Maastricht University

Department of Data Science and Knowledge Engineering

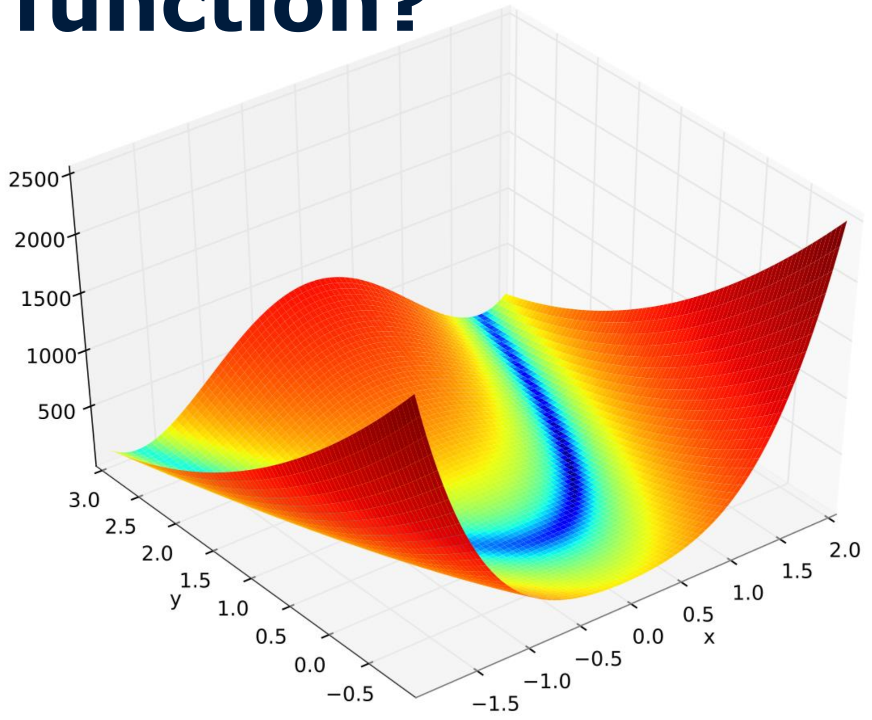
Autonomous Robotic Systems

Master Course

Assignment Mobile Robot Simulator

DISCUSSION ASSIGNMENT PSO

What happens on Rosenbrock function?

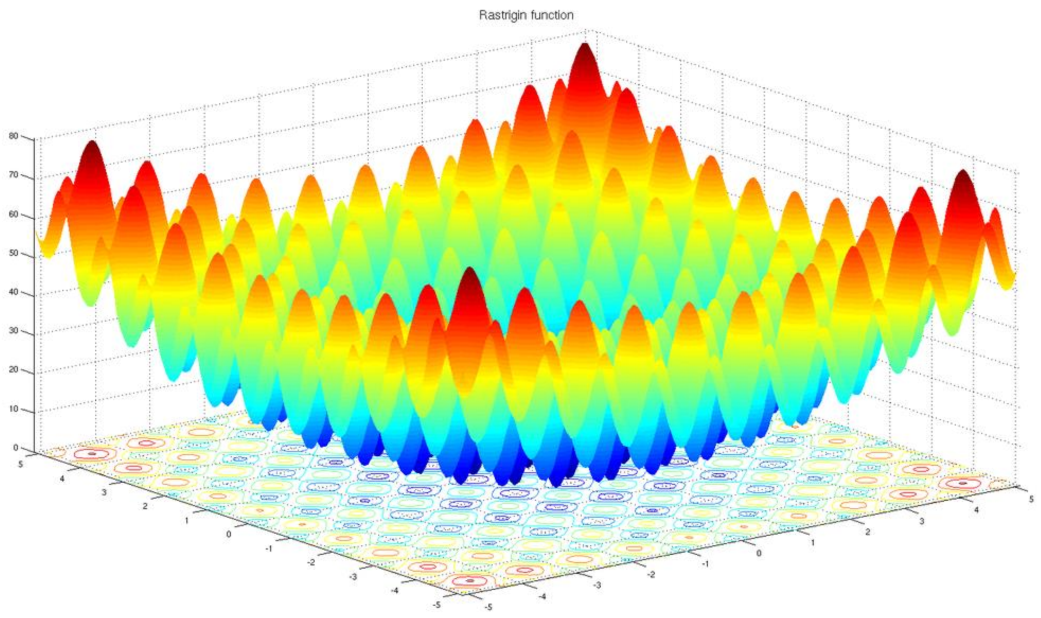


Rosenbrock function with two variables

$$f(x,y) = (a-x)^2 + b(y-x^2)^2$$

- Deep valley with known minimum minimum at (a,a^2)
- Easy to find valley
- Difficult to find global minimum
- **Is PSO guaranteed to find optimum?**
- **What would Gradient Descent do?**

What happens on Rastrigin function?



Rastrigin function with two variables

$$f_2(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10cos(2\pi x_i)\right)$$

- Many local minima
- Known global minimum at **x=0**
- **Is PSO guaranteed to find optimum?**
- **What would Gradient Descent do?**

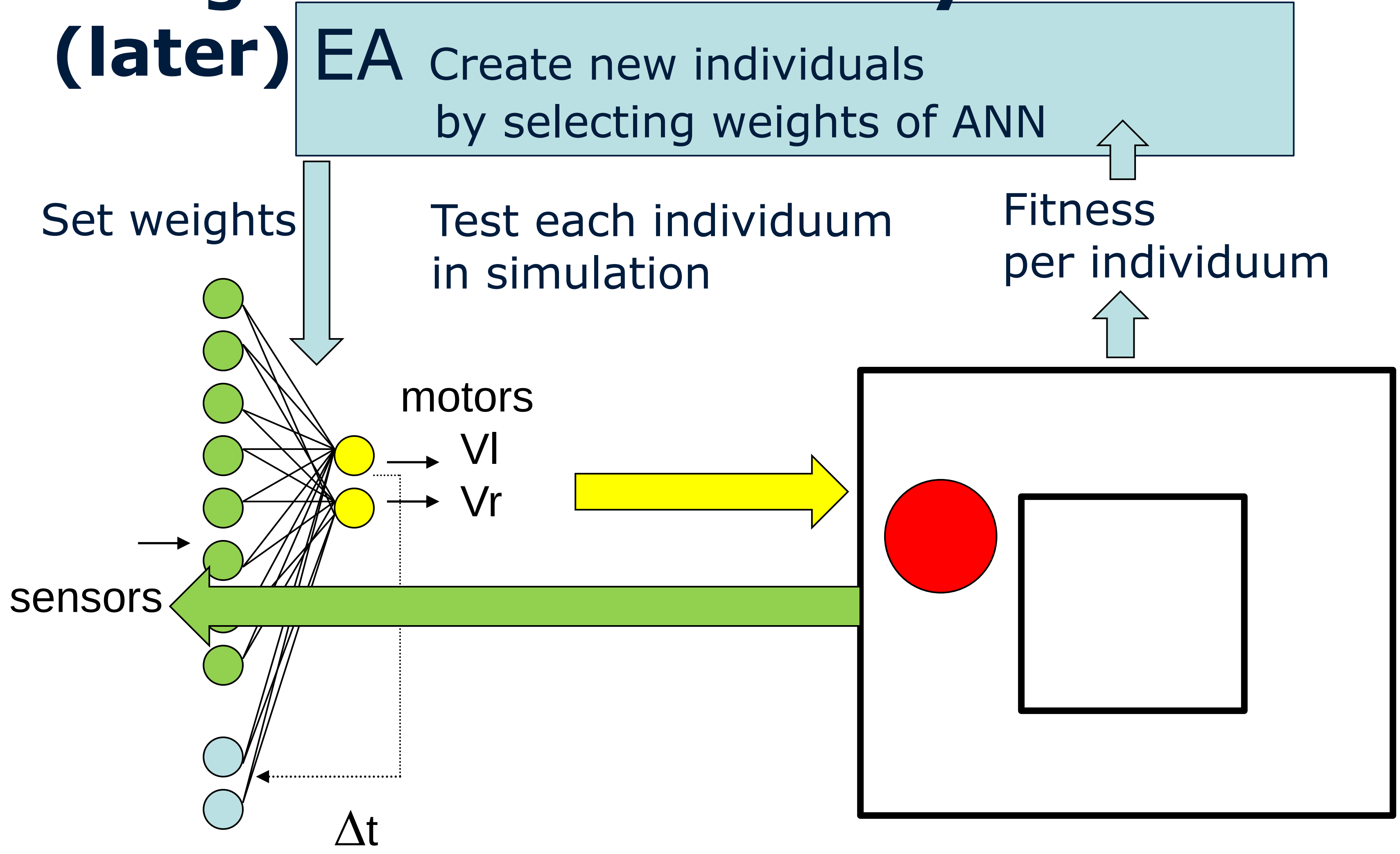
Get some deep understanding

- Which algorithm has better performance (PSO or gradient descent)? – why?
- Under which circumstances does is make sense to combine PSO and gradient descent? How would it be best to combine these?
- What would happen for benchmark function that does not contain smooth gradients?

MOBILE ROBOT SIMULATOR

Collision-free mobile robot movement

Assignment Evolutionary Robotics (later)



Required ingredients for Assignment Evolutionary Robotics

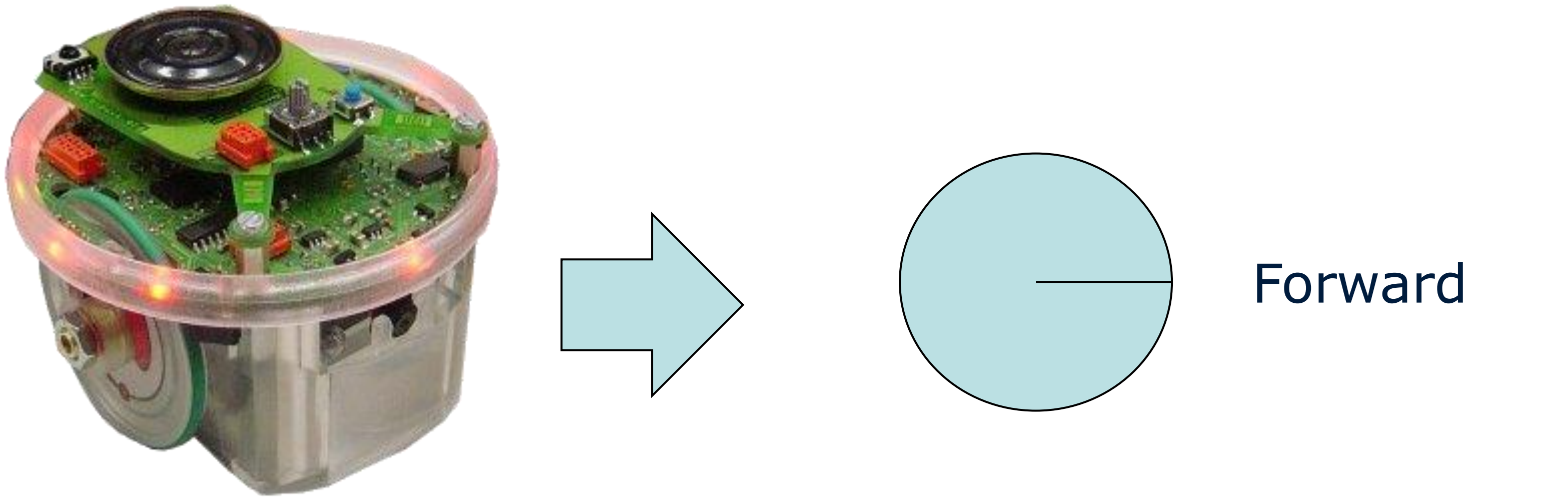
- ❑ Evolutionary Algorithm (EA)
- ❑ Testbed for EA
- ❑ Controller
- ❑ Genome
- ❑ Fitness evaluation
- ❑ **Motion model**
- ❑ **Sensor model**
- ❑ **Collision handling**

$\left.\begin{array}{l}\\ \\ \\ \end{array}\right\}$ **TODAY: Assignment Mobile Robot Simulator**

ASSIGNMENT MOBILE ROBOT SIMULATOR

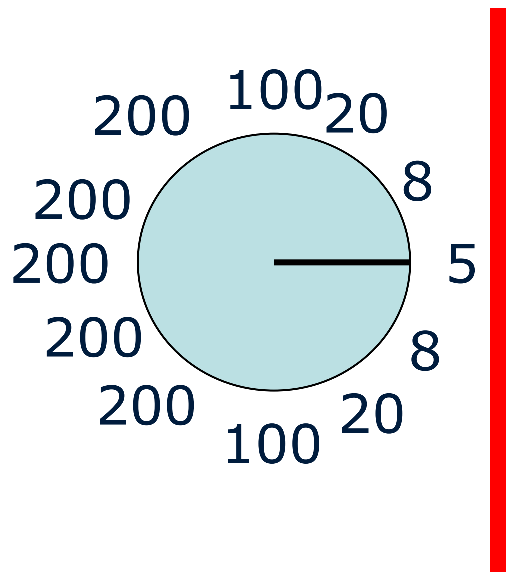
2D robot model

- Wheeled robot can be displayed a circle with line indicating forward direction



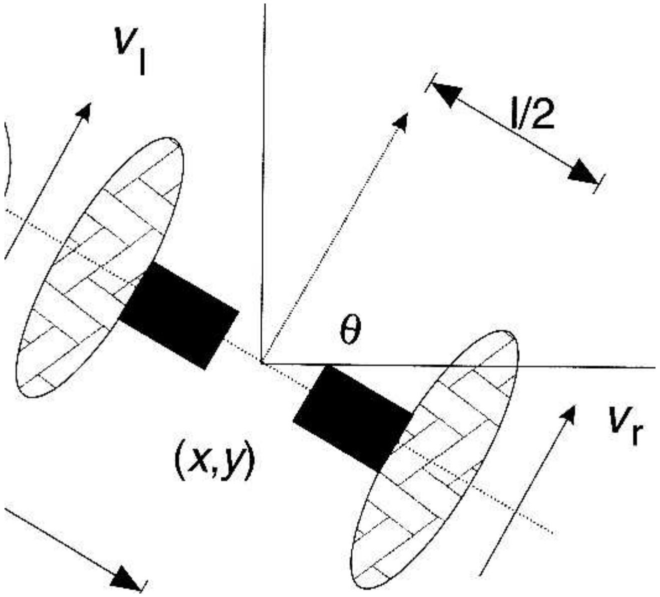
Goal of Assignment Mobile Robot Simulator

- Allow me to move the robot in a closed room using the buttons of your PC's keyboard
- If robot collides with wall: move robot realistically along side of walls (without entering the wall)
- Display values of distance sensors as plain numbers



Goal of Assignment Mobile Robot Simulator

- Control speed of motors of left and right wheel with keyboard
- W: positive increment of left wheel motor speed
- S: negative increment of left wheel motor speed
- O: positive increment of right wheel motor speed
- L: negative increment of right wheel motor speed
- X: both motor speeds are zero

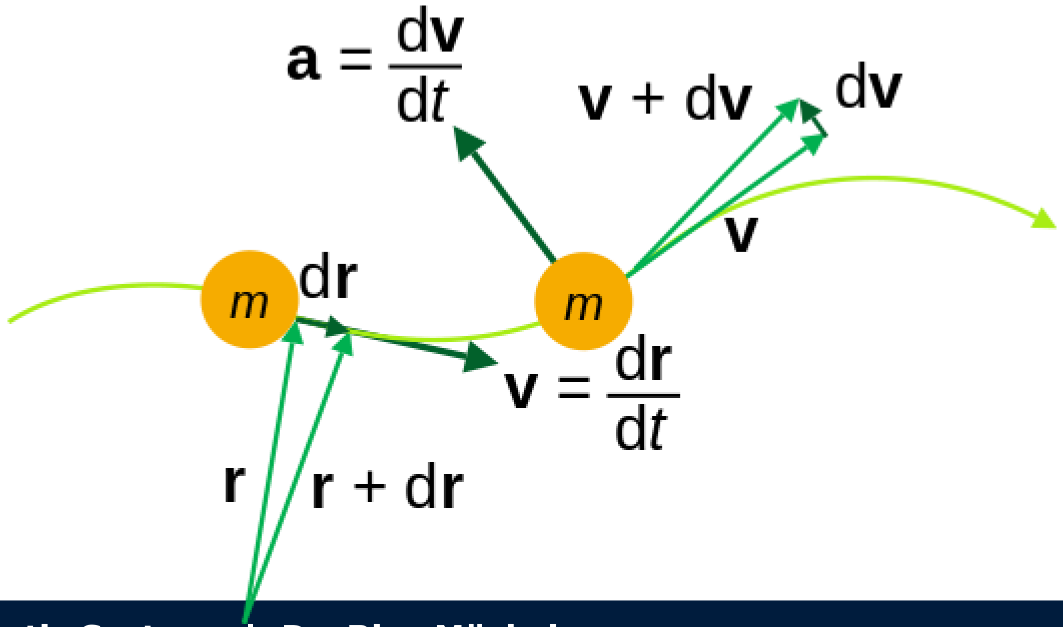


- T: positive increment of both wheels' motor speed
- G: negative increment of both wheels' motor speed

MOTION MODEL

Kinematics

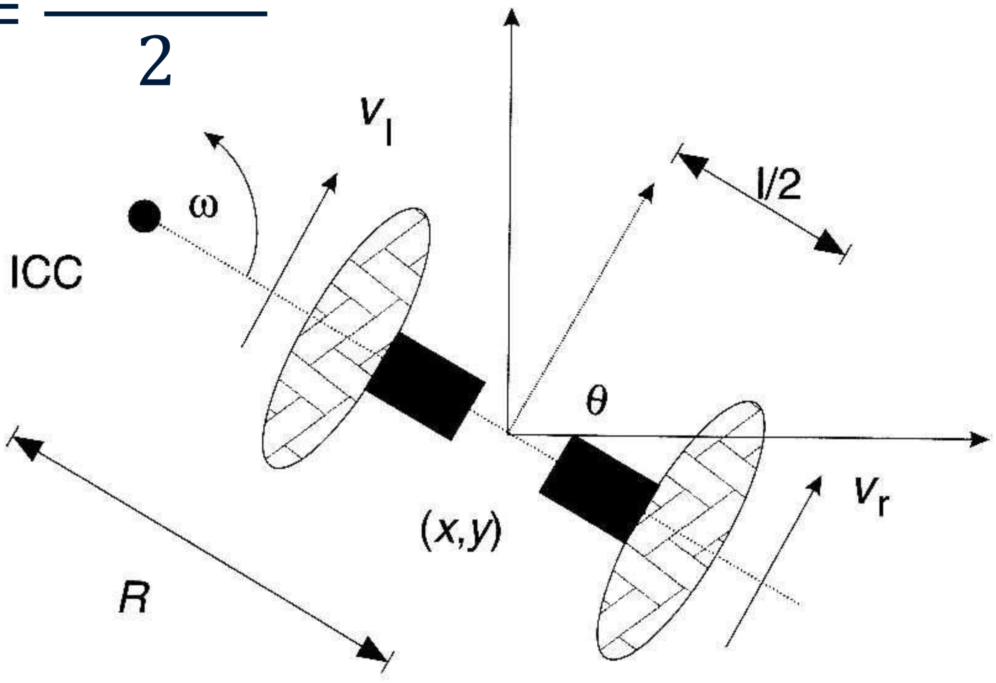
- Kinematics is a branch of classical mechanics that describes the motion of points, bodies (objects), and systems of bodies (groups of objects) without considering the mass of each or the forces that caused the motion.[Wikipedia, 2018]


$\mathbf{a} = \frac{d\mathbf{v}}{dt}$
$\mathbf{v} + d\mathbf{v}$
$d\mathbf{v}$
$\mathbf{v}$
m
$d\mathbf{r}$
m
$\mathbf{v} = \frac{d\mathbf{r}}{dt}$
$\mathbf{r}$
$\mathbf{r} + d\mathbf{r}$

Wheeled robot with differential drive

- How to calculate translation of robot?
- How to calculate rotation of robot?
- ICC - Instantaneous Center of Curvature: point that the robot rotates about

$$v = \frac{v_r + v_l}{2}$$



Differential Drive kinematics (from Dudek and Jenkin, *Computational Principles of Mobile Robotics*.

How to calculate rotation?

By varying the velocities of the two wheels, we can vary the trajectories that the robot takes. Because the rate of rotation ω about the ICC must be the same for both wheels, we can write the following equations:

$$\omega\,(R + l/2) = V_r \qquad (1)$$

$$\omega\,(R - l/2) = V_l \qquad (2)$$

where l is the distance between the centers of the two wheels, V_r , V_l are the right and left wheel velocities along the ground , and R is the signed distance from the ICC to the midpoint between the wheels. At any instance in time we can solve for R and ω:

$$R = \frac{l}{2}\frac{V_l + V_r}{V_r - V_l}\,;\;\; \omega = \frac{V_r - V_l}{l}; \qquad (3)$$

[Differential Drive Robots: notes compiled from Dudek and Jenkin, *Computational Principles of Mobile Robotics*.]

Check if model makes sense

There are three interesting cases with these kinds of drives.

1. If $V_l = V_r$, then we have forward linear motion in a straight line. R becomes infinite, and there is effectively no rotation - ω is zero.
2. If $V_l = -V_r$, then $R = 0$, and we have rotation about the midpoint of the wheel axis - we rotate in place.
3. If $V_l = 0$, then we have rotation about the left wheel. In this case $R = \frac{l}{2}$. Same is true if $V_r = 0$.

Note that a differential drive robot cannot move in the direction along the axis - this is a singularity. Differential drive vehicles are very sensitive to slight changes in velocity in each of the wheels. Small errors in the relative velocities between the wheels can affect the robot trajectory. They are also very sensitive to small variations in the ground plane, and may need extra wheels (castor wheels) for support.

Forward kinematics

In figure 1, assume the robot is at some positon (x, y), headed in a direction making an angle θ with the X axis. We assume the robot is centered at a point midway along the wheel axle. By manipulating the control parameters V_l , V_r, we can get the robot to move to different positions and orientations. (note: V_l , V_r) are wheel velocities along the ground).

Knowing velocities V_l, V_r and using equation 3, we can find the ICC location:

$$ICC = [x - R\,sin(\theta)\,,\, y + R\,cos(\theta)] \tag{4}$$

and at time $t + \delta t$ the robot's pose will be:

Rotation matrix

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} cos(\omega\delta t) & -sin(\omega\delta t) \\ sin(\omega\delta t) & cos(\omega\delta t) \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & \quad 0 \end{matrix} & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix} \tag{5}$$

This equation simply describes the motion of a robot rotating a distance R about its ICC with an angular velocity of ω.

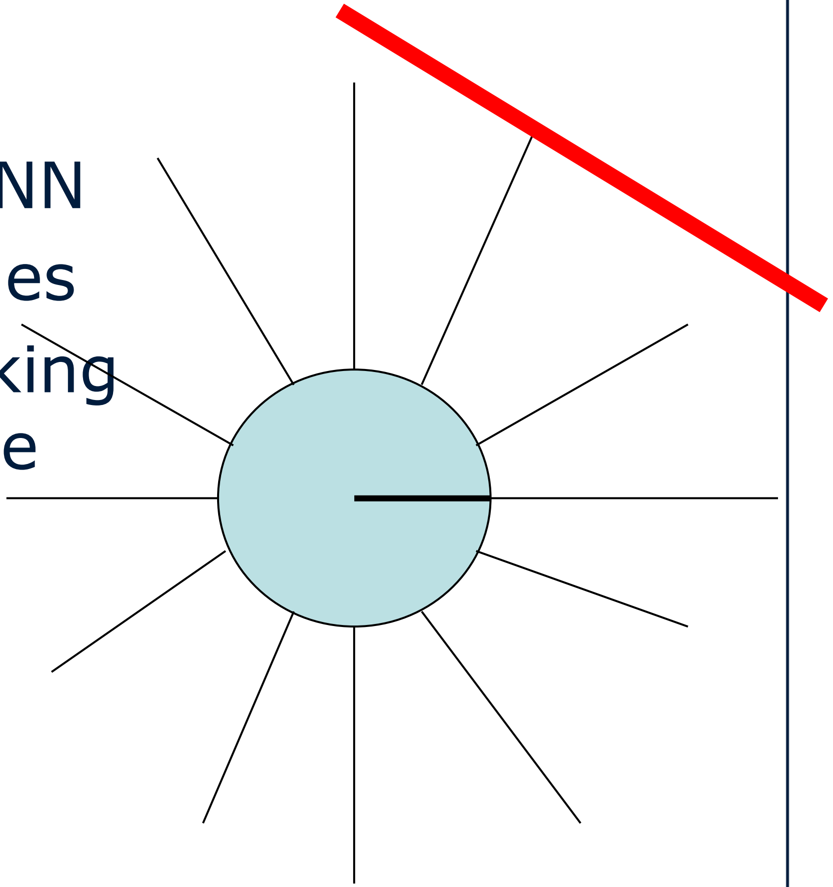
Refer to figure 2. Another way to understand this is that the motion of the robot is equivalent to 1) translating the ICC to the origin of the coordinate system, 2) rotating about the origin by an angular amount $\omega\,\delta t$, and 3) translating back to the ICC.

SENSOR MODEL

Sensor model

- 12 infrared distance sensors (30° distance) as input to ANN
- Model sensors as straight lines
- Intersections with lines marking walls can be used as distance measurement
- Limit distance measure
- Implement 12 sensors:

- Use software library for calculations



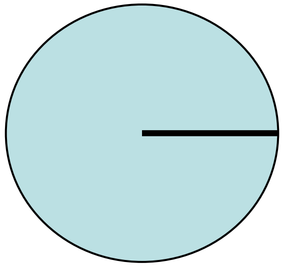
How to implement?

1) Convert environment into local coordinate system of the robot
2) Calculate virtual sensor lines starting from robot
3) Find intersection with lines representing walls
4) Calculate distances (center of robot to intersection points)
5) Apply distance threshold
6) Apply transformation for all distances below distance threshold

COLLISION DETECTION

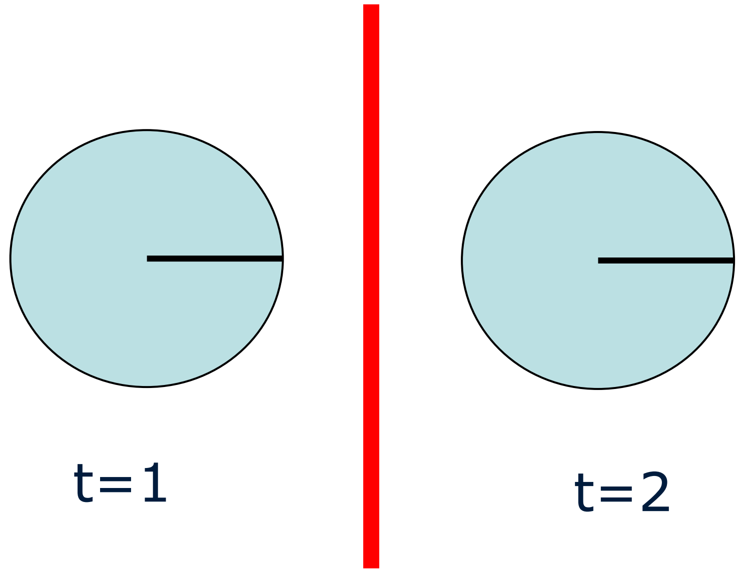
Collision detection and handling

- Use bounding object (circle) for robot
- Calculate intersection of body with all walls
- Make sure that robot speed and simulation time step allow for detection of collision
- For precise simulations calculate back point in time of collision
- For collision handling, apply motion only parallel to wall, rest motion perpendicular to wall (no friction model required)



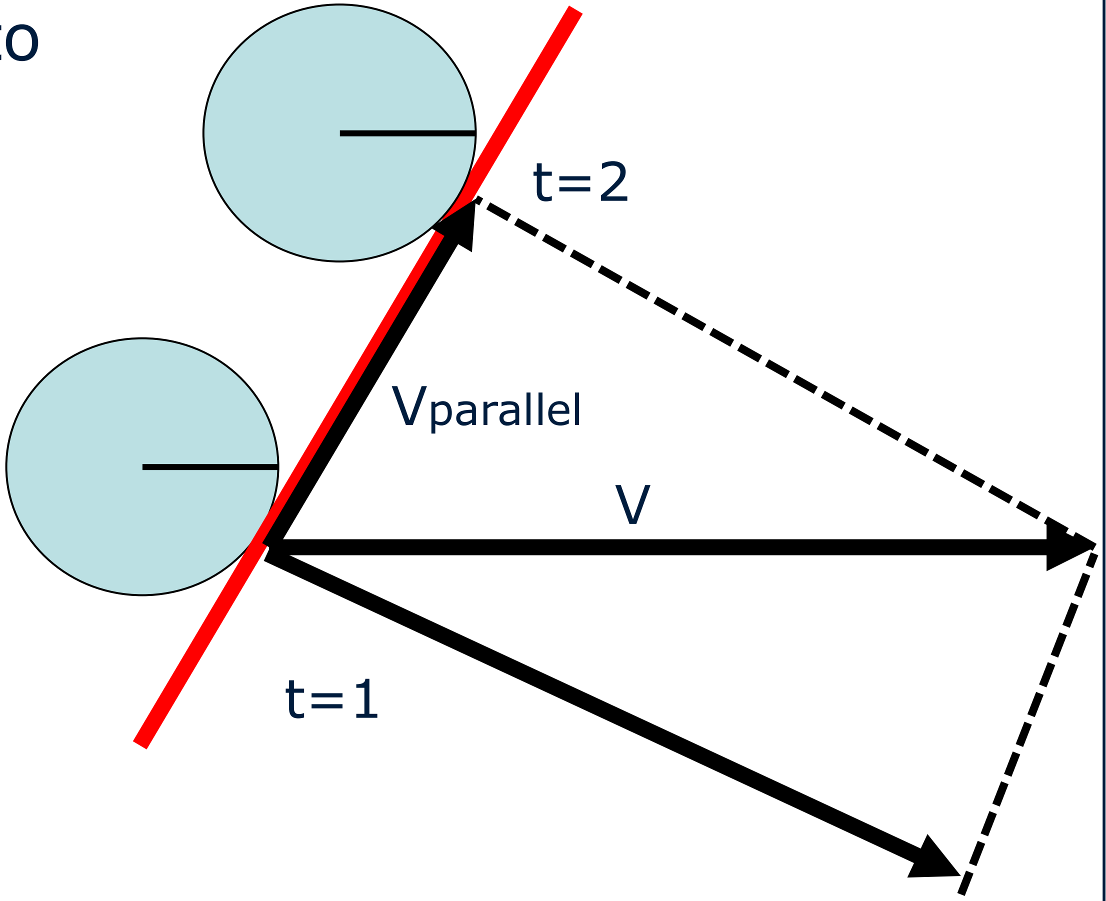
Collision handling – time step

- Fixed time step might allow robots to move through wall at high speeds
- → you must do some backwards check
- → if a collision occurred you might have to calculate intermediate time steps



Collision handling – moving along the wall

- If driver moves into the wall, properly calculate velocity components perpendicular and parallel to wall
- Only velocity parallel to wall contributes to actual motion



Distribution of work

- Make sure that all of you understand all components of the assignment (basis for exam)
- Main tasks
  - motion model (incl. collision handling)
  - sensor model
  - visualization
- Work together and help each other!

EVERY GROUP PROJECT
DOES 99% OF THE WORK
HAS NO IDEA WHAT'S GOING ON THE WHOLE TIME
SAYS HE'S GOING TO HELP BUT HE'S NOT
DISAPPEAR AT THE VERY BEGINNING AND DOESN'T SHOW UP AGAIN TIL THE VERY END
IN SCHOOL YOU HAVE EVER DONE
WeKnowMemes

Conflict management

- **Key to conflict management is honesty**
- It can happen that you have less experienced people in the group – then worst case everyone should try the assignments but use code of more experienced group members or you sit next to each other and code together.
- Do not give up! You need the knowledge from the assignments for the final exam!
- If you do not even try to contribute – you can get kicked from a group = no grade (tell me first!)

Plagiarism

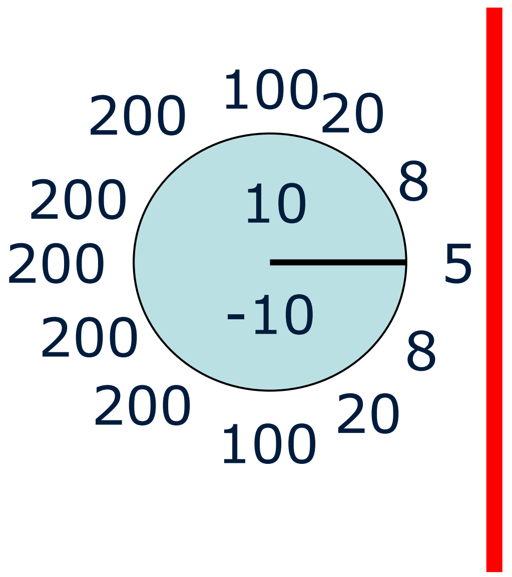
- This is a group assignment
- Help other members of your group
- Do not copy and hand in code from other groups
- You can use libraries for calculating intersections between lines, between lines and circles, and for visualization
- Write your own software

Write simple software

- No need to use (a lot of) objects
- Use functions
- Do not distribute code over too many files
- Avoid complex constructions and data structures

Summary: Goal of Assignment Mobile Robot Simulator

- Allow me to move the robot in a closed room using the buttons of your PC's keyboard
- If robot collides with wall: move robot realistically along side of walls (without entering the wall)
- Display values of distance sensors as plain numbers
- Display motor speed values as plain numbers



Hand in

- Documented code (Python, C++, C, Java, Matlab)
  - Make sure that each group member codes something, add names to code (who did what?)
  - Upload zip archive to CANVAS
- Video (mp4 only!, 5-6 minutes max, 150MB max)
  - Explain with your own voice
  - All team members should explain something
  - Analyse your results
  - Present and discuss all test cases from next slide

Test cases to be demonstrated in video

- Move simulated robot in room, demonstate steering behaviour of differential drive (moving forward/backwards, turning left/right while moving forward, turning on spot)
- Approach a wall slowly → so we can see that sensor values makes sense
- Approach a wall very fast & perpendicular → so we can see that robot does not glitch through wall (do not cheat by limiting speed)
- Approach a wall very fast with an angle of 45 degrees → so we can see that robot is sliding along the wall when hitting the wall

QUESTIONS?

LET‘S START