

# SIGNALS, INSTRUMENTS AND SYSTEMS

PROJECT 1

LINE-FOLLOWING USING CAMERA AND  
OBSTACLE AVOIDANCE ON E-PUCK  
ROBOT

GROUP 3

MAXWELL BERGSTRÖM

VICTORIA KUNZ



# OUTLINE



Main and self-  
assigned goals



Flow chart



Sensors and  
parameters



Results



Conclusion

# MAIN GOALS

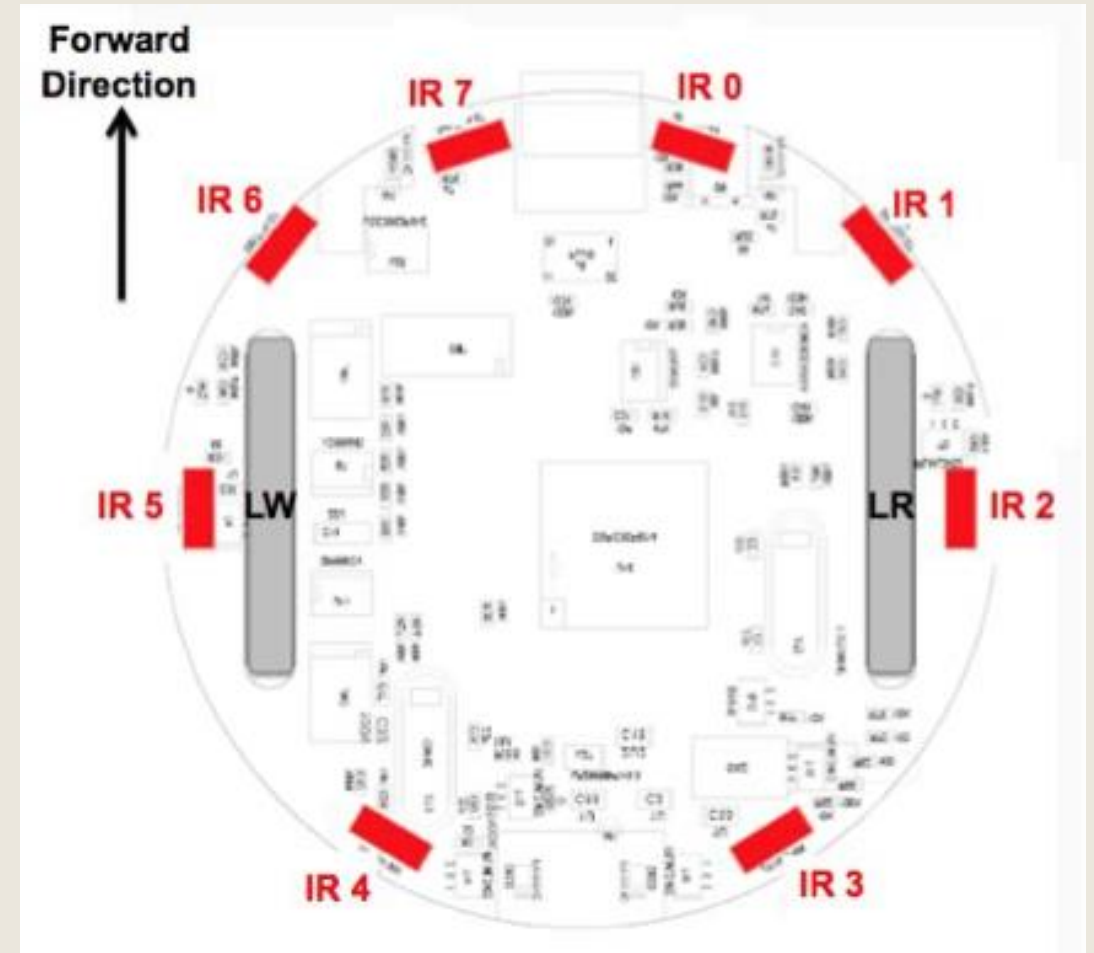
- Recognize and follow line
- Avoid obstacles and regain line
- Track the robot's position with odometry

# SELF-ASSIGNED GOALS

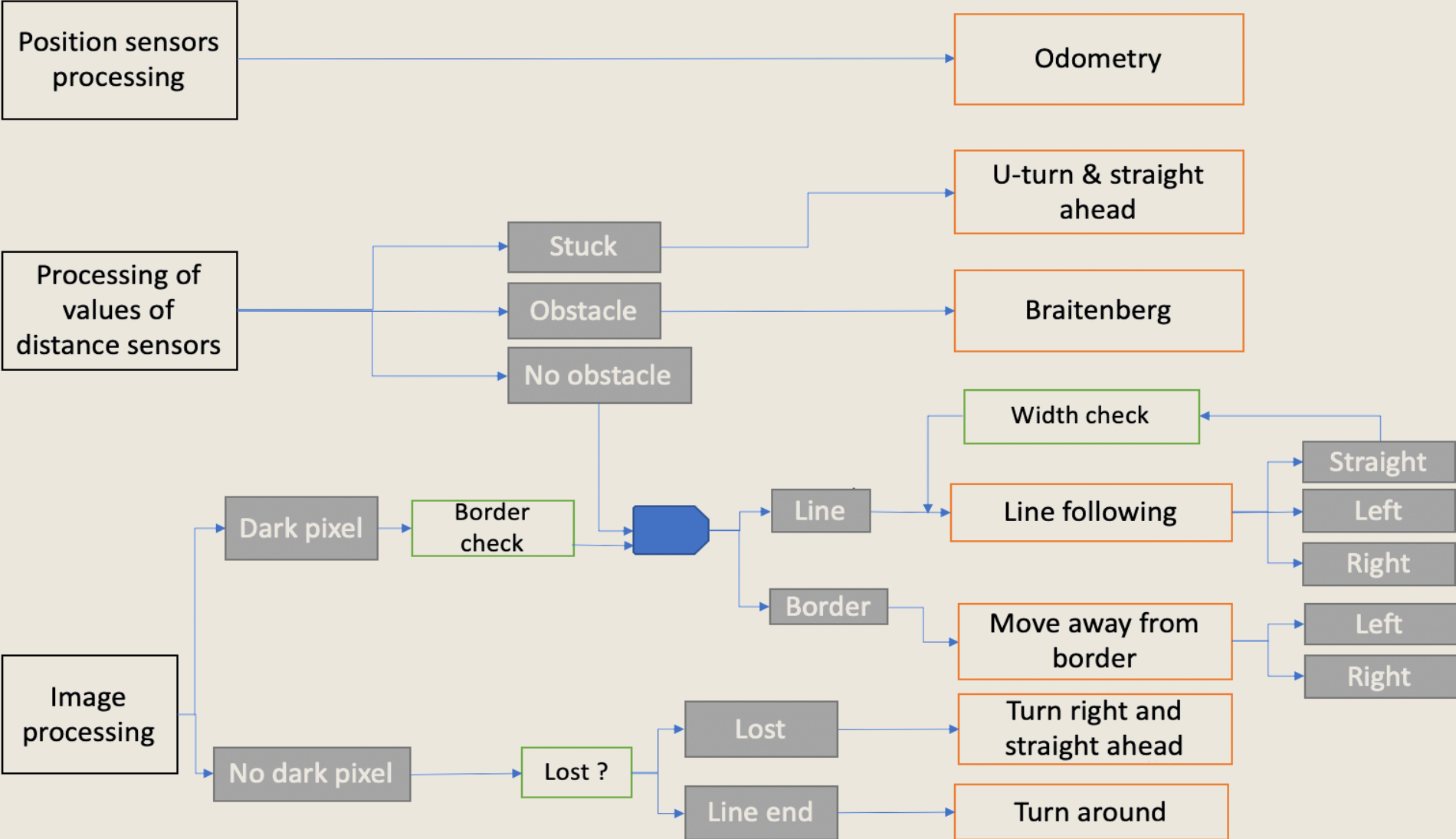
- Follow all shapes of lines given to us, detect end of lines
- No filter on border colour, larger spectrum of colours detected
- Adapt behaviour to widths of lines

# SENSORS AND PARAMETERS

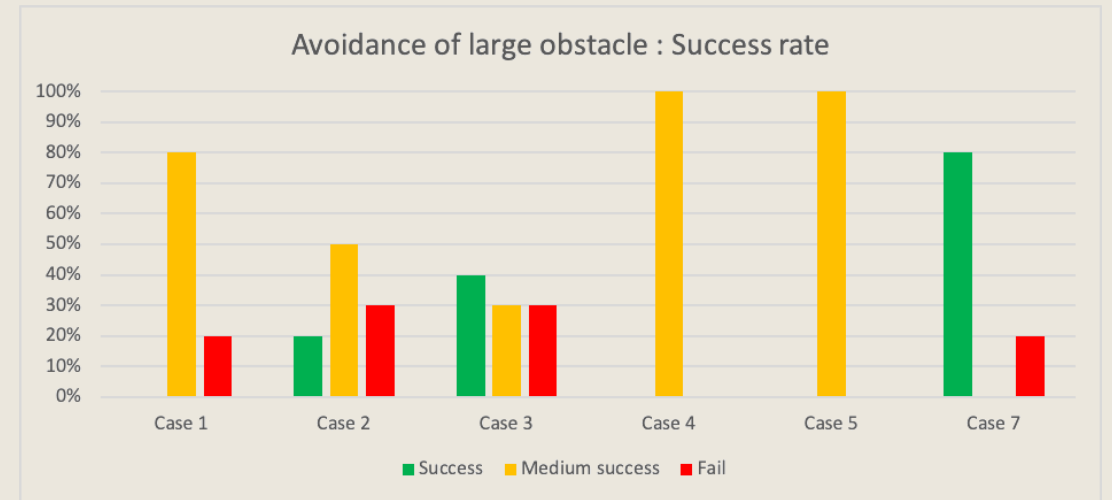
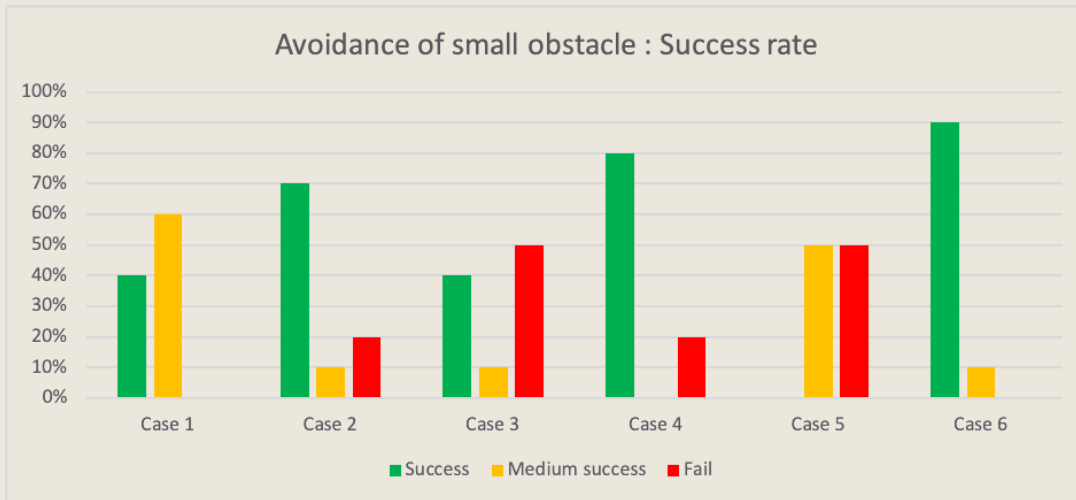
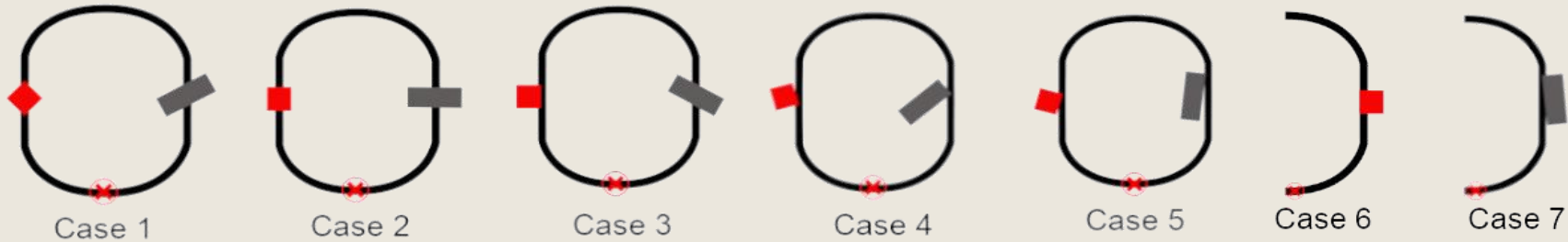
- Camera settings : width 40, height 10, angle - 0.3 rad, field of view 0.8
- Distance sensors for obstacle avoidance
- Position sensors on wheels for odometry



# FLOW CHART



# OBSTACLE AVOIDANCE: EXPERIMENT RESULTS



# OBSTACLE AVOIDANCE: SOLUTIONS

Absolute priority on obstacle avoidance



Reevaluate behaviour according to a distance sensors and camera readings after a certain amount of time

Different behaviour in a clock wise and an anti-clockwise directions



Work on symmetry

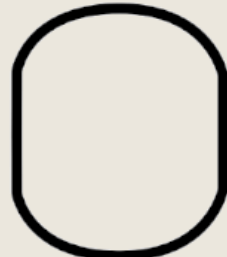
# SELF-ASSIGNED GOALS

- Follow all shapes of line given to us, and detect different colours

Lines used:



Square Line



Oval Line



Funky Line



Semi-Oval Line

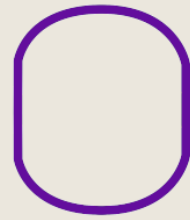
Colors used :



Oval gray



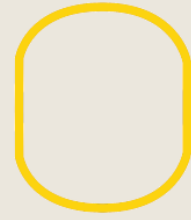
Oval green



Oval purple



Oval red



Oval yellow



Square multicolor

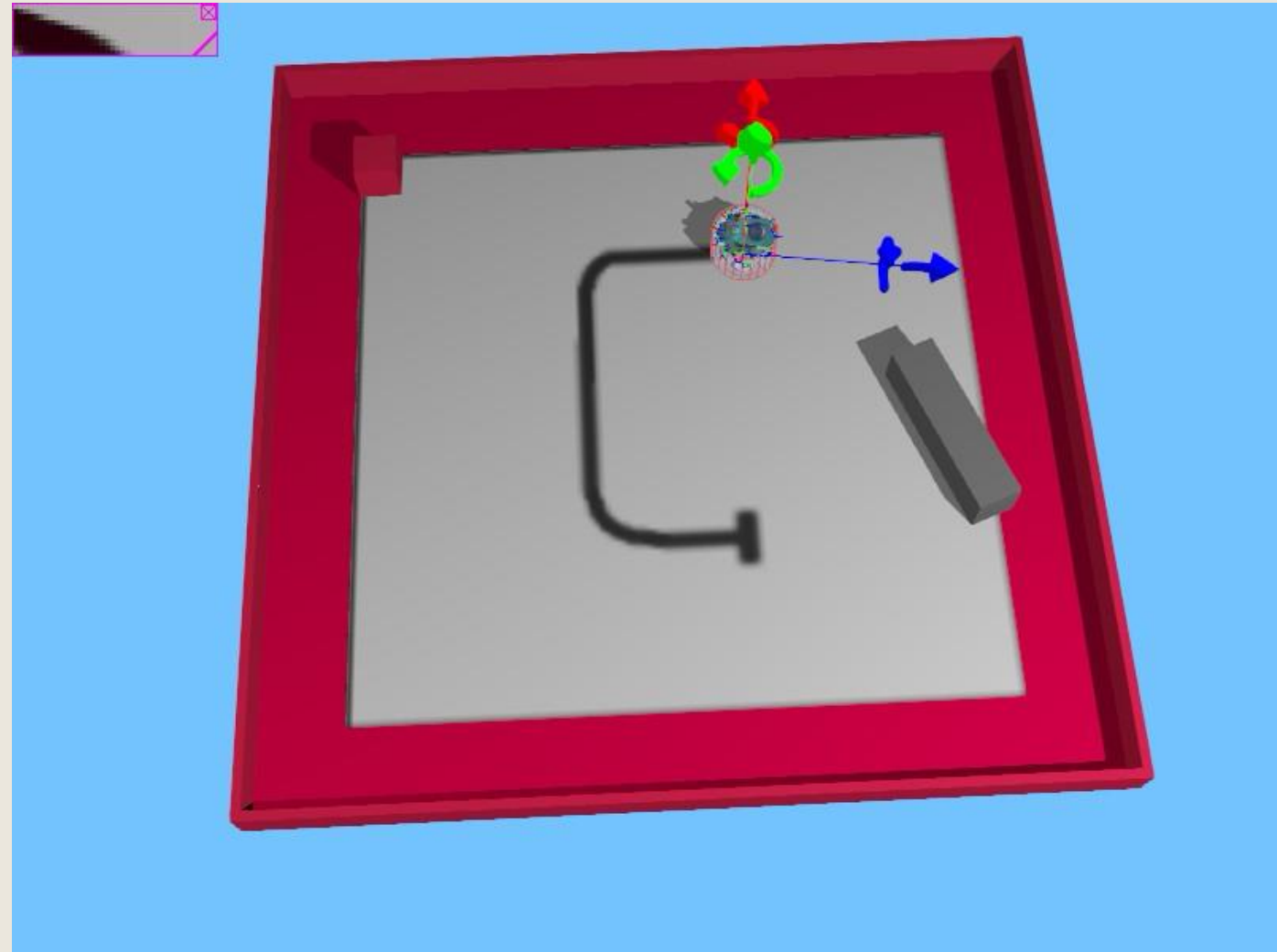


Funky blue

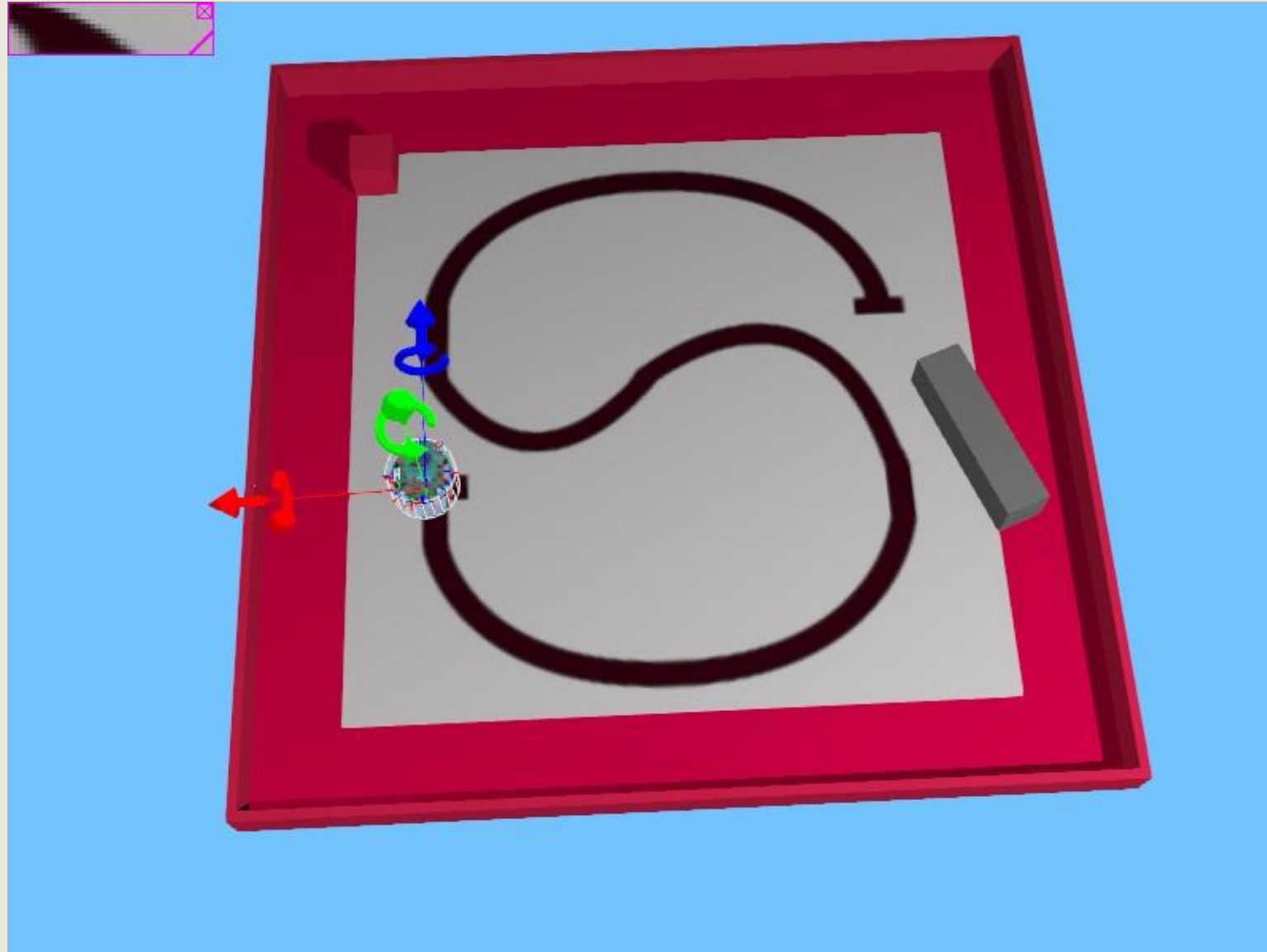


# WIDTH OF LINE

- Robot adapts to the width of the line  
->*width\_check*
- Robot detects the end of the line and turns around



# BORDER CHECK : simulation



# BORDER CHECK

- No filter on border colour to detect more colours

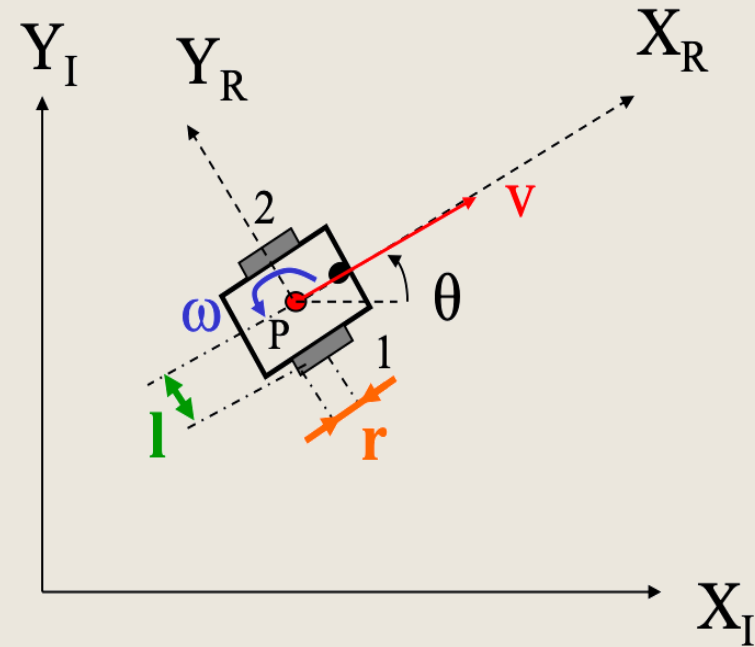
-> *border\_check*

- The error is due to the encircled case which is not considered by the robot



# ODOMETRY

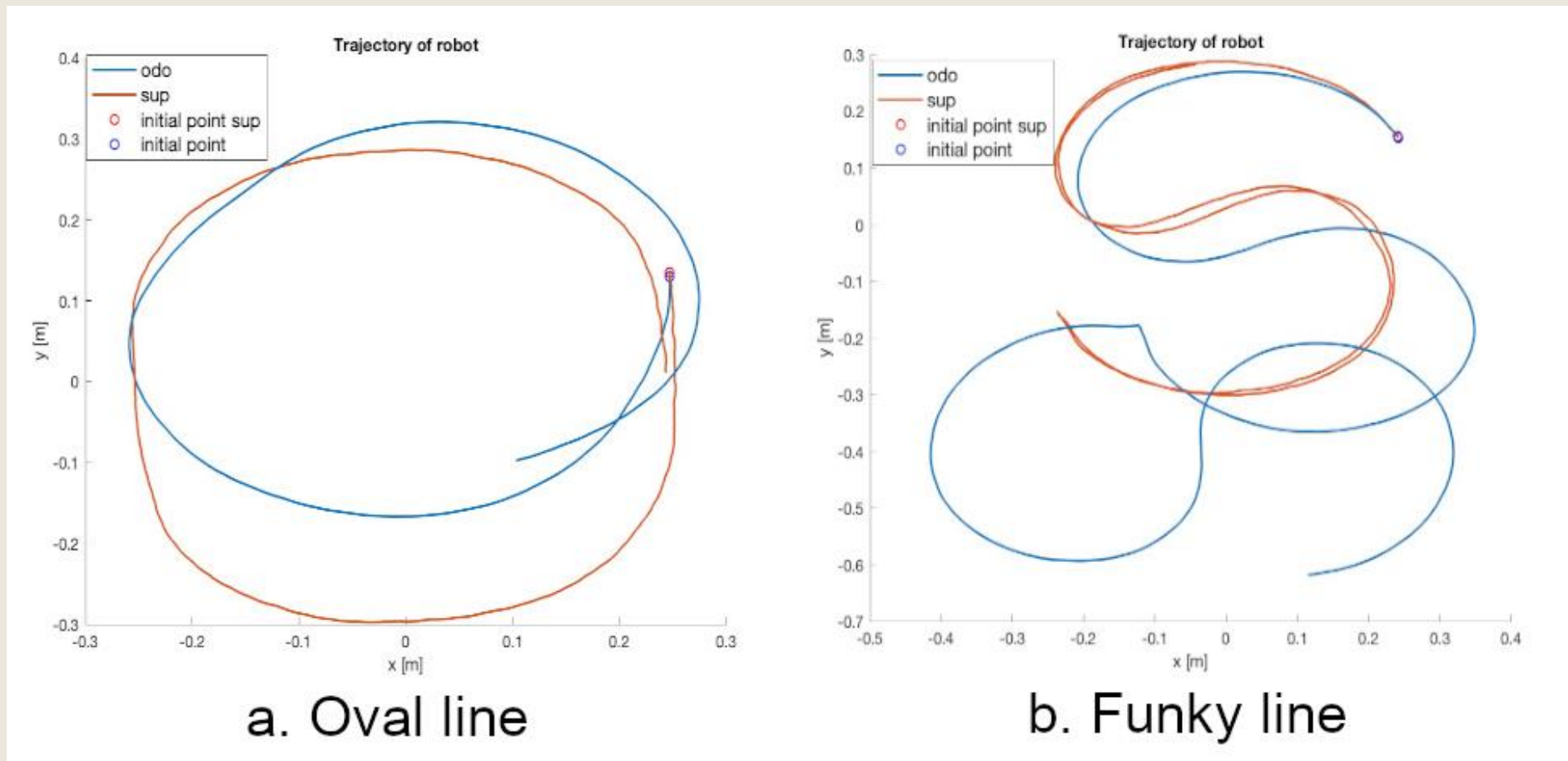
1.  $\dot{x}_R = v = \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2}$
2.  $\dot{y}_R = 0$
3.  $\dot{\theta}_R = \omega = \frac{r\dot{\phi}_1}{2l} + \frac{-r\dot{\phi}_2}{2l}$
4.  $\dot{\xi}_I = R^{-1}(\theta)\dot{\xi}_R$



$$\dot{\xi}_I = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} + \frac{-r\dot{\phi}_2}{2l} \end{bmatrix}$$

# ODOMETRY

- Calibration on axle length and positions calculated
- Were position sensors the best choice ?



# CONCLUSION



**Weaknesses were discovered during simulations**



**Room for improvement:**

- Border check problematic but part of the game
- Work on obstacle avoidance and line regaining
- Work on harmonisation of behaviours