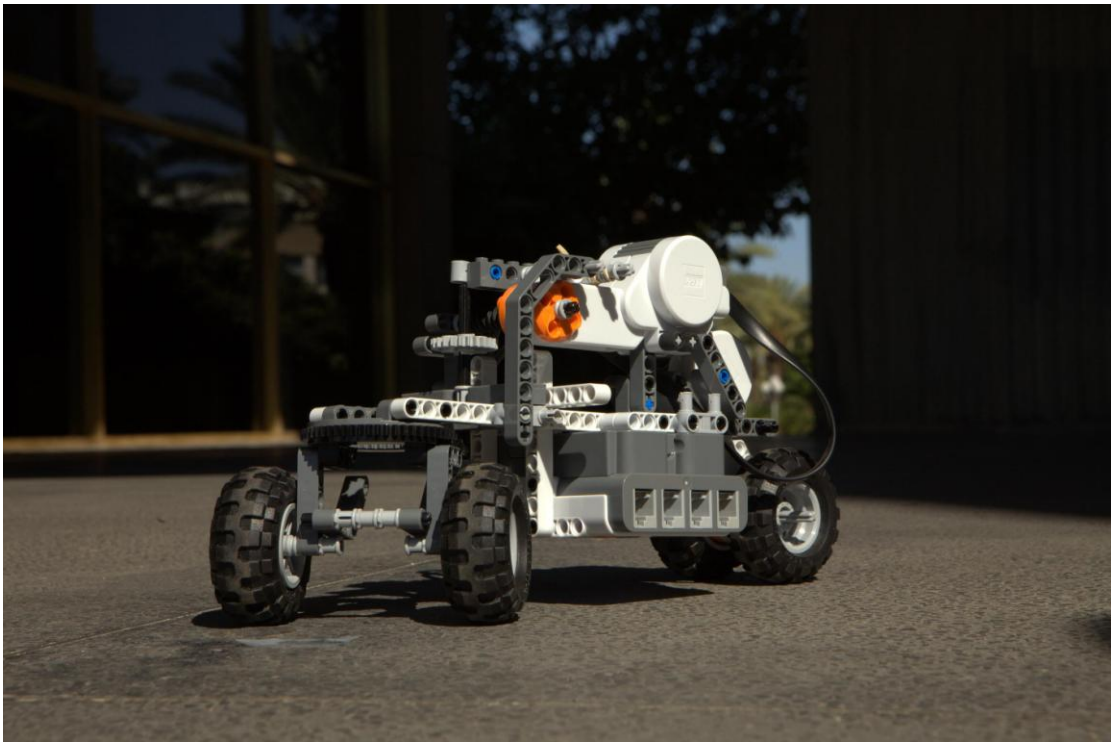
# ABS (Avinery Biton Skoran)

# Robot Milestones



# Contents



# Physical properties:

Length: 26.5 cm

Width: 12 cm

Wheels perimeter: 17.59 cm

Two motors:

- Motor A – controls the driving
- Motor B – controls the steering

# Constraints:

## Robot driving direction:

we had a problem of some degree of freedom in the wheel (even when the motor is not moving the wheels could slightly rotate around 3 degrees to each direction) and so, when driving forwards the robot was straying from the original direction, our solution was to change the direction of the robot so it will go backwards and so decreasing the amount of straying (like wagon that is better to pull then push since it will less influence the direction of movement).

## Lack of building parts:

When constructing the robot wanted to add some more accurate parts and use a different sort of steering, but since the basic construction kit that was provided was very limited we had to use these parts.

### Lack of symmetry:

The robot was constructed without full symmetry and thus when turning left and right the behavior is slightly different (more weight distribution on one side than the other).

### Using only one motor at a time:

The robot uses two motors and the commands are sent one by one (we wait until each command is completed before sending the next one). When we try to send more than one command at a time (not waiting until the previous command is completed) we got unpredictable results since the commands are cut off in the middle of the execution.

## Empiric measurements:

Rotation radius (when wheels are at 45 degrees): 40 cm

One rotation of Motor B makes the wheels rotate 6 degrees.

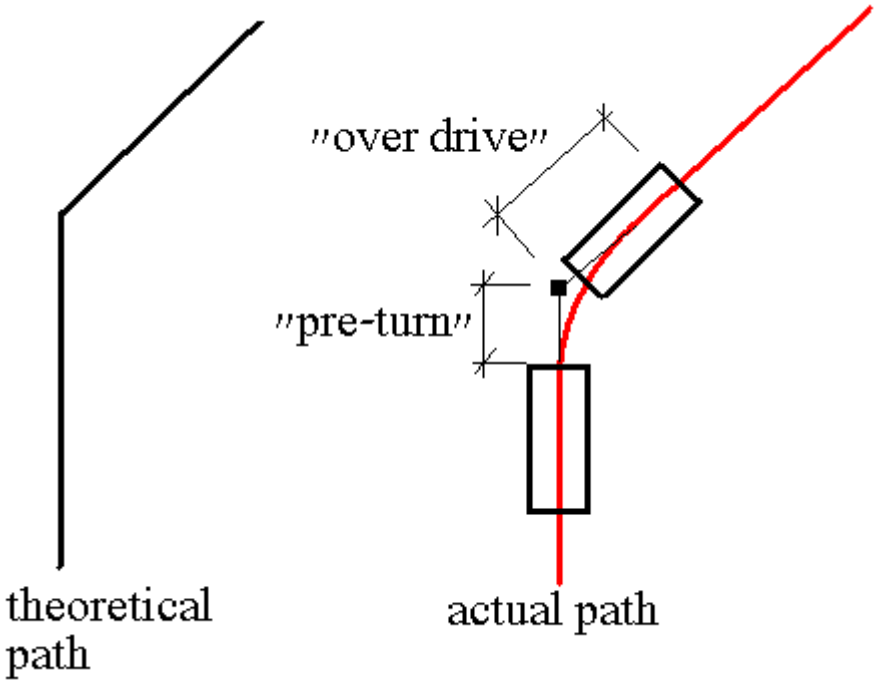
Rotation of the steering to 45 degrees takes 7.5 full motor rotations.

For the robot to make a turn:

- Rotate the wheel to 45 degrees (7.5 rotations)
- Rotate the driving motor for: `0.0248 * angles_to_rotate`
- Rotate the wheel back.

Note:

Since the robot is a car like robot it should start each turning earlier and it finishes it is after the designated point:




Hence the length of each strait line is: (length - previous_over_drive – pre_turn)

previous_over_drive and pre_turn were empirically measured to form the following equations:

$$\text{previousOverDrive} = (\text{previousAngle} / 90) * 16.2$$

$$\text{preTurn} = (\text{Angle} / 90) * 21.5$$

Since the new length can be negative the robot was programmed to "reverse" in case the length is negative.