3
sensor-2
sensor-4
(Reading above a threshold. Tilt the bot towards left)
45⁰
sensor-1
sensor-5

Fig 2.4

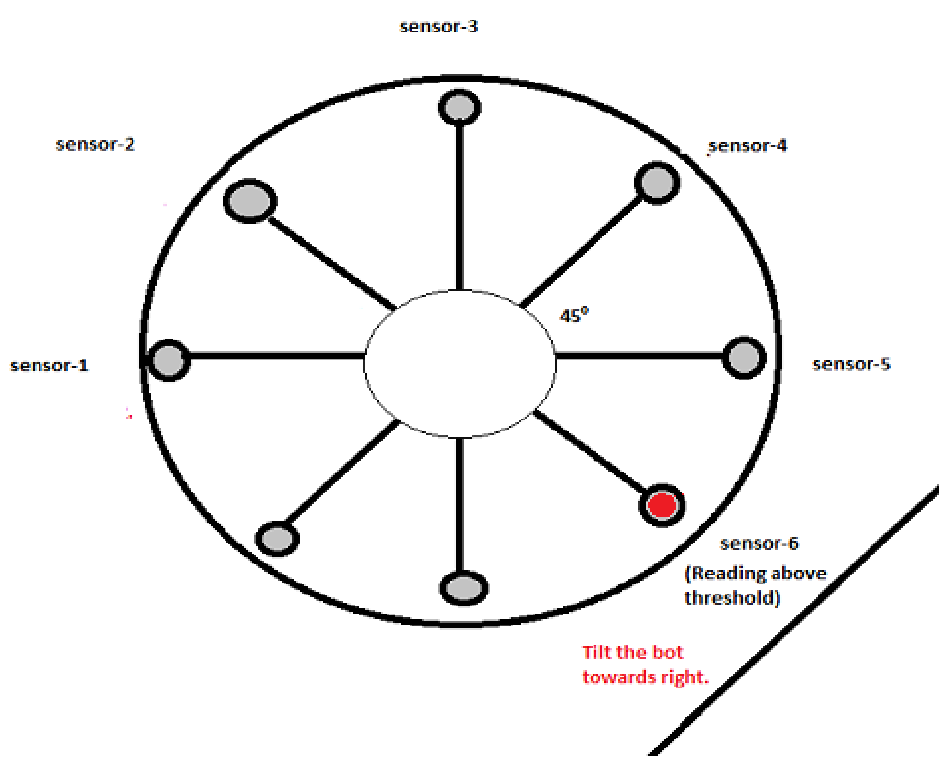
sensor-3
sensor-2
sensor-4
45⁰
sensor-1
sensor-5
sensor-6
(Reading above threshold)
Tilt the bot towards right.

Fig 2.5

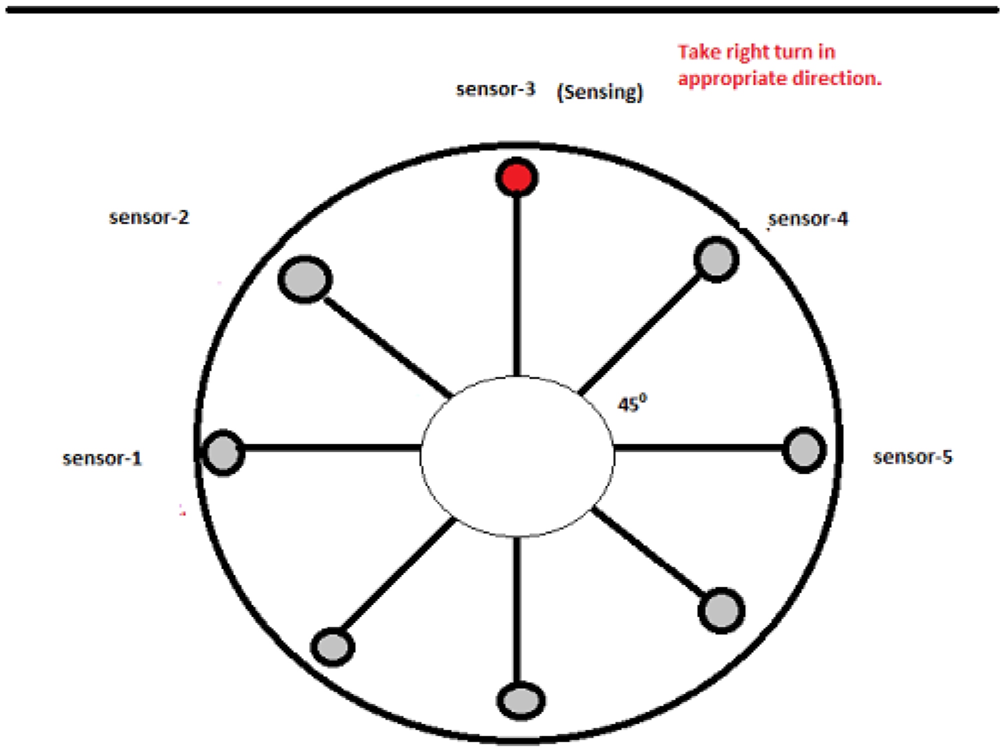

Fig 2.6

Explain how you will make sure that blocks are Placed Correctly (in the state of PC) in the arena; List the components required to detect placement of blocks.

Firstly, the initial alignment of the cubes is detected as described below. Then, if the blocks are found to be in PI state, the arm is actuated appropriately to turn the cubes by 180° to bring them in PC state.

Following the navigation scheme described above, the bot may reach the cubes from two possible directions-

1. When the block is on column D of B, in division D1, the bot will approach the cube from the front. See fig 3.1. In this case, the initial alignment can be easily detected by using only the front most IR proximity sensor (sensor-3).

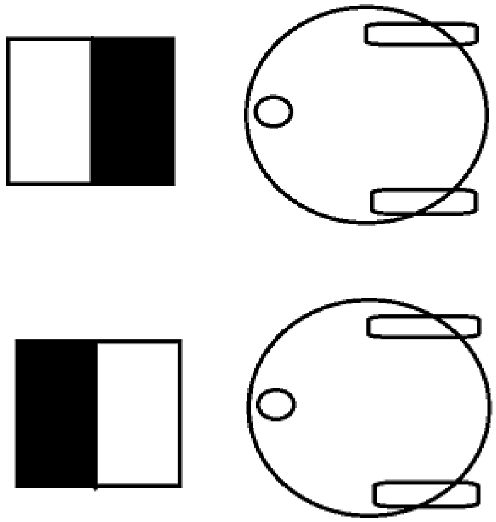

Fig. 3.1

2. When the cube is on column A or E and behind another cube in Division D1, or if the cube is in division D2, then the bot will have to sense the alignment from the side. See fig 3.2. In this case, we use the extra three sensors provided. Using the data from these sensors, the blocks are aligned properly.

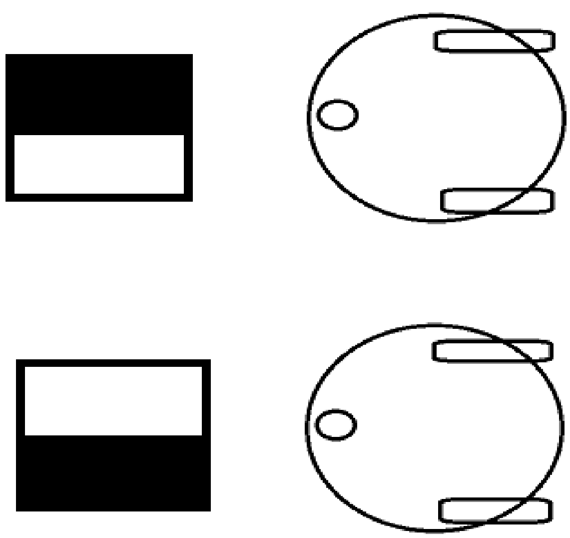

Fig. 3.2

When a cube is found to be incorrectly placed, the bot will place it correctly as described below.
See fig 3.3. (Circular parts in green are meshed spur gears). The arm will be placed such that the block is gripped from the top. Servo 2 will actuate the grippers and the block will be gripped. Then servo 1 will turn the entire gripper along with the block by 180º (Note that the block is turned about a vertical axis). Then servo 2 will release the cube. Thus, the cube will be correctly placed.

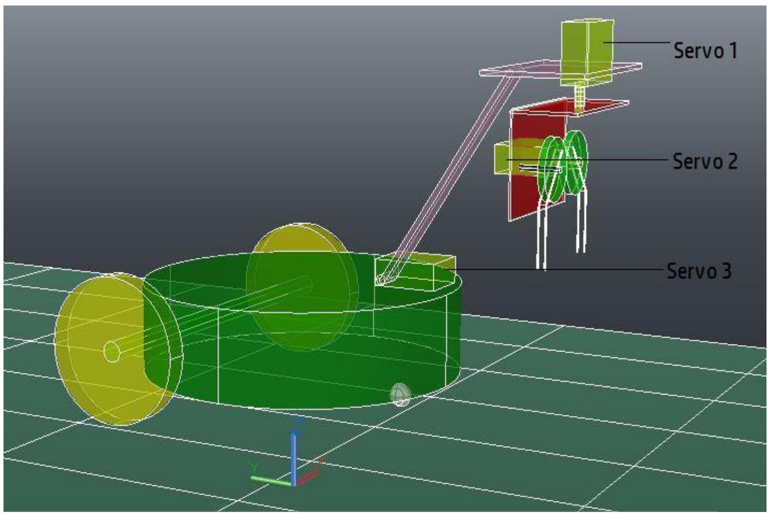

Fig 3.3

Challenges (5)

What are the major challenges that you can anticipate in addressing this theme?

1. Calibration of sensors: We plan to spend a lot of time in calibrating the sensors and understanding their behavior in all possible conditions. This step is going to be very crucial in completing the task and we consider it to be a major challenge.
2. Determining the optimum value of constants: The implementation of the developed navigation scheme has to be very sensitive. The constants used in the calculation of the motor speeds need to be refined very carefully during testing. We anticipate a lot of effort in doing so.
3. Handling errors: Our algorithm is not self-correcting. At any point, if the bot behaves in non-accordance to the algorithm due to some reason, then our algorithm does not guarantee that the rest of the task would be correctly completed. Adding a self-correcting component to our algorithm is another challenge we anticipate.
4. Arm design: The arm is inevitably the most important feature of the bot in completing this task. Since we may not use sophisticated machinery to build the components for the arm, we have to build an arm with maximum precision with available materials and tools. To build a robust arm as per the design with minimum error is going to be a challenge.
5. Sensitive arm: The arm has to be very sensitive so that the cubes are not misplaced during the task. To determine the exact profile of the gripper to be used is another challenge.
6. 90º turns: Taking exact 90º turns on the arena, without considerably displacing the robot away from the line is another challenge we anticipate.
7. Upgrading the algorithm: Our algorithm follows 'brute force' principle. It tends to take more time. Adding smartness to the algorithm in a manner that saves time is going to be another challenge for the team.

(NOTE: The algorithm employed, including the arm design and navigation scheme, is subject to improvement. Depending on the results obtained during testing, slight changes may be introduced in the algorithm and placement of the extra sensors)