Lidar Idea

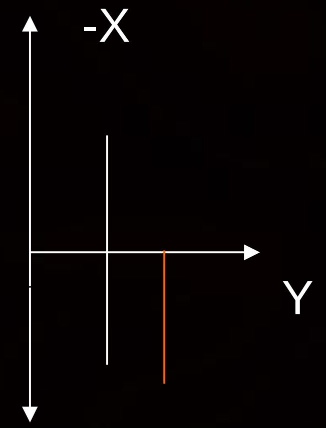
By: Douglas Guerra

Problem Description

- The hokuyo lidar gives us a 2D scan of the world around it. Due to this limitation it is necessary for us to have the lidar target a specific spot on the mining area, which we could use to spot the location of the robot.
- My idea is to place a rectangular board on the back of the robot. This way we can filter out a cluster of points that aren't in a straight line and are not the length of the board.
- By doing this we can ensure that we don't misidentify objects in the arena.
- Using the points on the line we can orient the robot by using the slope of the resulting line and the distance of the midpoint of the line from the lidar.

Explanation

- In order for a line to be centered about the y-axis it is required for points on the line to be both in the positive and negative x-axis.



- In the depiction above we can see that orange line is not centered about the x-axis and that all

of its points are in the negative x-axis.

- Meanwhile we can assume that the white line is nearly centered about the y-axis because the line has points that are in the negative-x axis and points on the positive x-axis.
- Using the midpoint of line can be used as a second check, if the midpoint is directly on the y-axis then the line is centered about the y-axis

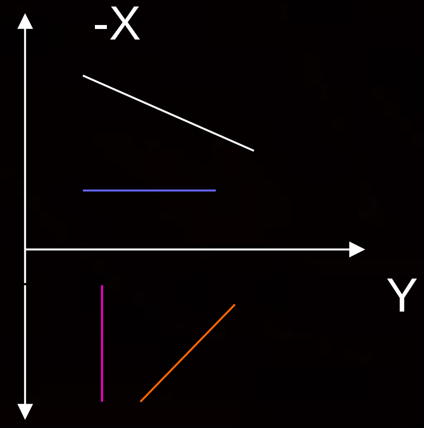
Explanation

Ideally, the robot will be placed in the arena in a direction that the target line on the robot faces the lidar. Since it is possible for target point to not be seen by lidar initially we will need to rotate the robot in place until the rectangular board is scanned.

We have to ensure that the lidar is indeed scanning the correct target point on the robot and not a random cluster of points.

Thus we should keep rotating until the slope of the line is parallel to the x-axis or to the y-axis, which would mean that the line is perfectly straight

Explanation

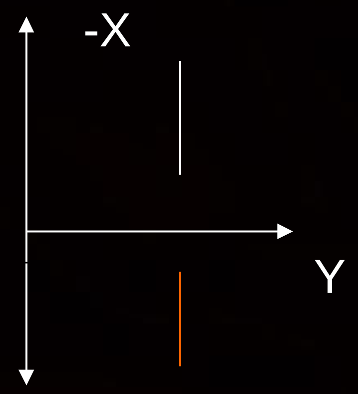


- We can see that white line has a steep positive slope and thus would have to be rotated counter-clockwise until we have an infinite slope, parallel to the y-axis.
  - Similar to the blue line
- Meanwhile, the orange line has a steep negative slope and thus would have to be rotated counter-clockwise until it is parallel to x-axis.
  - Similar to the pink line

Explanation

- Then we measure the length of the cluster of points, which should roughly be the same length as the board on the robot.
- If it is not then we have scanned the wrong object and should keep looking.
- If we have correctly scanned the plate on the robot, then we calculate the centroid of the plate and use the x-coordinate.
  - Either positive or negative to tell us whether we are on the right or left of the lidar

Explanation



We can easily figure out that the white line, hence the robot, is on the left of the origin because all of its points are in the negative x-axis.

- Therefore we need to move the robot right.

- Meanwhile, the orange line has all the points on the positive x-axis meaning we need to move the robot left.

Explanation

- Taking into account the offset between the lidar and the center of the arena we can calculate the distance we need to travel to center the robot
- We then just simply have to turn the robot correctly and travel for a designated period of time.
  - This could be determined by placing plates on the side of the robot.
  - We then just have to move the robot until the centroid of these plates are aligned with the origin.
  - Thus we constantly have a check if we are moving in the correct direction and know if we are aligned without having to rotate the robot back and look for the metal plate on the back of the robot.

Explanation

- We cannot just use the plates on the side of the robot because to know our orientation it would require us to slightly move in one direction and see if we went farther or closer to the center of the arena.
  - This could cause us to hit the wall unintentionally.
- The metal plate on the back of the robot will not interfere with the dumping mechanism as the hopper pivots on columns in the back and does not occupy the entire backside of the robot.
  - Also the plate has no need to be large enough to interfere with the pivoting points for the hopper.

Technical Explanation

The data points given by the hokuyo lidar can be filtered and processed by using Point Cloud Library.

The Point Cloud Library (PCL) is a standalone, large scale, open project for 2D/3D image and point cloud processing.

Technical Explanation

The hokuyo lidar gives us a 270° scan of up to 5 meters.

In the arena, the width of the starting region is 3.78 meters.

If we put the lidar on the rightmost part of the collector bin that would mean that we are positioned 0.7875 meters left of the center of the arena.

Meaning that from the perspective of the lidar there are 2.6775 meters to the left and 1.1025 meters to the right until we scan the walls.

Technical Explanation

- Therefore we can run a PassThrough filter on the set of data points.
  - This filter eliminates points that are not within a given range.
- This PassThrough filter will ensure that we are only analyzing points that are within the width of the arena.
- Point Cloud Library provides us the library to run a PassThrough filter with the range being:
  - All points on the y-axis (the axis that comes out from the lidar)
  - On the x-axis we only need points within the set [-2.6775, 1.1025]
- The time complexity of this function is O(n), where n represents the points in the data set.

Technical Explanation

- After running the PassThrough filter we ensure that we are only analyzing data points within the confines of the arena.
- Then on data set we pass a Statistical Outlier Removal Filter
  - This filter will eliminate noisy measurements by performing a statistical analysis on each point's neighborhood and trimming points which do not meet the criteria.
- Thus removing points that do not reference objects, such as sand, dirt or any other particles.
- Point Cloud Library provides the library and designated functions in order to implement a Statistical Outlier Removal Fitler.
  - The time complexity of this function is O(nlog(n)), where n represents the points in the data set.

Technical Explanation

- Having run the designated filters we are left with a data set that may represent more than one object.
- At this point we are looking for a metal plate on the robot, which in 2-dimensional world would look like a line.
  - Meaning we are trying to extract a line from the cluster of points

Technical Explanation

- Point Cloud Library gives us functions to find lines in a cluster:
  - SACMODEL_LINE, finds all lines in the cluster
  - RANSAC_MODEL_LINE, returns the best line
- Point Cloud Library also gives us a way to determine if a line is parallel to an axis:
  - SACMODEL_PARALLEL_LINE, determines a line parallel to a given axis

Technical Explanation

Using the RANSAC_MODEL_LINE we can extract the best line in a cluster of points and then we can analyze the line and move on if it is not what we are looking for.

In order to check if our line represents our metal plate then we can determine the minimum and maximum points on the line and determine the length of the line by calculating the euclidean distance between the points.

Technical Explanation

There is a function in PCL that returns the min and max of cluster of points (pcl::getMinMax)

Once we have the coordinates of the minimum and maximum points we can determine the length of the line by calculating the euclidean distance between the points.

Using the length of the line we can compare it to the length of the metal plates on the robot as a check to see if we are scanning the correct object.

Technical Explanation

- If the length of the line we have scanned does not match that of the metal plate then we move on the next best line in the cluster of points.
  - We continue doing so for a set amount of times, and if we still do not find the correct object then we rotate our robot and repeat the process.
- Once we have found a line that meets our criteria then we move on to figure out which way our robot needs to rotate and then move toward the center.

Technical Explanation

- Using the SACMODEL_LINE and SACMODEL_PARALLEL_LINE functions in PCL we can determine the direction of the line.
  - Meaning we can determine which axis we are oriented on and thus can determine the direction we need to rotate.
- Using the PCL function Compute3DCentroid we can determine the midpoint of our line and it returns a vector from the origin to this point.
  - Using this vector we can determine whether or not we are to the right or left of the center of the arena.
  - **Note**: For PCL the origin is the base of the lidar but since we know the offset between the position of the lidar and the center of the arena we can use the center of the arena as our origin but shifting our data points by 0.7875 meters

Technical Explanation

- The slope of the metal plate can be calculated using the min and max points on the line
- Knowing the slope with reference to the x and y plane will allow us to determine how to rotate so we can move toward the center correctly
- We can rotate the robot until our line is nearly parallel to the x-axis or to the y-axis
- In order to determine if we are parallel to an axis we can use the Point Cloud Library function SACMODEL_PARALLEL LINE
  - This function determines if a line is parallel to an axis, with a maximum specified angular deviation

Technical Explanation

If the metal plate is parallel to the y-axis, then we move either forward or backward depending on the vector returned by pcl::compute3DCentroid

If the metal plate is parallel to the x-axis, then we have to rotate and move forward or backward depending on the rotation done

In both cases we then want to use the sides of the robot to ensure two things:

1. We are moving the correct direction
2. To stop moving and rotate when we have reached the center of the arena

Technical Explanation

To identify the sides of the robot we will also place metal plates of desired length.

- We then will use the same filters as before Passthrough and StatisticalOutlierRemoval
- Then we will use RANSAC_MODEL_LINE to find the best line and check if its the correct object by calculating the length of the object by determining the euclidean distance between the min and max points
- If it is our line we calculate the centroid and continue following it until we have made it to the center of the arena
- Then we rotate the robot and are now in the center of the arena and thus could move forward to the mining area

Alternate Options

- Doing a 3-dimensional scan:
  - It is possible to do a 3D scan but this means that the odroid will have to process more points per scan. The filters will have to parse through more points.
  - We can run a hough transform to determine the position of the robot.
  - This seems inefficient because we do not need that many points to determine where we are, when moving in a 2-dimensional plane
- Using metal plates just on the side of the robot:
  - We will not be able to determine the front or back of the robot if we put plates on the both sides of the robot.
  - Meaning that in order to determine the direction we are facing we need to move slightly and the robot might accidentally hit the wall

Execution Plan

Week 1 & 2: Run a StatisticalOutlierRemoval and PassThrough filter

Week 2 & 3: Attempt to detect an object and determine its length.

Week 4: Within a dense cluster of points extract the desired object.

Week 5 & 6: Having extracted the object from a cluster, determine the angle of the line or if it is parallel to a specified axis.

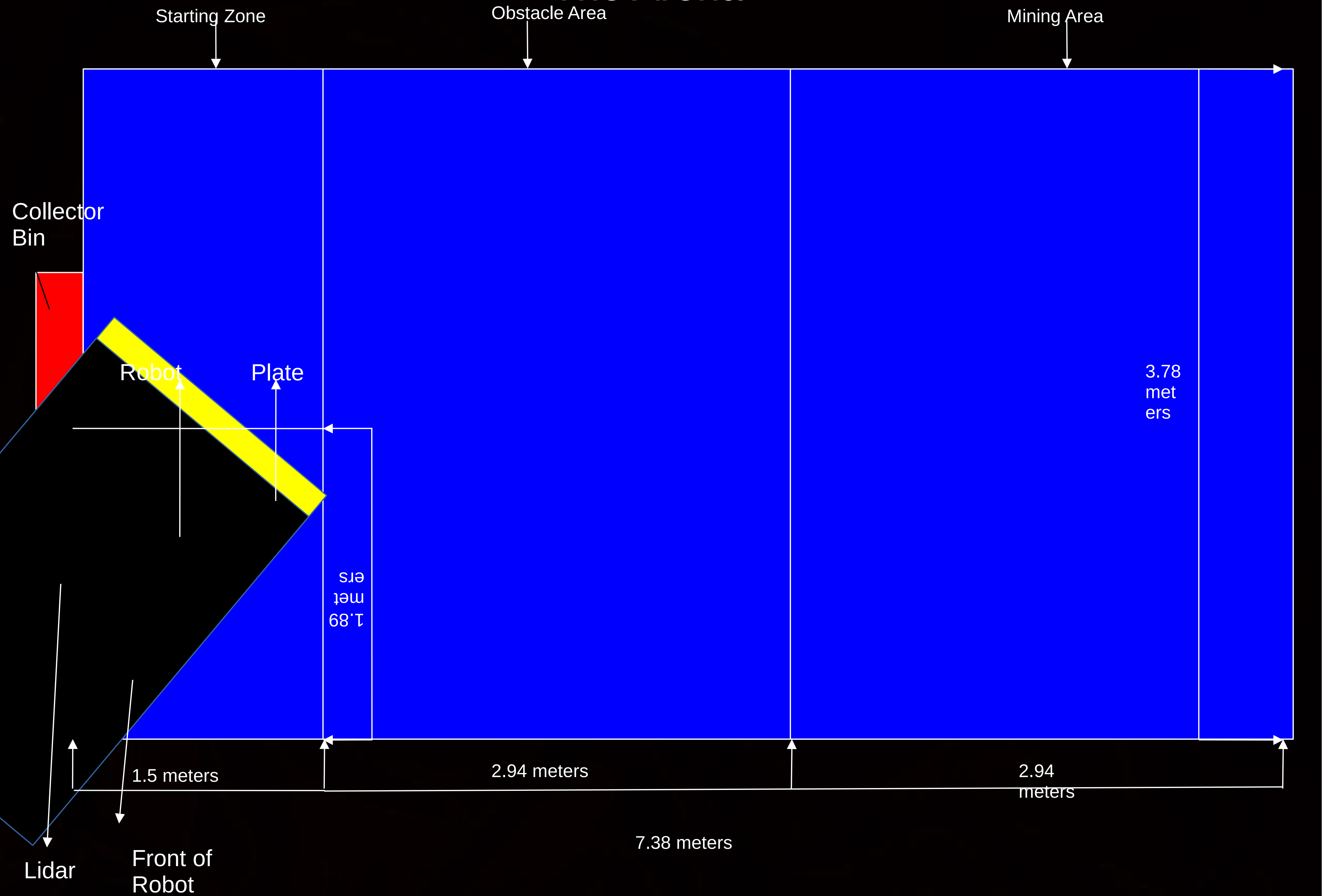
Week 7 & 8: Determine the centroid of an object and track the object as it is in motion

Week 9 – 12: Testing

Positioning of the Lidar

- The dimensions of the arena are 7.38 meters long by 3.78 meters wide. The arena is split into three sections, which are the starting zone, the obstacle area and the mining area.
  - The starting zone is split down the middle into two halves, each half is identical has a dimension of 1.5 meters long by 1.89 meters wide.
- The walls around the starting zone are a height of 0.55 meters.
  - Meaning that base of the lidar will be placed at a height of 0.55 meters.
- The height of the lidar is 0.07 meters, meaning that standing upright the total scan will be made at an approximate height of 0.62 meters ± 0.001 meters.
  - Depending on the angle the height of lidar will vary slightly.

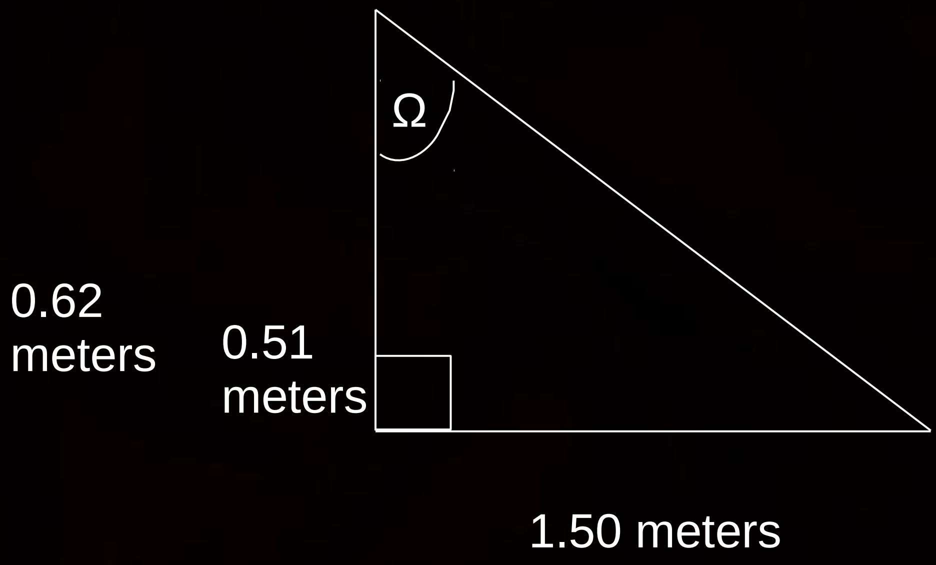
The Arena



Starting Zone
Obstacle Area
Mining Area
Collector Bin
Robot
Plate
3.78 met ers
1.89 met ers
1.5 meters
2.94 meters
2.94 meters
7.38 meters
Lidar
Front of Robot

Positioning of the Lidar

- For calculation purposes, the lidar is assumed to placed at a height of 0.62 meters.
- The dimensions of the robot are 1.5 meters length x 0.75 meters wide x 0.75 meters height.
- The placement of the plate on the robot will be at a height of 0.51 meters ± 0.01 meters.
- With the starting zone being 1.5 meters of length it leaves us with the following right triangle:



Positioning of the Lidar

- Resulting in the angle, Ω, to be precisely 85.8°. This angle could be approximated to 86°, which will result in a percentage error of 0.233%.
- The plate of the robot should ideally be 0.15 meters long and the width of the plate should be the necessary width to attach it to the robot.
  - The width of the plate would not affect the scanning region within the 1.5 meters but knowing the width will allow us to check the cluster of points to ensure that we are scanning the plate and not stray objects in the arena.

Simulation 1

- The robot initially starts on the same half of the arena as the lidar.
- The robot is at a distance of 0.15 meters from the wall of the collector bin

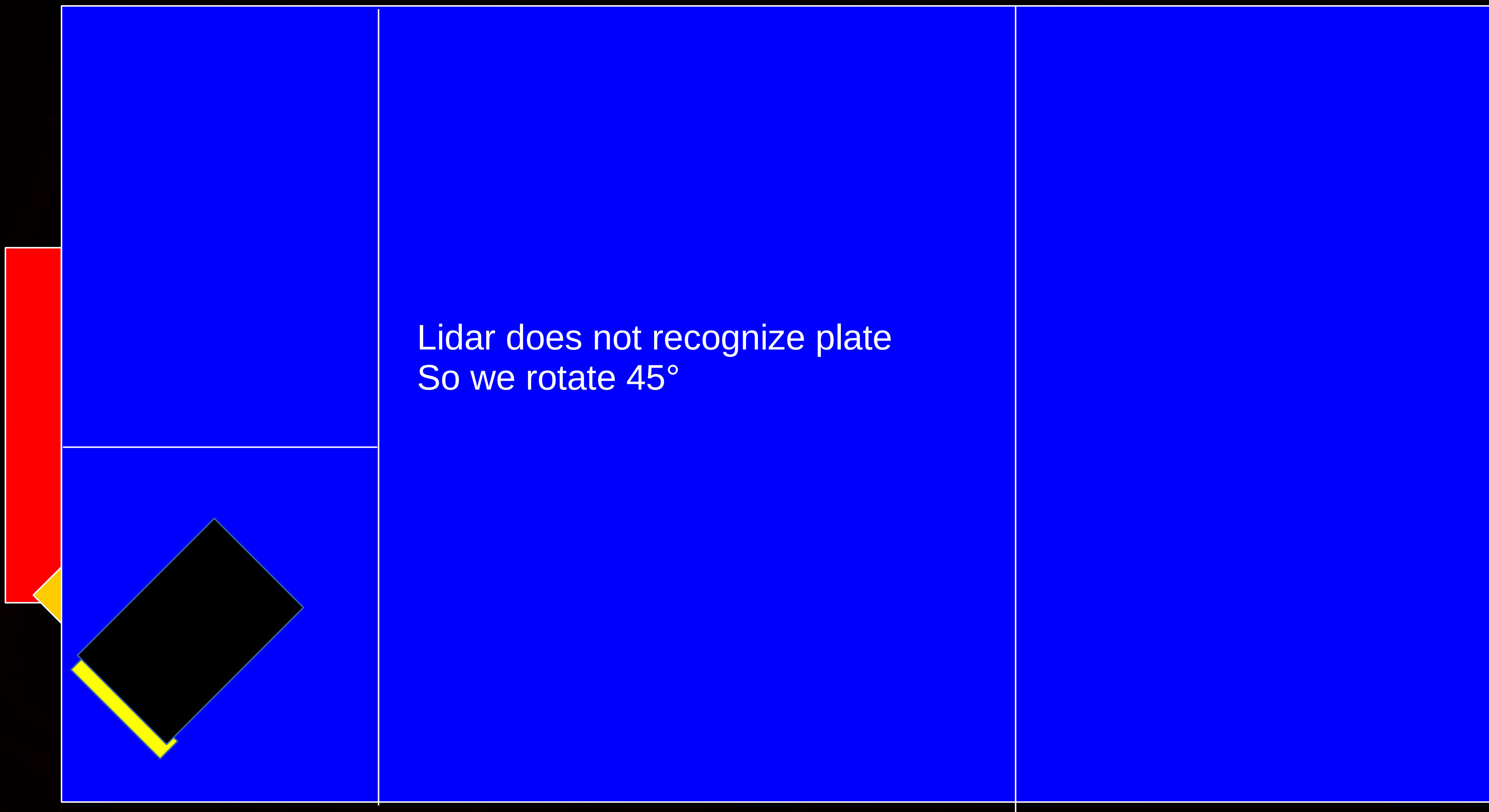
- Initially, the lidar cannot see the plate on the robot
- We rotate the robot and then move towards the center

Simulation 1

Simulation 1

Simulation 1

Simulation 1

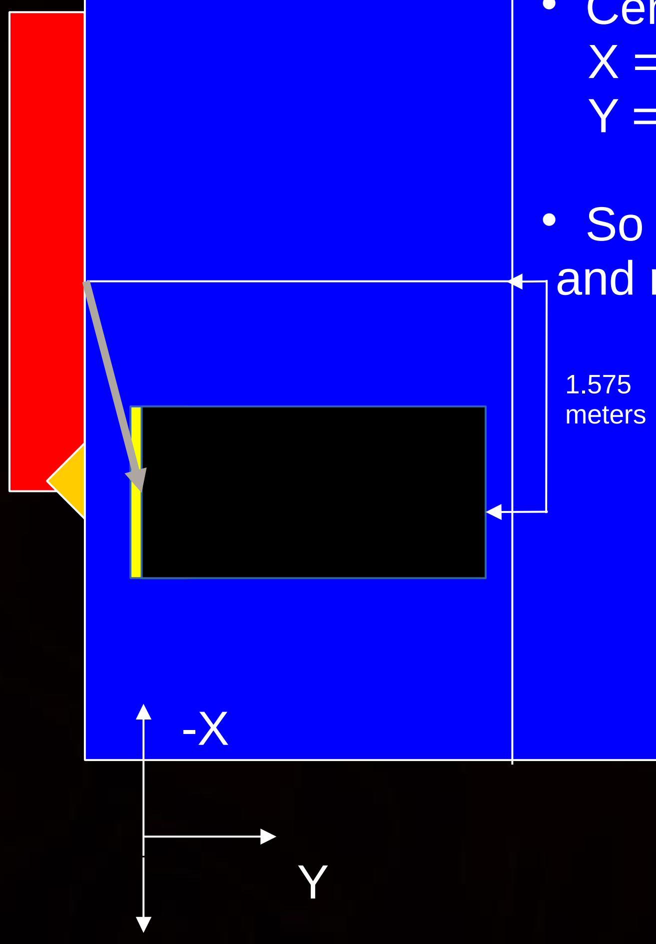


Simulation 1

- Lidar recognizes the plate so we calculate the centroid

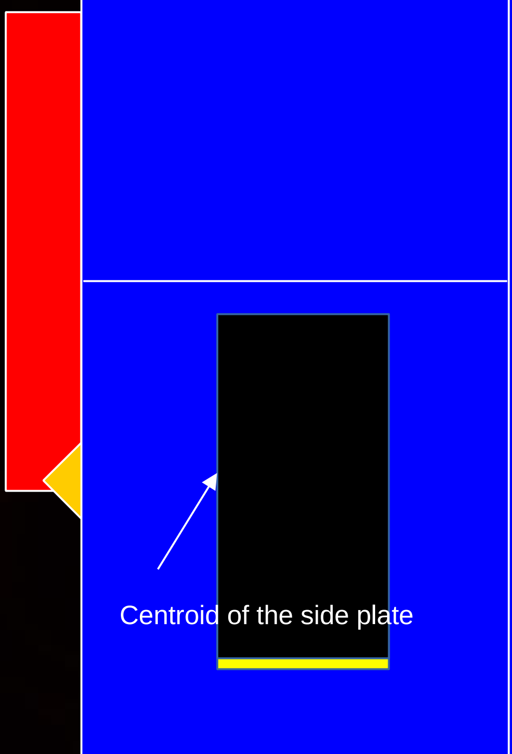
- Centroid of the plate from the origin:
  X = 1.575 meters
  Y = 0.15 meters

- So to be centered we have to rotate counter-clockwise 90° and move approximately 1.575 meters forward



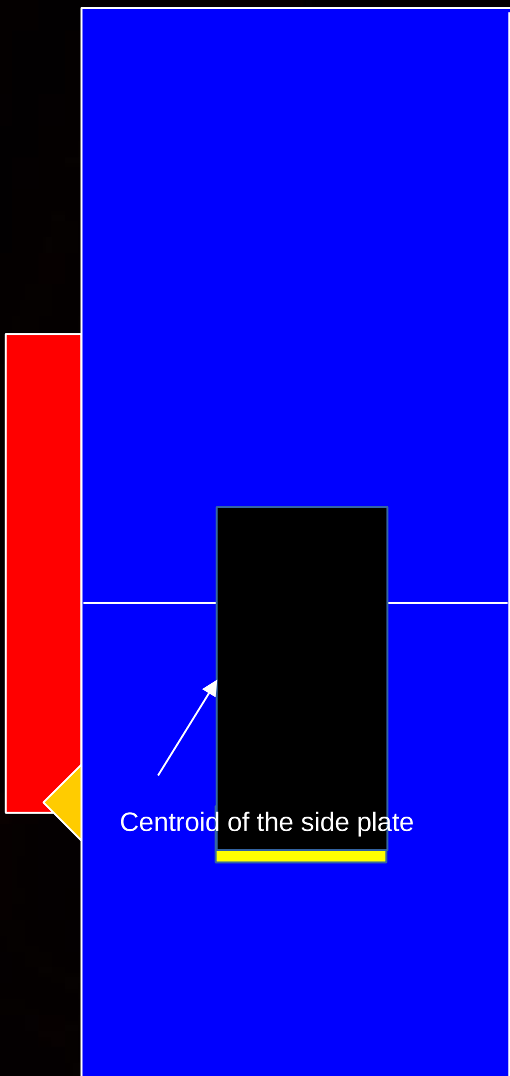
Simulation 1

- Rotated 90° counter-clockwise
- Moving forward



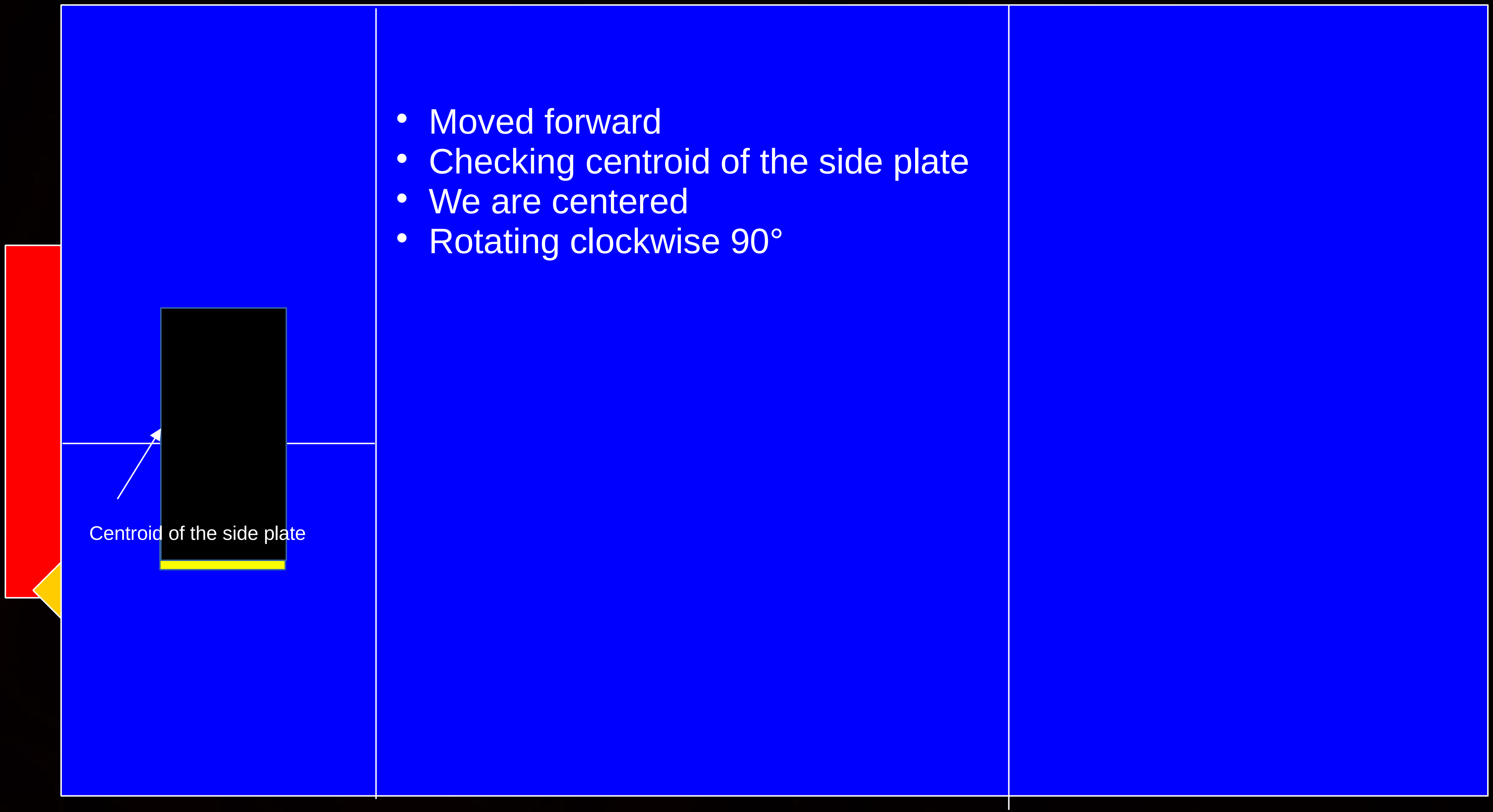
Simulation 1

- Moved forward checking centroid of side plate
- Continue moving forward



Simulation 1

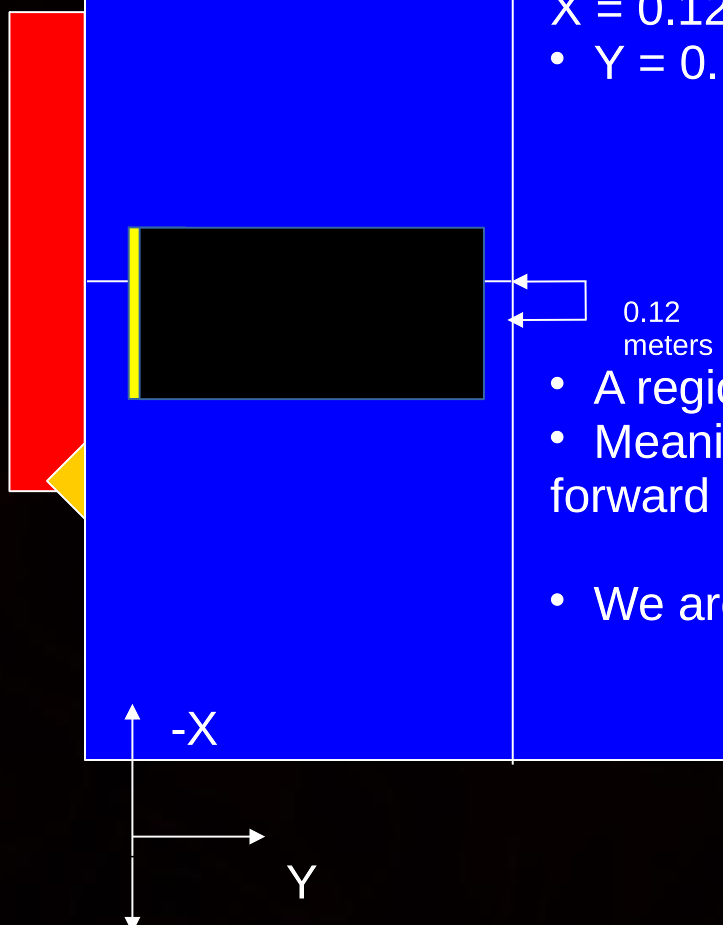
- Moved forward
- Checking centroid of the side plate
- We are centered
- Rotating clockwise 90°



Simulation 1

- Turned left
- Re-calculating position

- Centroid
X = 0.12 meters
- Y = 0.15 meters



- A region that is +/- 0.15 meters from the origin is centered enough
- Meaning that if we are within that region we are ready to move forward

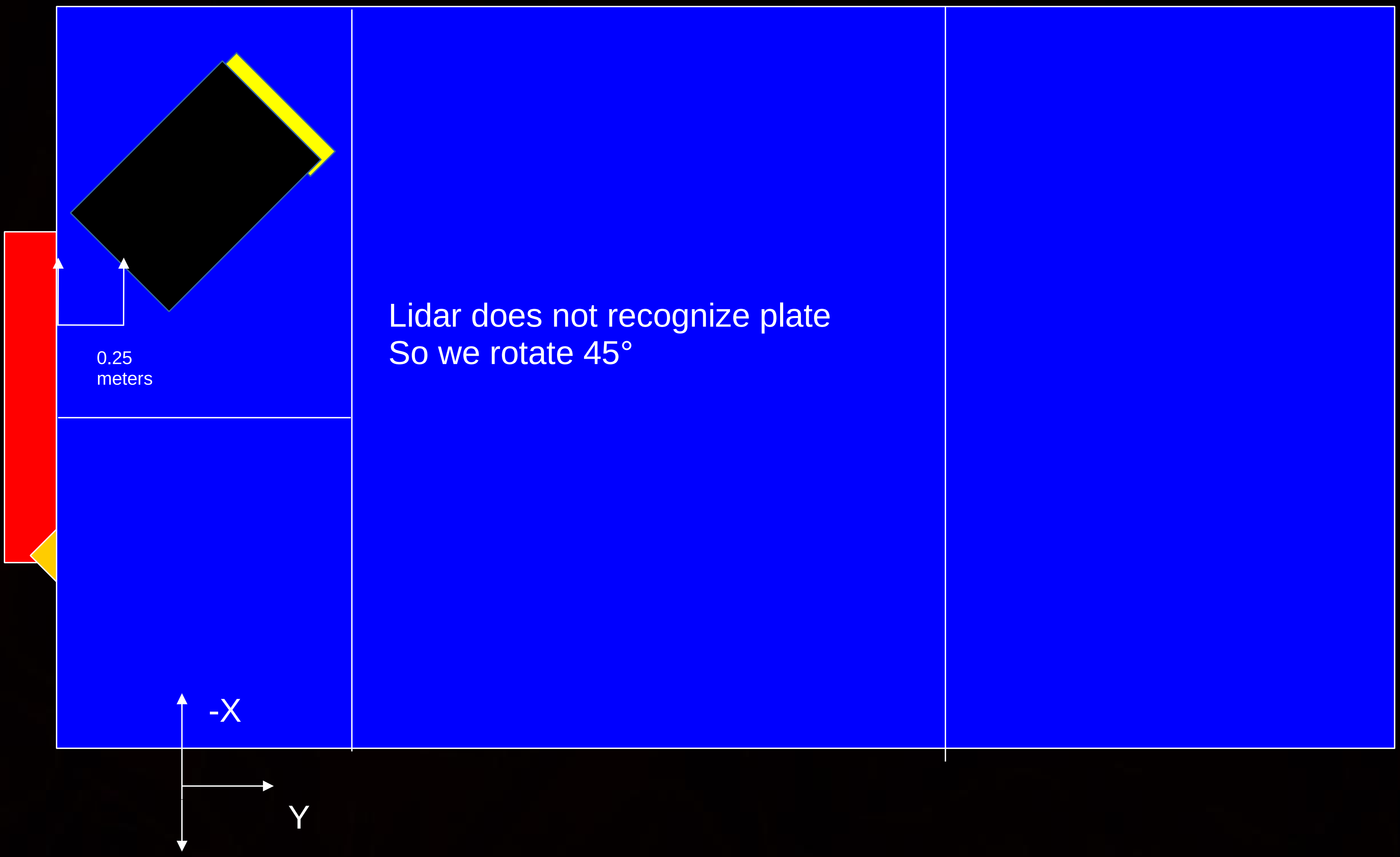
- We are centered and ready to move forward towards mining area

Simulation 2

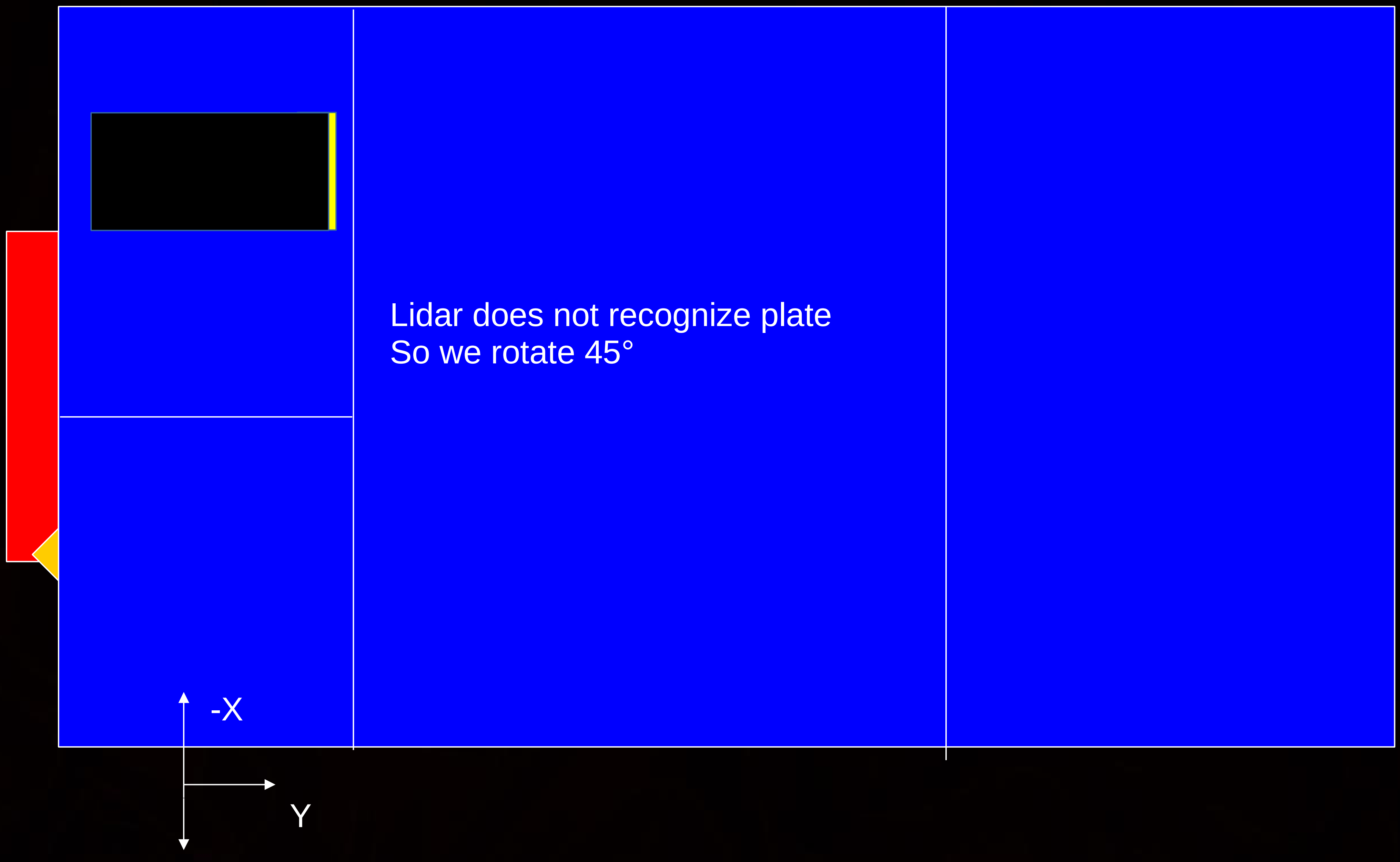
- The robot initially starts on the other half of the arena as the lidar.
- The robot is at a distance of 0.25 meters from the wall of the collector bin

- Initially, the lidar cannot see the plate on the robot
- We rotate the robot and then move towards the center

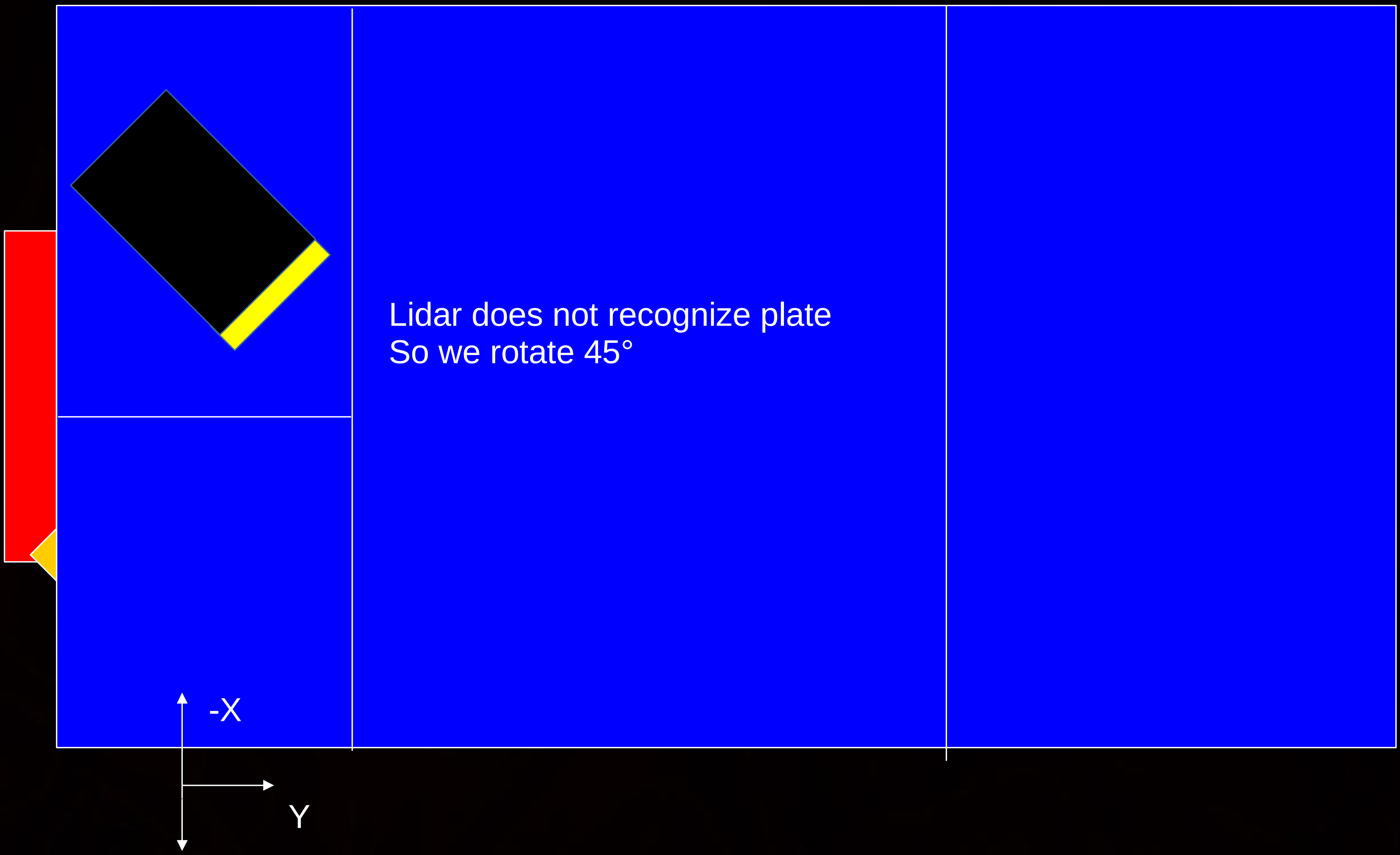
Simulation 2



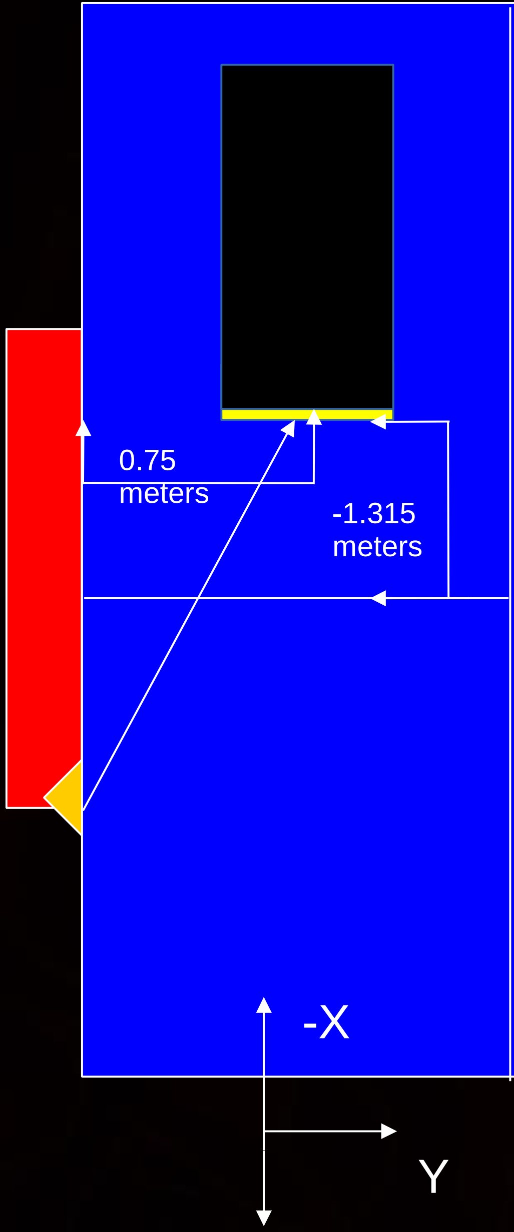
Simulation 2



Simulation 2



Simulation 2



- Lidar detects the plate
- Since the plate aligns with y-axis we calculate the centroid

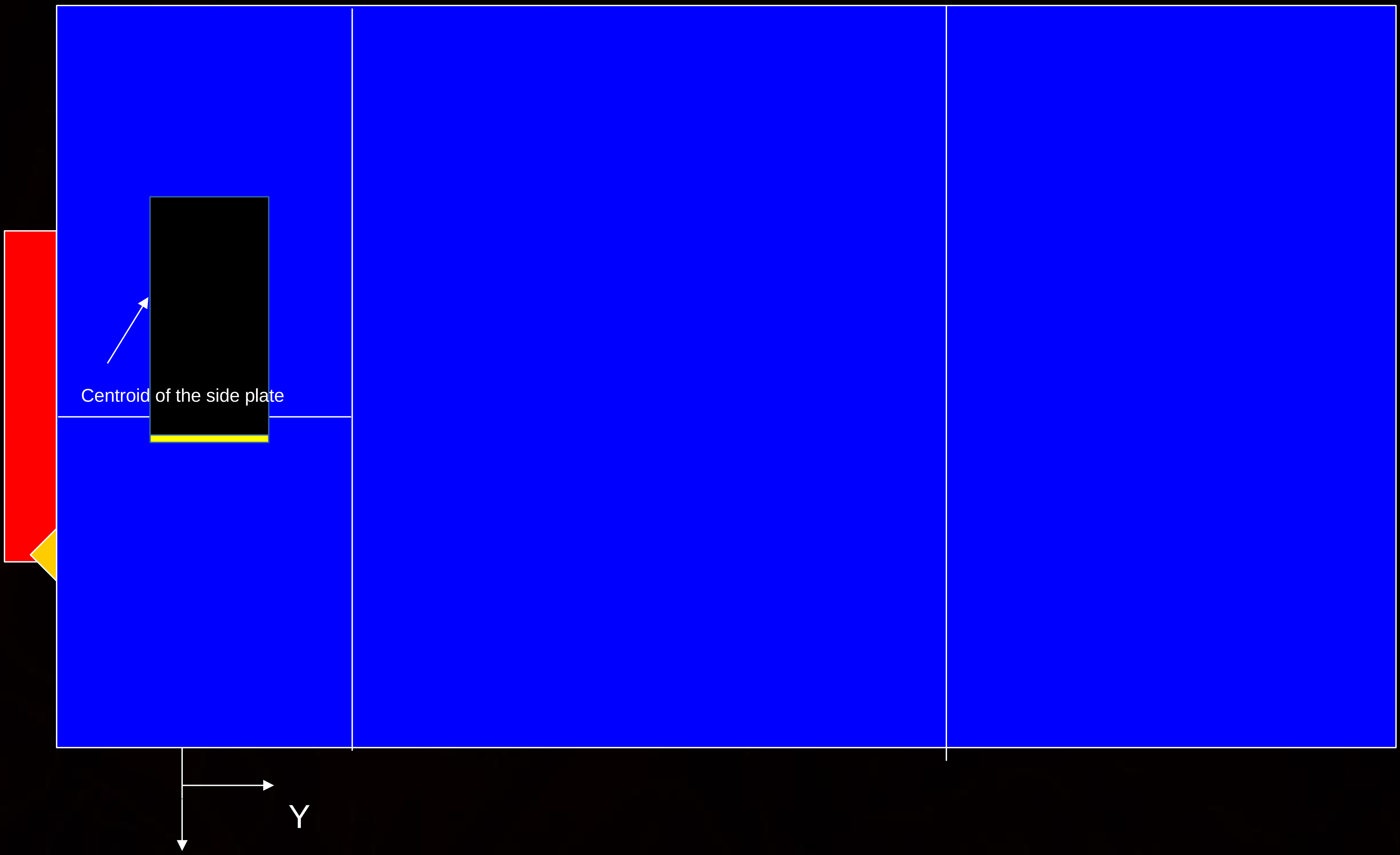
- Centroid:

X = -1.315 meters

Y = 0.75 meters

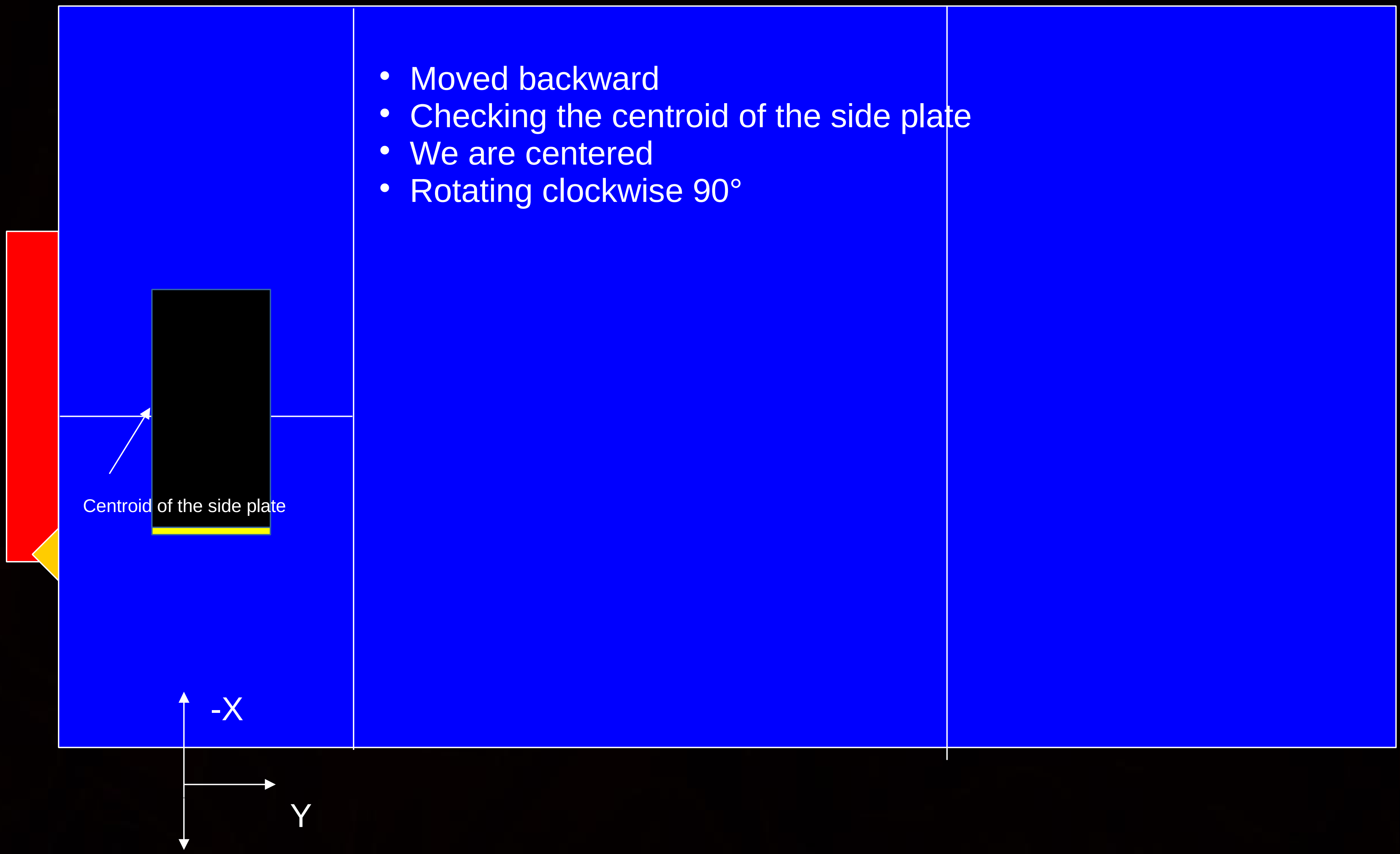
- Move backward approximately 1.1315 meters

Simulation 2

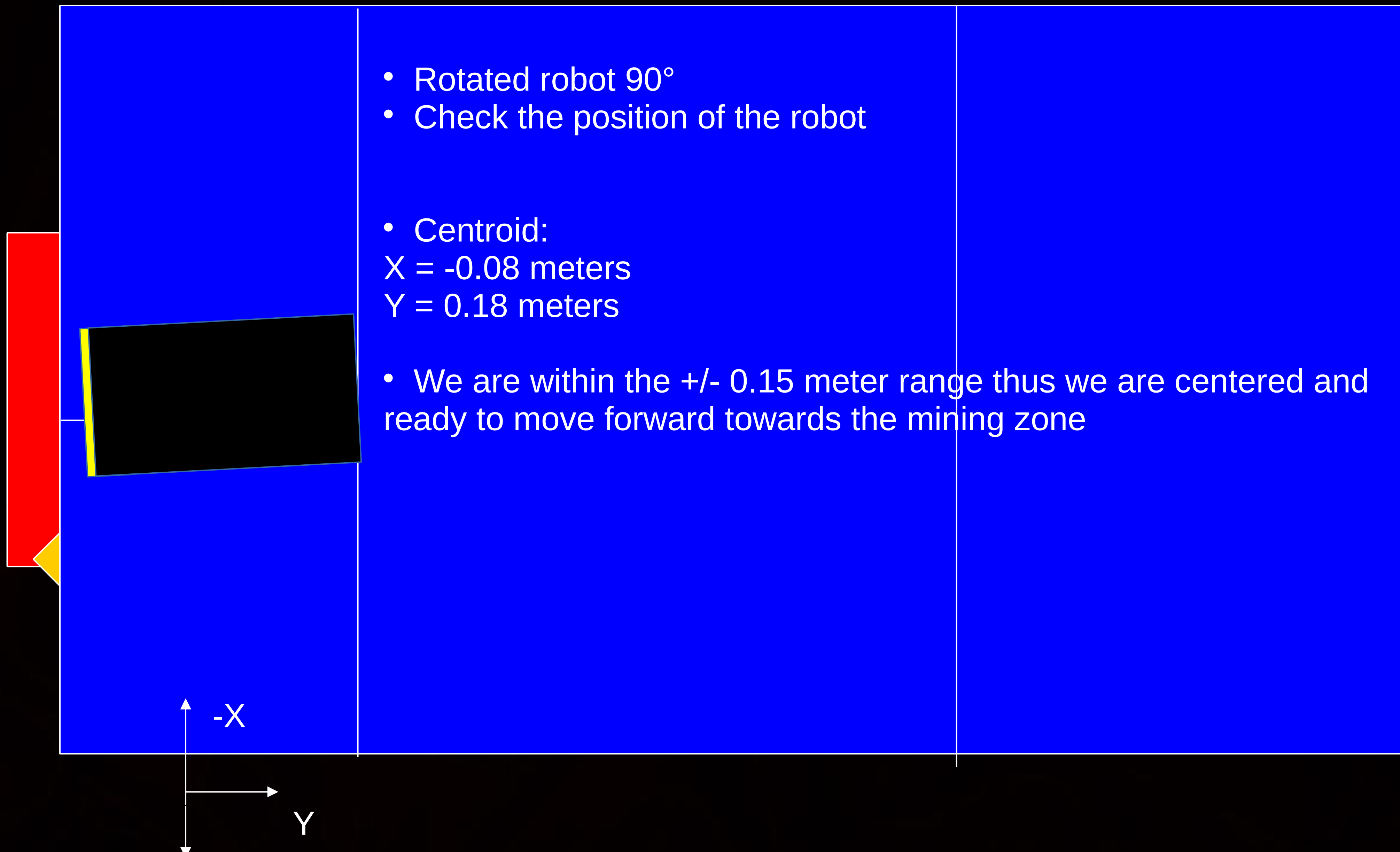


Simulation 2

- Moved backward
- Checking the centroid of the side plate
- We are centered
- Rotating clockwise 90°



Simulation 2



- Rotated robot 90°
- Check the position of the robot

- Centroid:

X = -0.08 meters
Y = 0.18 meters

- We are within the +/- 0.15 meter range thus we are centered and ready to move forward towards the mining zone

Time Analysis

Worst Case:

- The worst case scenario would be similar to the first simulation. The lidar does not detect the metal plate on the back of the robot initially so we have to rotate nearly 180° until we find the plate. Then rotate in order to move forward towards the center of the arena.
- Then we use the side plates to track our movement.
- In this worst case scenario depending on processing speed of the data points we will most likely take about a minute and a half to orient the robot.
- This is due to having to run through multiple incorrect lines during the first rotations of the robot, which will approximately be 1 minute to process
- After we have the metal plate scanned it then we only need to worry about finding the plate on the side of the robot, but knowing we have rotated correctly ensures that finding this plate will not pose a problem.
- Thus estimating the time it will take to worst case 1 minute and 30 seconds.

Time Analysis

Best Case:

- The best case scenario would be that the lidar immediately recognizes the plate, without any rotations needed to be done.
- Therefore we would just need to localize the side plates and then move forward or backward.
- In the best case scenario we find both plates immediately, without a necessity to rotate
- This scenario should approximately take 30 seconds, this is assuming that we are far from the center of the arena.
- NOTE: This excludes the fact that we could start already within the desired range to be considered centered.

Time Analysis

Average Case:

- In the average case we would take an approximate time of 45 seconds to 1:15 minutes.
- This will determine on the conditions of the arena and how accurately the lidar can detect the metal plate on the back of the robot.

Execution Plan

Week 1 & 2: Run a StatisticalOutlierRemoval and PassThrough filter.

Week 2 & 3: Attempt to detect an object and determine its length.

Week 4: With multiple objects in the cluster, extract the desired object.

Week 5 & 6: Having extracted the object from a cluster, determine the angle of the line or if it is parallel to a specified axis.

Week 7 & 8: Determine the centroid of an object and track the object as it is in motion.

Week 9 – 12: Testing