Wheeled mobile robots

Description, Kinematics, Modeling, Planning

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

December 20, 2018

Agenda

- Types of wheels and wheeled robots
- Moving around
- Kinematics, modeling
- Navigation, planning

Why wheeled robots?

Why wheeled robots are useful?
Provide some examples of applications

Wheels

Types of wheels

Wheels

Types of wheels

- Fixed wheel

Wheels

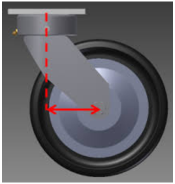
Types of wheels

- Fixed wheel
- Centered wheel

Wheels

Types of wheels

- Fixed wheel
- Centered wheel
- Off-centered wheel



Wheels

Types of wheels

- Fixed wheel
- Centered wheel
- Off-centered wheel
- Swedish wheel

Wheeled robots

Wheel configuration

Wheeled robots are categorized based on the type of wheels and configurations that they use.

- By-wheel
- Tricycle
- Four wheel
- Omnidirectional
- etc. etc.

Wheeled robots

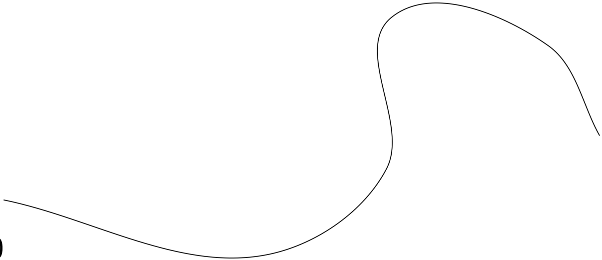
Instantaneous center of rotation

Every motion can be modeled as a rotation around a point. For a circular motion, this point is fixed, but for a more complex motion it is constantly moving.

Wheeled robots

Instantaneous center of rotation

Every motion can be modeled as a rotation around a point. For a circular motion, this point is fixed, but for a more complex motion it is constantly moving.

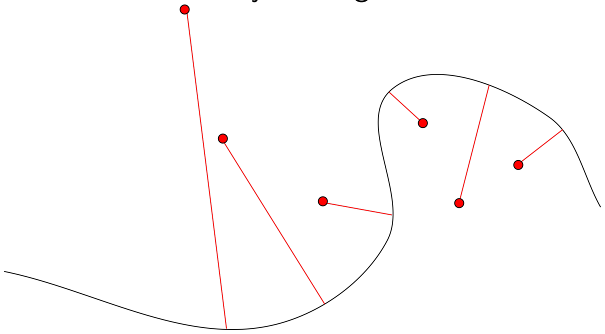


50

Wheeled robots

Instantaneous center of rotation

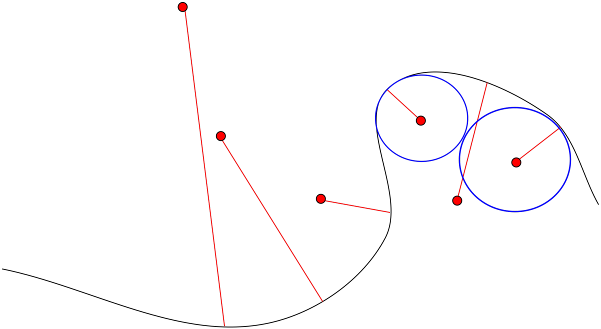
Every motion can be modeled as a rotation around a point. For a circular motion, this point is fixed, but for a more complex motion it is constantly moving.



Wheeled robots

Instantaneous center of rotation

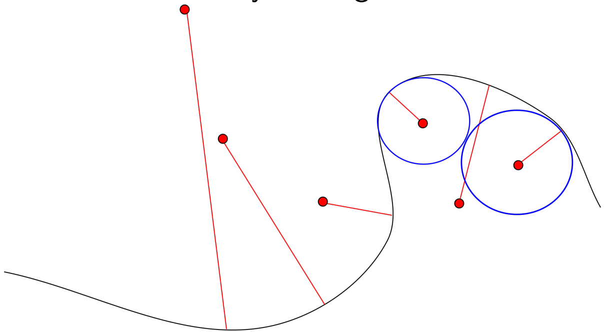
Every motion can be modeled as a rotation around a point. For a circular motion, this point is fixed, but for a more complex motion it is constantly moving.



Wheeled robots

Instantaneous center of rotation

Every motion can be modeled as a rotation around a point. For a circular motion, this point is fixed, but for a more complex motion it is constantly moving.



Where is the ICR for straight motion?

Wheeled robots

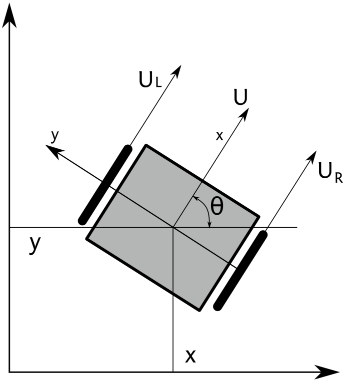
Differential drive

The differential drive is implemented:

- Two driving wheels
- Each can rotate independently
- Need for a third balancing point (usually a roller-ball)
- Sensitive to relative velocity of the two wheels

Differential drive

Kinematics modeling



Posture of the robot

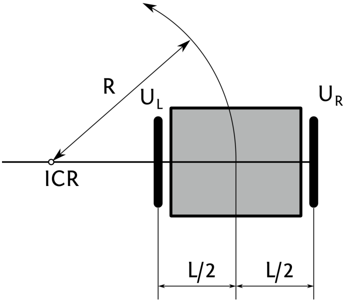
$$P = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Control input

$$U = \begin{bmatrix} u \\ \omega \end{bmatrix}$$

Differential drive

Kinematics modeling

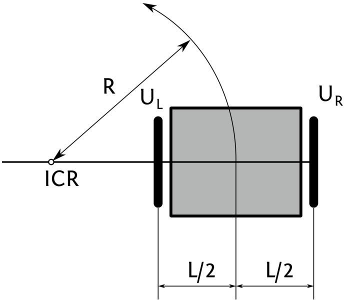


Definitions:

$U_R(t)$: Linear velocity of the right wheel
$U_L(t)$: Linear velocity of the left wheel
r: nominal radius of each wheel
R: Instantaneous Curvature Radius

Differential drive

Kinematics modeling



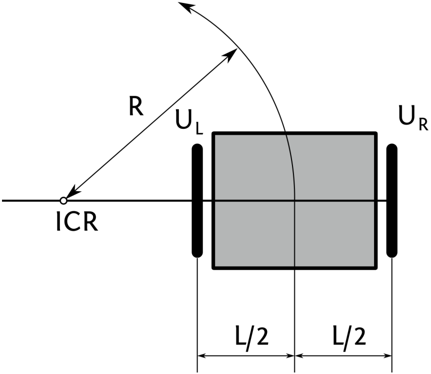
Definitions:

$U_R(t)$: Linear velocity of the right wheel
$U_L(t)$: Linear velocity of the left wheel
r: nominal radius of each wheel
R: Instantaneous Curvature Radius

If we want to follow a specific trajectory (i.e. a specific R), the wheels must move in such rate so the rotate around the ICR with the same angular velocity

Differential drive

Kinematics modeling



$$U_R = \omega(R + \frac{L}{2})$$

$$U_L = \omega(R - \frac{L}{2})$$

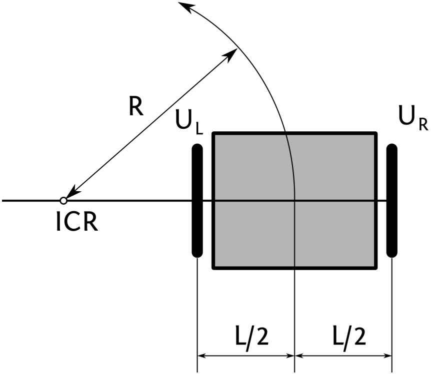
Where:

$$U_L = r\omega_L$$

$$U_R = r\omega_R$$

Differential drive

Kinematics modeling



$$U_R = \omega(R + \frac{L}{2})$$

$$U_L = \omega(R - \frac{L}{2})$$

Where:

$$U_L = r\omega_L$$

$$U_R = r\omega_R$$

We can also observe:

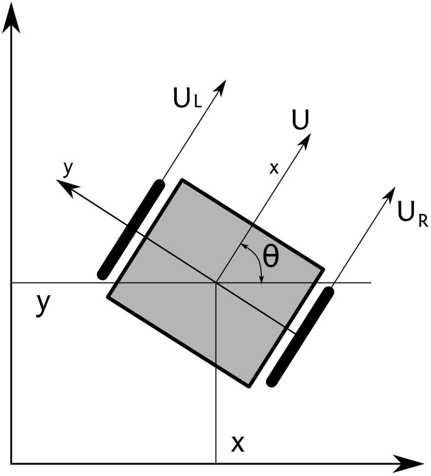
$$\omega = \frac{U_R - U_L}{L}$$

$$u = \frac{U_R + U_L}{2}$$

Differential drive

Kinematics modeling

Kinematics model in the world frame

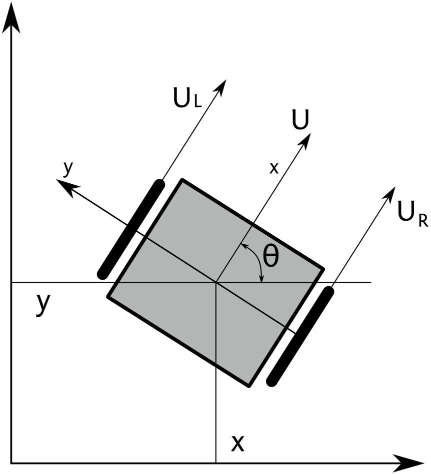


$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}$$

Differential drive

Kinematics modeling

Kinematics model in the robot frame

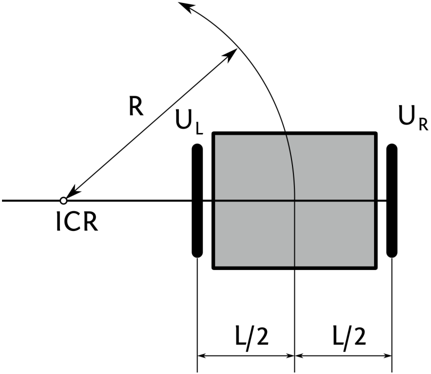


$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{-r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix}$$

Differential drive

Kinematics modeling

Controlling the Instantaneous center of rotation



From:

$$U_R = \omega(R + \tfrac{L}{2})$$

$$U_L = \omega(R - \tfrac{L}{2})$$

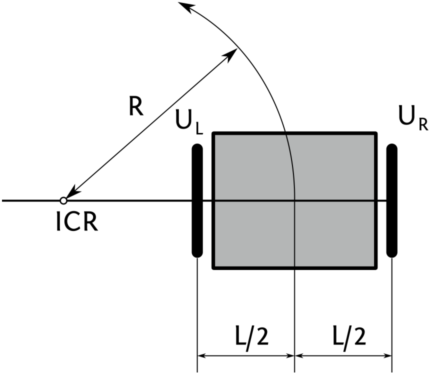
we can obtain:

$$R = \frac{L}{2}\frac{U_R + U_L}{U_R - U_L}$$

Differential drive

Kinematics modeling

Controlling the Instantaneous center of rotation



From:

$$U_R = \omega(R + \frac{L}{2})$$

$$U_L = \omega(R - \frac{L}{2})$$

we can obtain:

$$R = \frac{L}{2}\frac{U_R + U_L}{U_R - U_L}$$

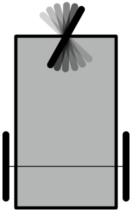
What happens when $U_R = U_L$ or $U_R = -U_L$?

Tricycle

Description

A wheeled robot with three wheels:

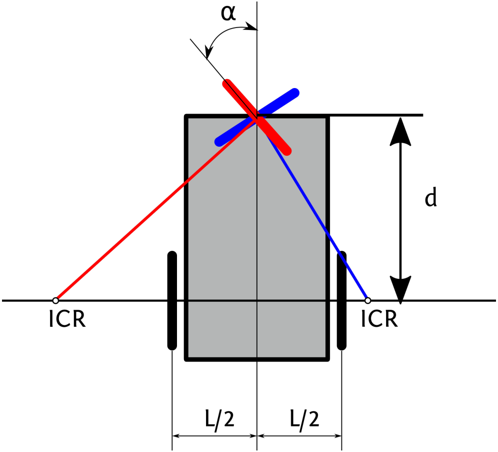
- Two fixed wheels with the same axis
- They two wheels can move independently
- One wheel that steers and pushes the robot
- The third wheel is usually between the other two with an offset



Tricycle

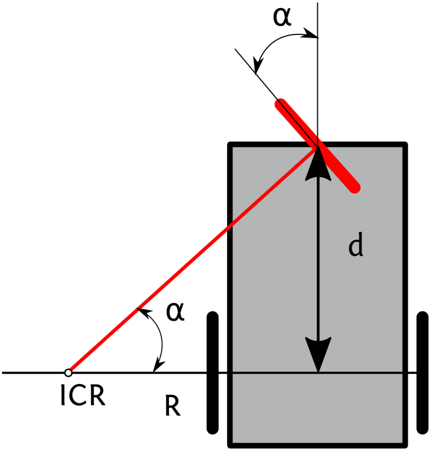
Kinematics

We control the location of the ICR by changing the steering angle α, and the velocity, by changing the wheel velocity ω



Tricycle

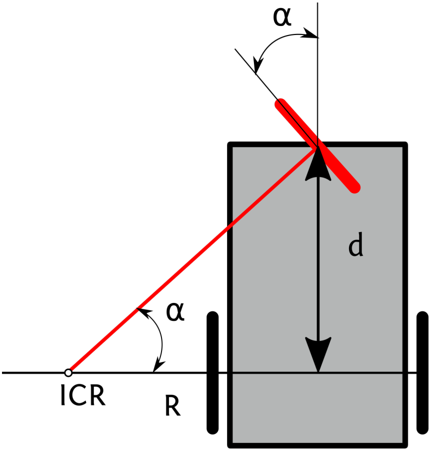
Kinematics model



If r is the steering wheel radius, then:

Tricycle

Kinematics model

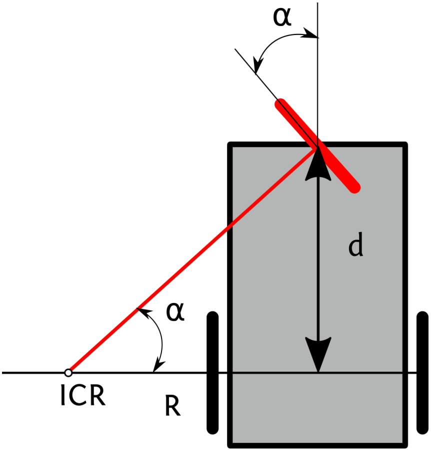


If r is the steering wheel radius, then:

$$u_s = \omega r$$
$$R(t) = d * tan(\frac{\pi}{2} - \alpha(t))$$

Tricycle

Kinematics model



If r is the steering wheel radius, then:

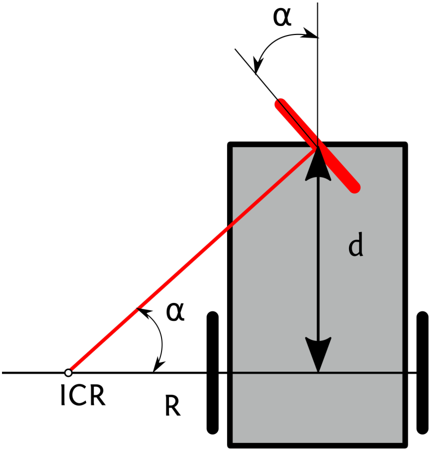
$$u_s = \omega r$$
$$R(t) = d * tan(\frac{\pi}{2} - \alpha(t))$$

The angular velocity of the robot relative to the base frame:

$$\dot{\theta}(t) = \frac{u_s(t)}{\sqrt{d^2 + R^2(t)}}$$

Tricycle

Kinematics model



If r is the steering wheel radius, then:

$u_s = \omega r$

$R(t) = d * tan(\frac{\pi}{2} - \alpha(t))$

The angular velocity of the robot relative to the base frame:

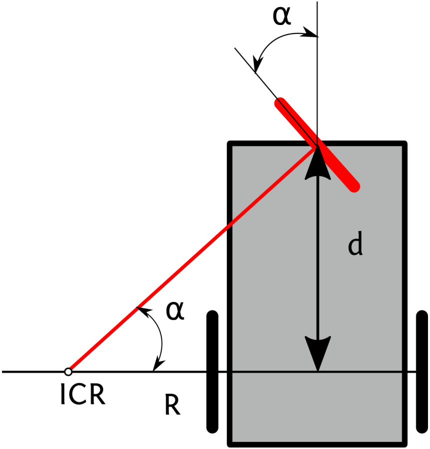
$$\dot{\theta}(t) = \frac{u_s(t)}{\sqrt{d^2 + R^2(t)}}$$

$$\dot{\theta}(t) = \frac{u_s(t)}{d} sin(\alpha(t))$$

Tricycle

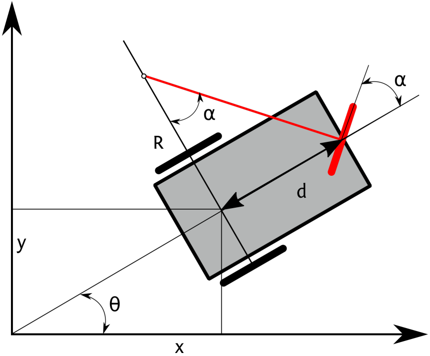
Kinematics model

Kinematics model in the robot body frame:



$$\dot{x}(t) = u_s(t)cos(\alpha(t))$$
$$\dot{y}(t) = 0$$
$$\dot{\theta}(t) = \frac{u_s(t)}{d}sin(\alpha(t))$$

Kinematics model in the world body frame:

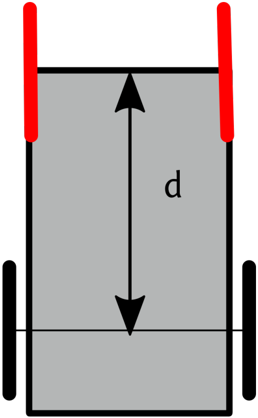


$$\dot{x} = u_s cos(\alpha(t))\cos(\theta(t))$$
$$\dot{x} = u_s cos(\alpha(t))\sin(\theta(t))$$
$$\dot{\theta}(t) = \frac{u_s(t)}{d} sin(\alpha(t))$$

Four wheels

Description

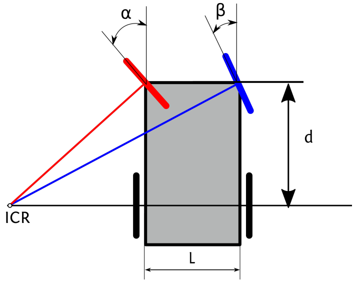
Another type of wheeled robot, is with four wheels. The two front are transmitting the power and are steered, while the back ones are fixed wheels



Four wheels

Ackerman drive

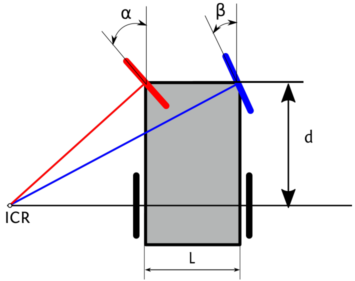
For this to work, the steering of the two wheels must be coordinated:



Four wheels

Ackerman drive

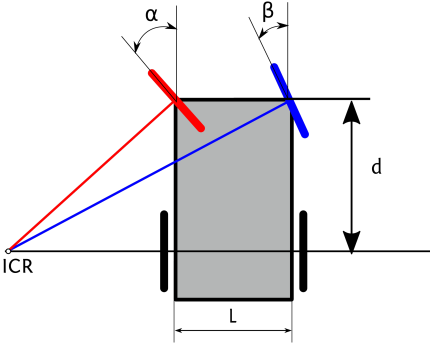
For this to work, the steering of the two wheels must be coordinated:



$\alpha > \beta$: when turning left
$\beta > \alpha$: when turning right

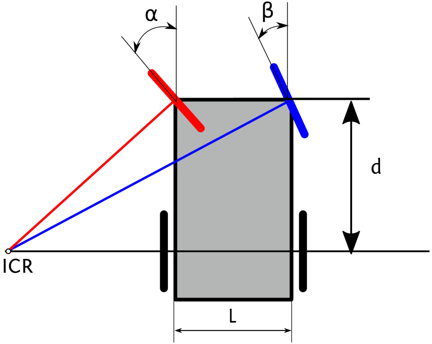
Four wheels

Ackerman drive



Four wheels

Ackerman drive

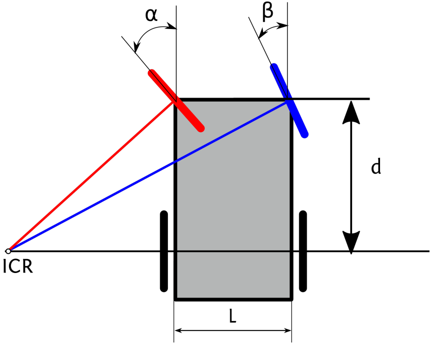


$$cot(\alpha) = \frac{R - \frac{L}{2}}{d}$$

$$cot(\beta) = \frac{R + \frac{L}{2}}{d}$$

Four wheels

Ackerman drive



$$cot(\alpha) = \frac{R - \frac{L}{2}}{d}$$

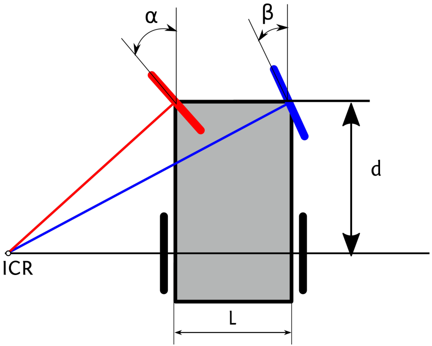
$$cot(\beta) = \frac{R + \frac{L}{2}}{d}$$

Therefore:

$$cot(\beta) - cot(\alpha) = \frac{L}{d}$$

Four wheels

Ackerman drive



$$cot(\alpha) = \frac{R - \frac{L}{2}}{d}$$

$$cot(\beta) = \frac{R + \frac{L}{2}}{d}$$

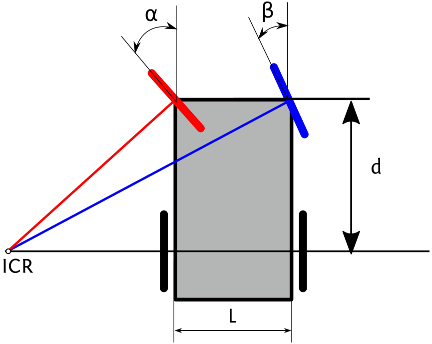
Therefore:

$$cot(\beta) - cot(\alpha) = \frac{L}{d}$$

What happens when $\alpha = \beta = 0$?

Four wheels

Ackerman drive



$$cot(\alpha) = \frac{R - \frac{L}{2}}{d}$$

$$cot(\beta) = \frac{R + \frac{L}{2}}{d}$$

Therefore:

$$cot(\beta) - cot(\alpha) = \frac{L}{d}$$

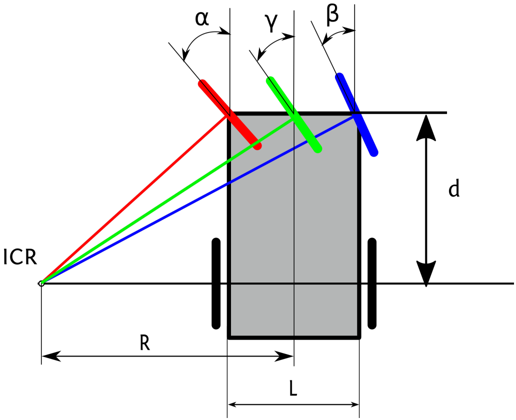
What happens when $\alpha = \beta = 0$?

What is the relationship between angular velocities ω_L and ω_R?

Four wheels

Ackerman and tricycle

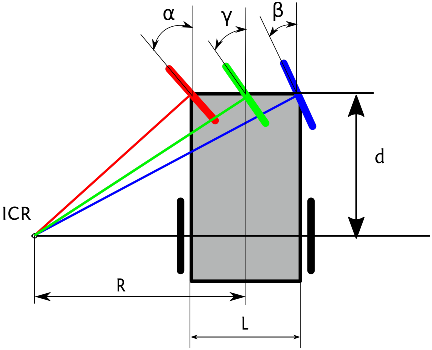
We can describe the ackerman drive kinematics, the same way as for the tricycle, if we consider a virtual fifth wheel between the two front ones



Four wheels

Ackerman drive

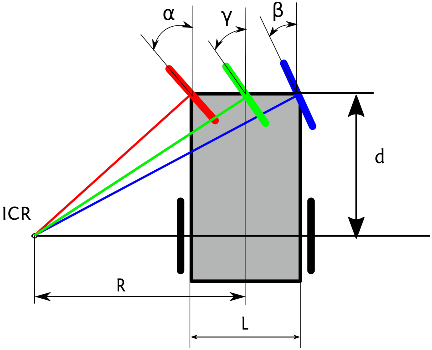
We can easily calculate the equivalent virtual angle γ



Four wheels

Ackerman drive

We can easily calculate the equivalent virtual angle γ



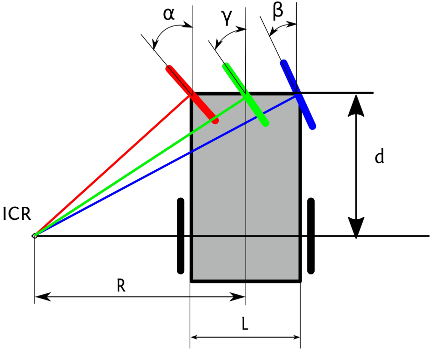
$$cot(\gamma) = cot(\alpha) + \frac{L}{2d}$$

$$cot(\gamma) = cot(\beta) - \frac{L}{2d}$$

Four wheels

Ackerman drive

We can easily calculate the equivalent virtual angle γ



$$cot(\gamma) = cot(\alpha) + \frac{L}{2d}$$

$$cot(\gamma) = cot(\beta) - \frac{L}{2d}$$

The kinematics models then are the same as for a tricycle with steering angle γ

Mobile robots

Motion planning

Why we need planning?

Mobile robots

Motion planning

Why we need planning?

- The world is full of obstacles

Mobile robots

Motion planning

Why we need planning?

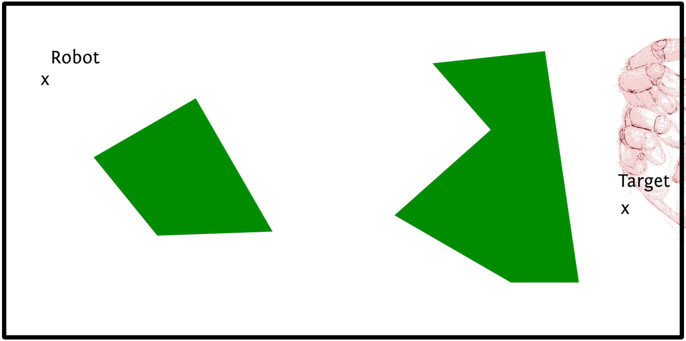
- The world is full of obstacles
- We want to find the most efficient way

Mobile robots

Motion planning

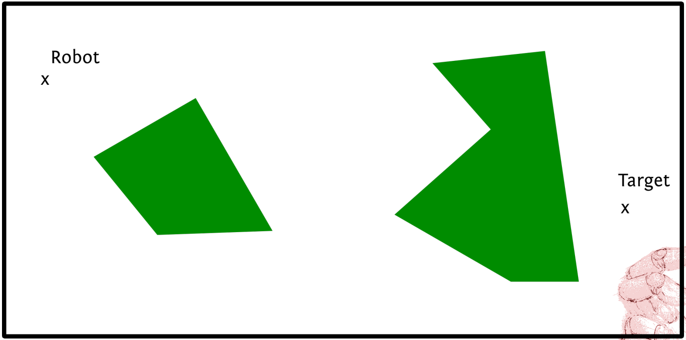
Why we need planning?

- The world is full of obstacles
- We want to find the most efficient way



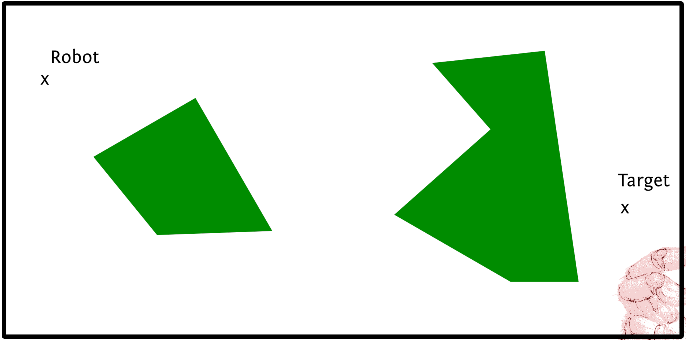
Mobile robots

Motion planning



Mobile robots

Motion planning

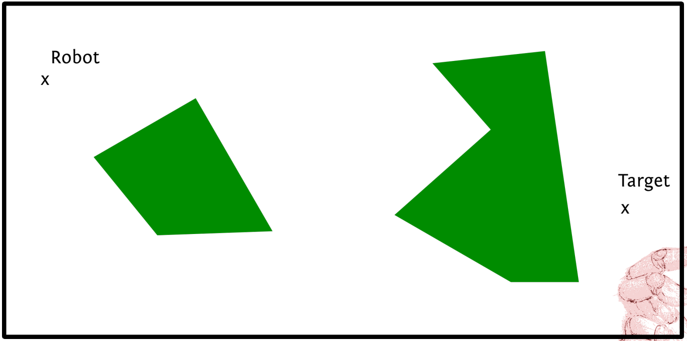


Input

- Geometric description of robot
- Geometric description of the environment
- Initial position and goal

Mobile robots

Motion planning



Input

- Geometric description of robot
- Geometric description of the environment
- Initial position and goal

Output

A path from the initial position until the goal

Mobile robots

Motion planning methods

Mobile robots

Motion planning methods

Roadmap approaches:
Reduce all the possible paths to a subset of them

Mobile robots

Motion planning methods

Roadmap approaches:
Reduce all the possible paths to a subset of them

Cell decomposition:
Account for all of the free space

Mobile robots

Motion planning methods

Roadmap approaches:
Reduce all the possible paths to a subset of them

Cell decomposition:
Account for all of the free space

Potential fields:
Local control strategies, optimality

Mobile robots

Motion planning methods

Roadmap approaches:
Reduce all the possible paths to a subset of them

Cell decomposition:
Account for all of the free space

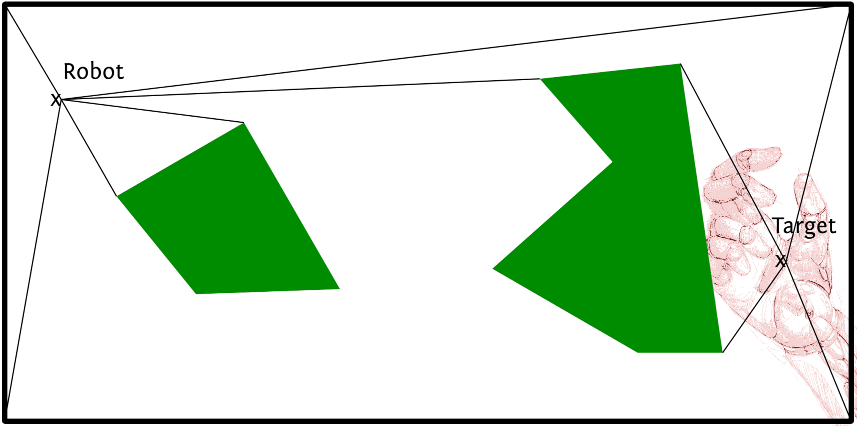
Potential fields:
Local control strategies, optimality

Bug algorithms:
Limited knowledge of environment

Motion planning

Roadmap approaches

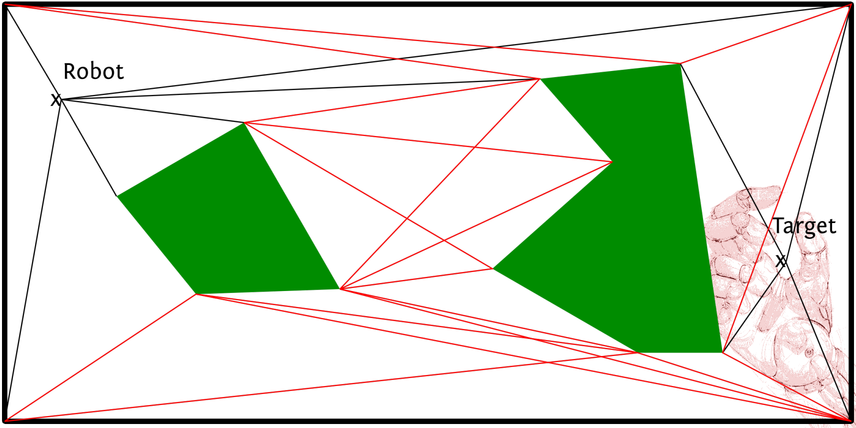
We construct by drawing lines of 'sight' from the initial position and target to all their 'visible' vertices.



Motion planning

Roadmap approaches

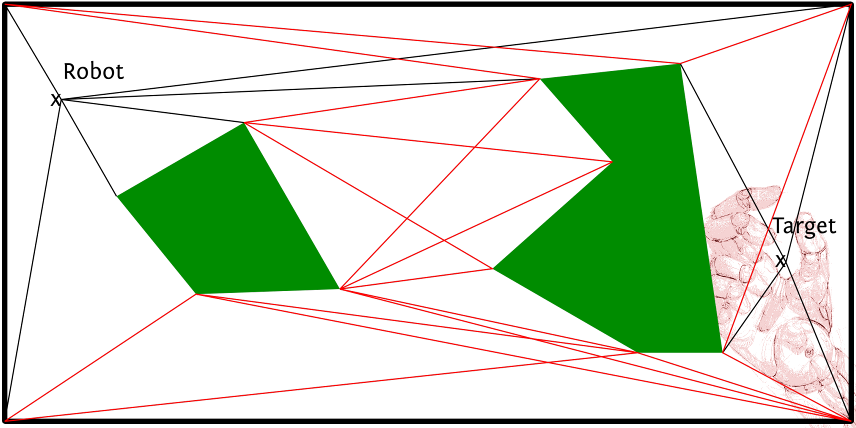
We then do the same for all vertices



Motion planning

Roadmap approaches

We then do the same for all vertices



These are the possible paths for our robot. By searching, we can find the shortest ones.

Motion planning

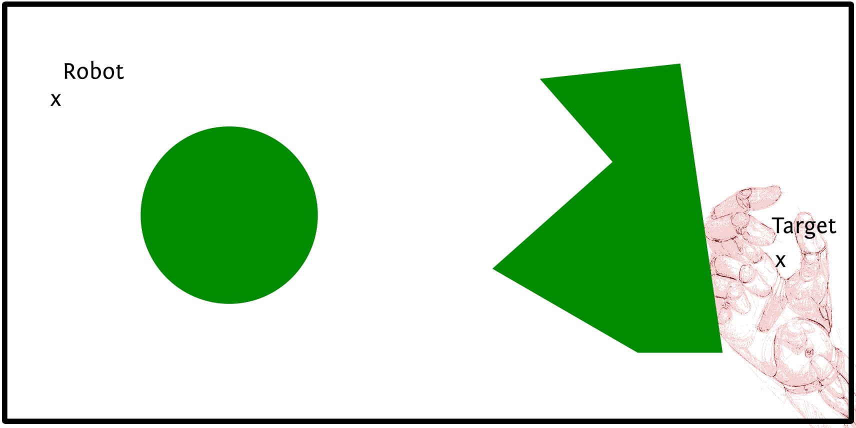
Roadmap approaches

Do you see any drawback with this technique?

Motion planning

Roadmap approaches

Do you see any drawback with this technique?

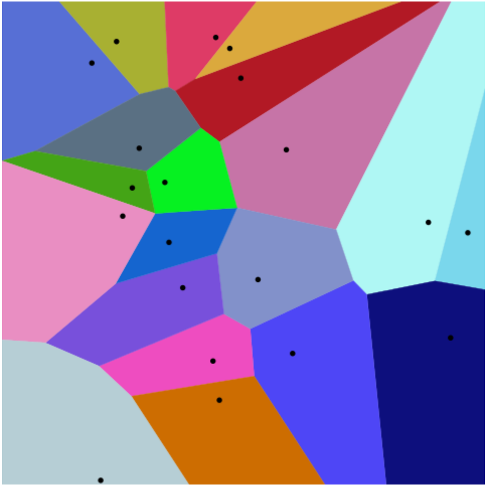


What do we do here?

Motion planning

Voronoi diagrams

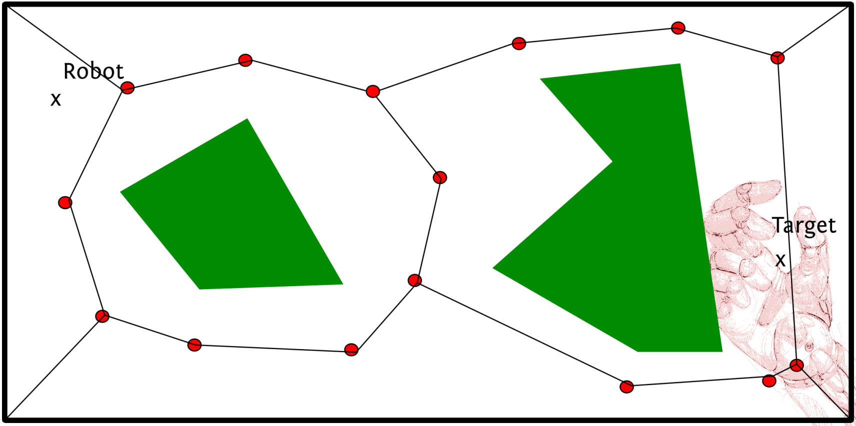
A Voronoi diagram is a partitioning of a plane so that different areas are the closest to a specific point.



Motion planning

Voronoi diagrams

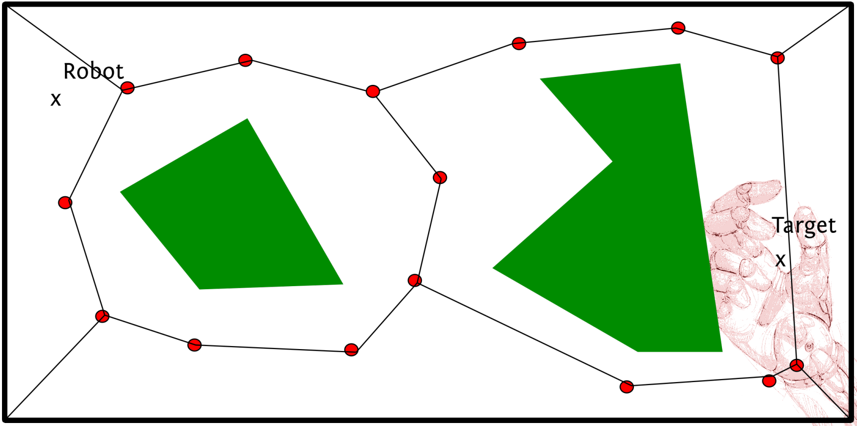
We use a very similar approach for constructing lines that are equally appart from obstacles



Motion planning

Voronoi diagrams

We use a very similar approach for constructing lines that are equally appart from obstacles



This technique can be used for curved surfaces, and it also generates clearence for the robot

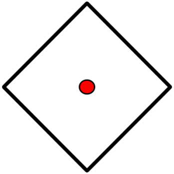
There are different metrics for defining the distance:

Motion planning
Voronoi diagrams

There are different metrics for defining the distance:

L1 metric:

$$(x, y) : |x| + |y| = const$$

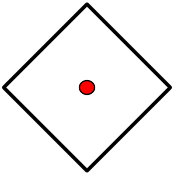


Motion planning
Voronoi diagrams

There are different metrics for defining the distance:

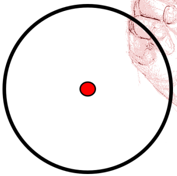
L1 metric:

$$(x, y) : |x| + |y| = const$$



L2 metric:

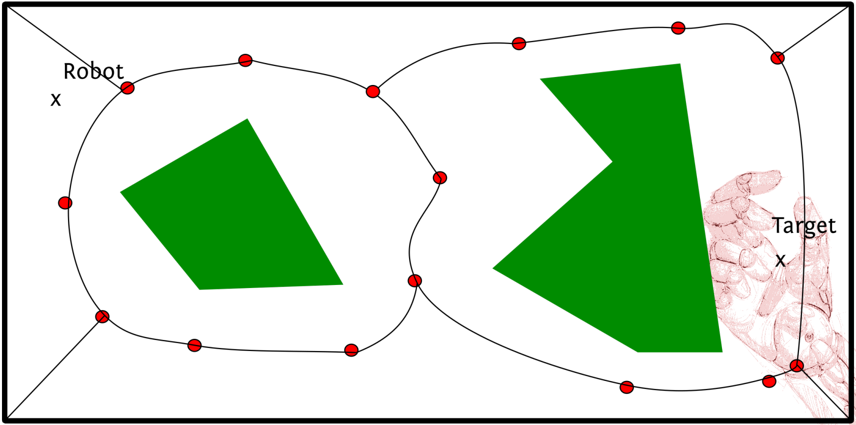
$$(x, y) : x^2 + y^2 = const$$



Motion planning

Voronoi diagrams

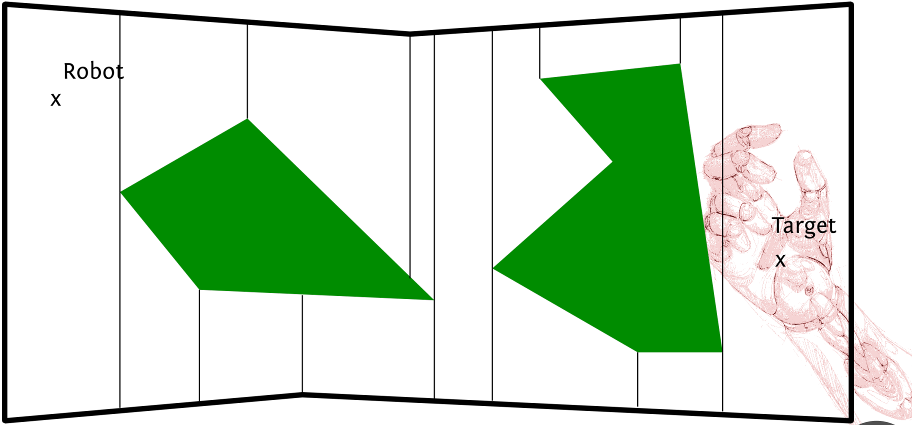
Different metrics, result in different paths



Motion planning

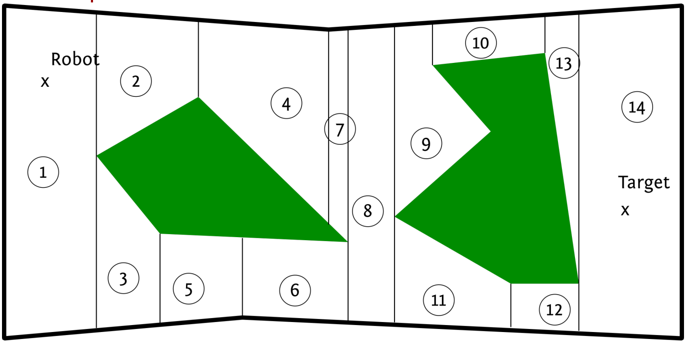
Cell decomposition

We decompose the available space into cells, and we create a connectivity graph, which helps us identify possible paths. There are different ways of performing the decomposition.



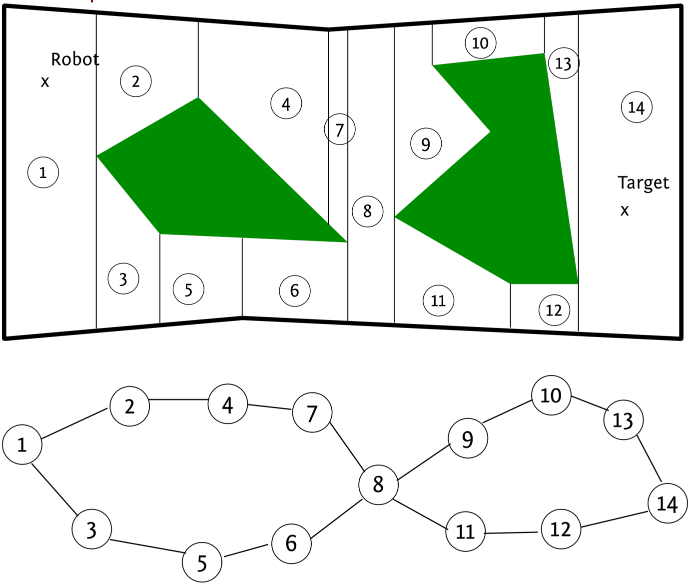
Motion planning

Cell decomposition



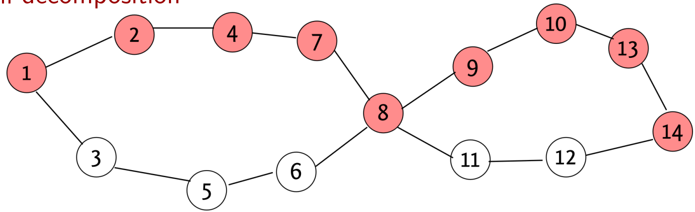
Motion planning

Cell decomposition



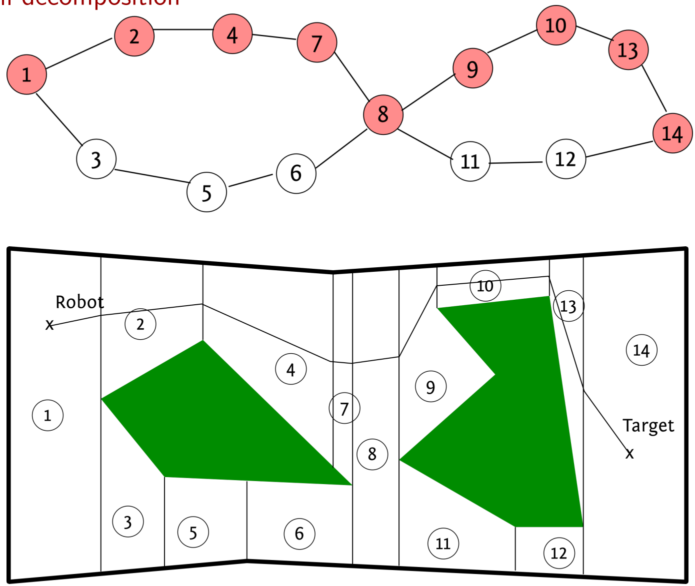
Motion planning

Cell decomposition



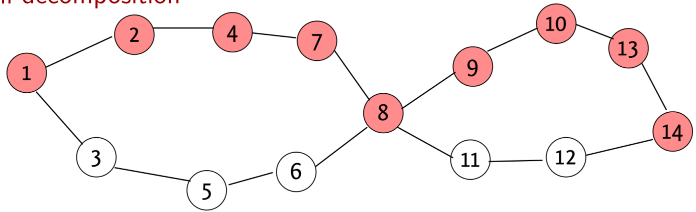
Motion planning

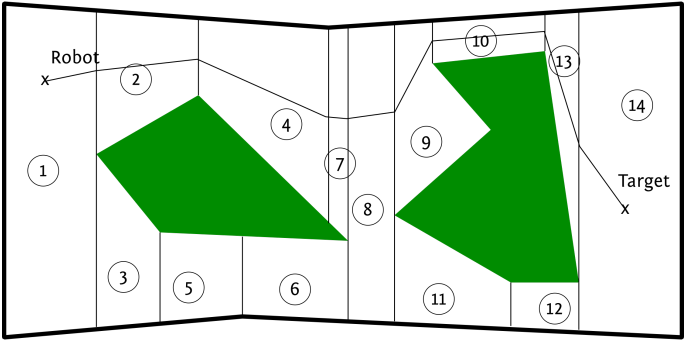
Cell decomposition



Motion planning

Cell decomposition





This

Motion planning

Potential field method

We construct a potential function that 'pulls' our robot towards the goal and is being 'pushed' by the obstacles. To do this, we need to:

Motion planning

Potential field method

We construct a potential function that 'pulls' our robot towards the goal and is being 'pushed' by the obstacles. To do this, we need to:

- Generate an attractive potential function centered at the goal

Motion planning

Potential field method

We construct a potential function that 'pulls' our robot towards the goal and is being 'pushed' by the obstacles. To do this, we need to:

- Generate an attractive potential function centered at the goal
- Generate repulsive potential functions at the edges of the obstacles

Motion planning

Potential field method

We construct a potential function that 'pulls' our robot towards the goal and is being 'pushed' by the obstacles. To do this, we need to:

- Generate an attractive potential function centered at the goal
- Generate repulsive potential functions at the edges of the obstacles
- Add the two together to come up with a complex potential function

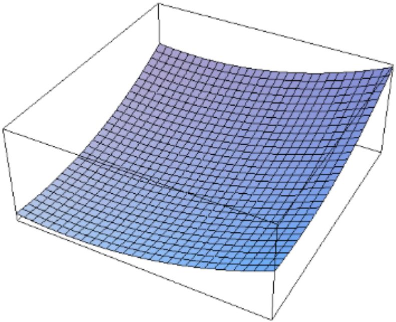
Motion planning

Potential field method

We construct a potential function that 'pulls' our robot towards the goal and is being 'pushed' by the obstacles. To do this, we need to:

- Generate an attractive potential function centered at the goal
- Generate repulsive potential functions at the edges of the obstacles
- Add the two together to come up with a complex potential function
- The gradient of the total potential is an artificial force that drives the robot. This ensures optimal path

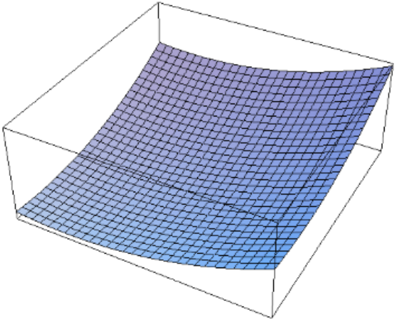
Attractive field



Motion planning

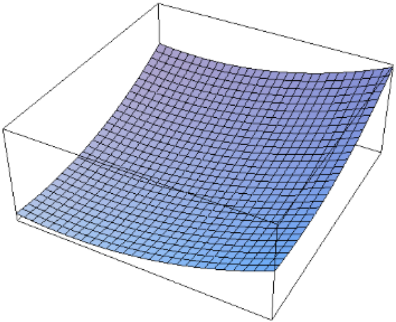
Potential field method

Attractive field



$$U_{at} = \frac{1}{2}\xi\left\|q - q_{goal}\right\|^2$$

Attractive field



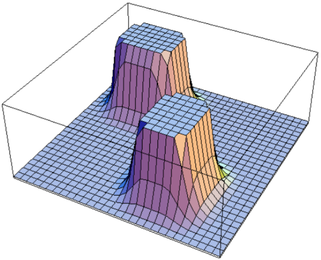
$$U_{at} = \frac{1}{2}\xi\left\|q - q_{goal}\right\|^2$$

The potential is parabolic and centered at position q_{goal}

Motion planning

Potential field method

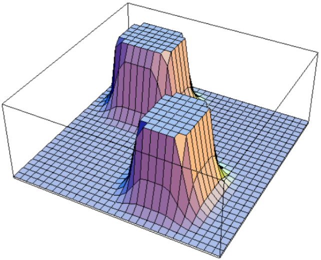
Repulsive field



Motion planning

Potential field method

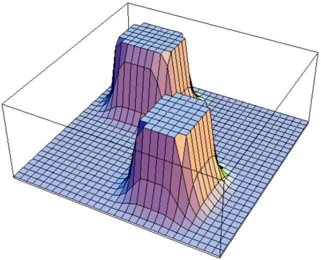
Repulsive field



$$U_{rep} = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2, & \text{if } \rho(q) \leq \rho_0 \\ 0, & \text{if } \rho(q) > \rho_0 \end{cases}$$

The potential is reciprocal with distance ρ, which is the distance from the edge of the obstacle.

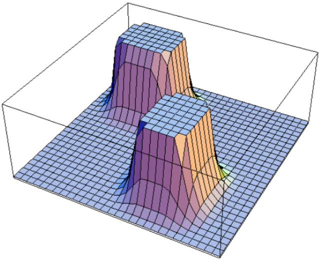
Repulsive field



$$U_{rep} = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2, & \text{if } \rho(q) \leq \rho_0 \\ 0, & \text{if } \rho(q) > \rho_0 \end{cases}$$

The potential is reciprocal with distance ρ, which is the distance from the edge of the obstacle. We want the effect of the repulsion to wear off after distance ρ_0

Repulsive field

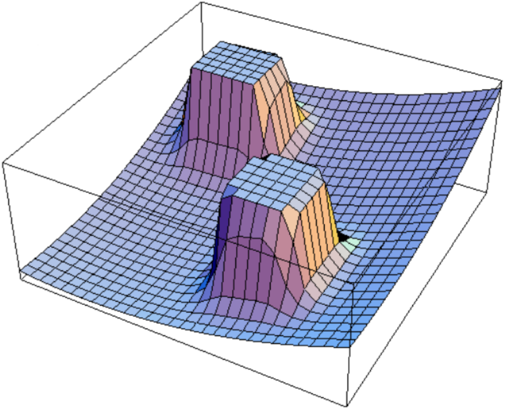


$$U_{rep} = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2, & \text{if } \rho(q) \leq \rho_0 \\ 0, & \text{if } \rho(q) > \rho_0 \end{cases}$$

Motion planning

Potential field method

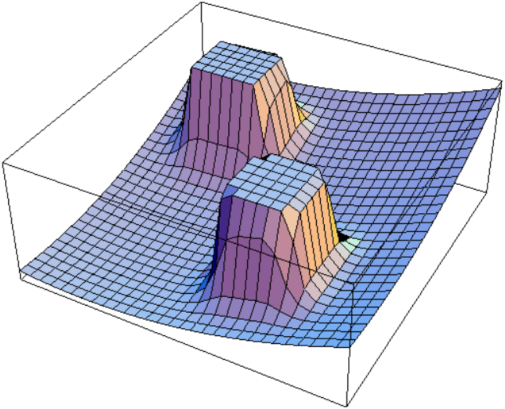
When adding the two potentials, we get a complex potential that can guide our robot



Motion planning

Potential field method

When adding the two potentials, we get a complex potential that can guide our robot



Of course, this isn't a perfect solution. What do you think are its limitations?

Questions?