Autonomous Vehicle Guidance Systems

ENG:4175 & ENG:5017

Dr Dezong Zhao
Reader in Autonomous Systems

James Watt School of Engineering

Course Notes 2024

Before we continue . . .

- Please stay on Mute.
- Please type questions in Chatbox if you have any. They will be answered after the lecturing.
- Lecturing videos with transcripts will be shared on Moodle after class.

Course Overview

- Class website: Moodle

- Assessment...
  - Final Exam (80%)
  - Term Projects (20%): Four term projects 1, 2, 3 and 4

- Text Books (Recommended reading only)
  - Stengel, R.F. Optimal Control & Estimation, Dover, 1994.
  - Siegwart, R., Nourbakhsh, I.R., Introduction to autonomous mobile robots, MIT Press, 2004.
  - Bryson. A. E. Jr. and Ho, Y.-C., Applied optimal control: Optimization, Estimation, and Control, John Wiley & Sons (Chapter 1 & 2)
  - Bertsekas, D., Dynamic programming and optimal control, Vol. 1, Athena Scientific, (Chapter 1 & 2)
  - Several journal papers (available on the moodle site)

Course Overview

- Introduction & Simple Guidance Method
  - Autonomous vehicle path planning
  - Obstacle avoidance using potential functions (APF)
- Continuous path planning
  - Parameter optimisation
  - Optimisation for dynamic systems
- Discrete path planning
  - Discretise the problem space
  - Dynamic programming algorithm
  - Deterministic shortest path problem
- Path following

Introduction

What is an Autonomous System?

Autonomous systems become increasingly crucial to human life in almost every direction.

Diverse types of autonomous systems.

Demo video 1: *Autonomous farming of Spot from Boston Dynamics*.
Demo video 2: *Choreographed demonstration of Atlas, Spot and Handle*.

What is an Autonomous System?

Before we proceed further on this course, it is worth to *define* autonomy and in particular autonomous vehicles. Best to define *levels* of autonomy.

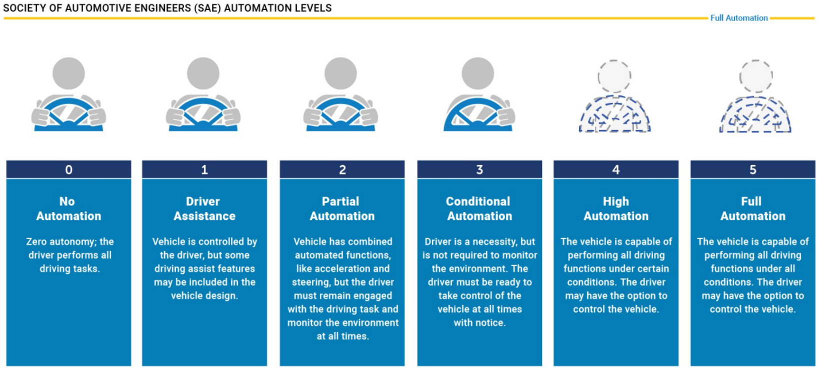
Level	Name	Description
1	Human operated	All activity within the system is the direct result of human-initiated control inputs. The system has no autonomous control of its environment, although it may have information-only responses to sensed data.
2	Human assisted	The system can perform activity in parallel with human input, acting to augment the ability of the human to perform the desired activity, but has no ability to act without accompanying human input. An example is automobile automatic transmission and anti-skid brakes.
3	Human delegated	The system can perform limited control activity on a delegated basis. This level encompasses automatic flight controls, engine controls, and other low-level automation that must be activated or deactivated by a human input and act in mutual exclusion with human operation.
4	Human supervised	The system can perform a wide variety of activities given top-level permissions or direction by a human. The system provides sufficient insight into its internal operations and behaviours that it can be understood by its human supervisor and appropriately redirected. The system does not have the capability to self-initiate behaviours that are not within the scope of its current directed tasks.
5	Mixed initiative	Both the human and the system can initiate behaviours based on sensed data. The system can coordinate its behaviour with the human's behaviours both explicitly and implicitly. The human can understand the behaviours of the system in the same way that he understands his own behaviours. A variety of means are provided to regulate the authority of the system with respect to human operators.
6	Fully autonomous	The system requires no human intervention to perform any of its designed activities across all planned ranges of environmental conditions.

Office of Naval Research (ONR) definition of autonomy.

Researchers desire to move autonomous systems down this table to level 6.

What is an Autonomous System?

The road to full automation:

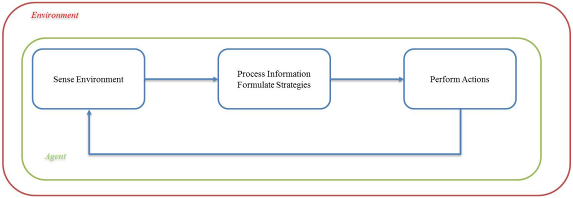


SAE Autonomy Level.

Reference: *Automated Vehicles for Safety*

What is an Autonomous System?

But how does an automated system operate without human intervention?



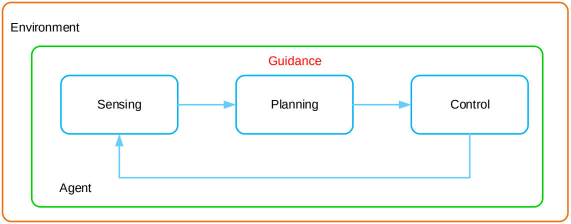
Information block diagram of an autonomous agent.

Every autonomous system requires subsystems to perform three functions,

- Perception: where am I?
- Cognition: where am I going?
- Actuation: how can I get there?

What is an Autonomous Vehicle?

Autonomous vehicles are essentially autonomous decision-making systems.



Functioning blocks in an autonomous vehicle.

Sensor perception and prior knowledge are required in situation awareness.

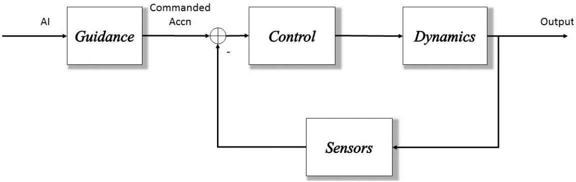
Decision making process is hierarchical, including planning (*guidance*) and control.

Guidance v.s. Control

If the purpose is to move the vehicle from the start point to the end, surely this is a *control* problem?

Guidance and control are closely-related. The main difference is in the level of detail used to define the vehicle dynamics.

- Guidance - simple model
- Control (autopilot) - complex dynamics model

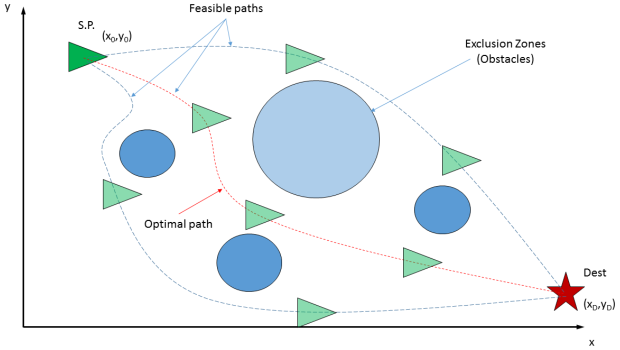


In most cases, the output from the guidance loop is used to drive the control loop.

Demo videos: *Guidance* and *Control*. What is the difference between the planned and the actual trajectory?

Autonomous Vehicle Guidance

The guidance problem is one of calculating a *feasible* path from the start point $S.P.$ to destination D that is *optimal* in some sense.



Typical AVGS problem.

If we want to choose the optimal path, we need to encapsulate the problem mathematically. This is done by first defining a *cost function*.

Cost Functions

The purpose of a cost function is to mathematically encapsulate a design objective. For example, consider the design objective of minimising manoeuvre time,

$$J_1 = \int_{\text{S.P.}}^{\text{Dest}} 1 \, \mathrm{d}t$$

Alternatively, minimising the path length would be,

$$J_2 = \int_{\text{S.P.}}^{\text{Dest}} 1 \, \mathrm{d}l$$

Cost functions can also be combined,

$$J = \alpha J_1 + (1 - \alpha) J_2$$

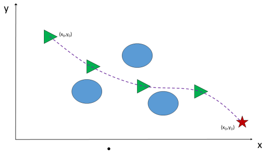
where $\alpha \in [0, 1]$

Vehicle Models

In guidance, there are two similar methods for defining the vehicle dynamics,

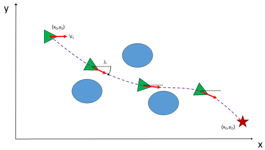
- double integrator
- velocity-heading

Double integrator



$$\dot{x} = v_x$$
$$\dot{y} = v_y$$
$$\dot{v}_x = a_x$$
$$\dot{v}_y = a_y$$

Velocity-Heading



$$\dot{x} = V_f \cos\lambda$$
$$\dot{y} = V_f \sin\lambda$$
$$\dot{v}_x = \dot{V}_f \cos\lambda - V_f \sin\lambda\dot{\lambda}$$
$$\dot{v}_y = \dot{V}_f \sin\lambda + V_f \cos\lambda\dot{\lambda}$$

Equations of motion...

Vehicle Models

Both model are point mass models and their formulations can be expressed in state-space form.

In the double-integrator model, define $\underline{x} \in \mathbb{R}^4 = [x, y, v_x, v_y]^{\mathrm{T}}$ and $\underline{u} \in \mathbb{R}^2 = [a_x, a_y]^{\mathrm{T}}$. Then

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \underline{u}$$

$$\Rightarrow \dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u}$$

Clearly, this is a linear state-space model. For the velocity-heading model, the equations of motion are fundamentally coupled and non-linear i.e.,

$$\dot{\underline{x}} = f(\underline{x}, \underline{u})$$

For simplicity, the double-integrator model will be used.

Continuous Path Planning

We are now ready to state the continuous path planning problem.

$$\min_{(a_x, a_y)} \quad J$$

subject to,

$$\begin{aligned} \dot{x} &= v_x, \quad \dot{y} = v_y \\ \dot{v}_x &= a_x, \quad \dot{v}_y = a_y \end{aligned}$$

and

$$\begin{aligned} t &= 0 : x(0), y(0), v_x(0) = 0, v_y(0) = 0 \\ t &= t_f : x(t_f) = x_D, y(t_f) = y_D, v_x(t_f) = 0, v_y(t_f) = 0 \end{aligned}$$

Often, there are constraints on the available accelerations,

$$\begin{aligned} a_{x,L} &\leq a_x \leq a_{x,U} \\ a_{y,L} &\leq a_y \leq a_{y,U} \end{aligned}$$

This is a nonlinear programming problem.

Example: Automated Parking Valet in Matlab.

Discrete Path Planning

One of the main issues in solving the general nonlinear program is the tremendous processor load required to find a solution. A practical alternative is to discretise the problem

Recall the definition of the derivative applied to the $x - \mathrm{position}$,

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{x(t + \Delta t) - x(t)}{\Delta t} = \dot{x}(t) = v_x(t)$$

Removing the limit, the timestep Δt becomes a design parameter. Rearranging the above equation,

$$x(t + \Delta t) = x(t) + v_x(t)\Delta t$$

Using the notation $t + \Delta t = t_{k+1}$, this becomes,

$$x_{k+1} = x_k + v_{x,k}\Delta t$$

This is Eulers method for numerical integration.

Discrete Path Planning . . .

We can then define a set of difference equations to describe the vehicle dynamics,

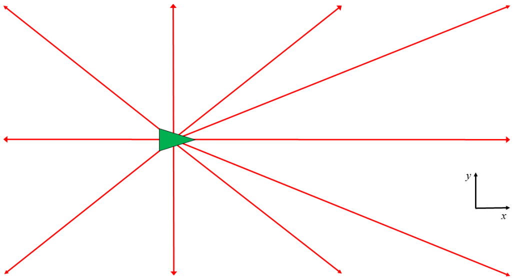
$$x_{k+1} = x_k + v_{x,k}\Delta t$$
$$y_{k+1} = y_k + v_{y,k}\Delta t$$
$$v_{x,k+1} = v_{x,k} + a_{x,k}\Delta t$$
$$v_{y,k+1} = v_{y,k} + a_{y,k}\Delta t$$

Now assume that the acceleration inputs are confined to a finite set, for example

$$a_{x,k} = [-5, 0, 5, 10]^{\mathrm{T}}$$
$$a_{y,k} = [-2, 0, 2]^{\mathrm{T}}$$

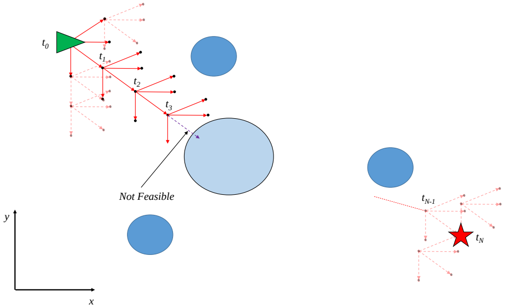
In this example, there are 12 possible combinations of acceleration input at each time step.

At each time step, the system can apply accelerations along one of the following paths,



Integrating the chosen acceleration twice over the timestep yields 12 possible new positions.

Repeating this process, we then have a *graph* of nodes spanning the operating environment,



The discrete path planning problem is then one of selecting the series of control inputs $\underline{u}_i, i = 1 \ldots N$ that is feasible and reaches the destination while minimising some discrete cost function.

Discrete Path Planning . . .

Three examples to see how discrete path planning works in a closed space:

- Arrive the target without colliding into obstacles.
- Find a path to connect all points avoiding obstacles.
- Find a shortest path between two points avoiding obstacles.

Simple Guidance Methods

Potential Function Guidance

The concept of a potential function should be familiar to most scientists and engineers. Potential functions are used to describe, for example,

- fluid flow around obstacles
- gravitational force
- electrostatic force

These are all examples of *conservative forces* i.e. the force can be represented by a potential field function U. For our purpose, we note a property of conservative forces i.e.,

$$F = -\nabla U = -\left[\frac{\partial U}{\partial x}\ \frac{\partial U}{\partial y}\right]^{\mathrm{T}}$$

Recalling Newton's second law, we can define acceleration using a potential function

$$a_x = \frac{F_x}{m} = -\frac{\partial U}{\partial x}\frac{1}{m}$$
$$a_y = \frac{F_y}{m} = -\frac{\partial U}{\partial y}\frac{1}{m}$$

To see how this works, consider the simple case of gravity. Gravitational potential energy on earth is $U = mgh$ where h is the height of the object. The gravitational force acting on an object of mass m is its weight W. It is easy to see that,

$$W = -\frac{\partial U}{\partial h} = -\frac{\partial (mgh)}{\partial h} = -mg$$

as expected. Extending this concept, an admissible potential function has the form $U(x, y) = scalar$. For example, lets set

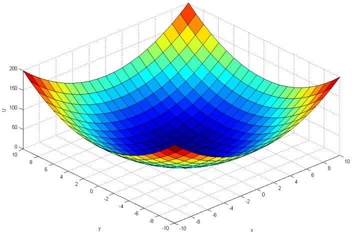
$$U = x^2 + y^2$$

Then

$$\begin{aligned} a_x &= \frac{F_x}{m} = -\frac{\partial U}{\partial x}\frac{1}{m} = \frac{-2x}{m} \\ a_y &= \frac{F_y}{m} = -\frac{\partial U}{\partial y}\frac{1}{m} = \frac{-2y}{m} \end{aligned}$$

Without loss of generality, set $m = 1$.

. . . Potential function

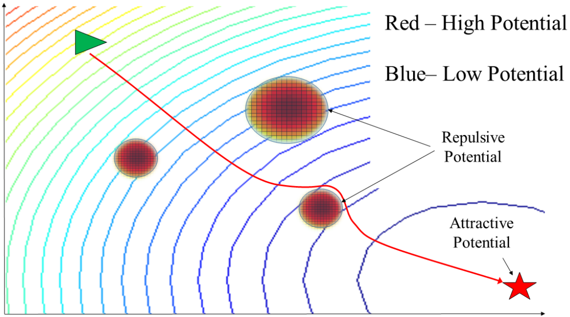


We then have,

$$a_x = \frac{\mathrm{d}v_x}{\mathrm{d}t} = \dot{v}_x = -2x$$
$$a_y = \frac{\mathrm{d}v_y}{\mathrm{d}t} = \dot{v}_y = -2y$$

i.e. this attractive potential forms a position feedback control law.

Returning to the guidance problem, how does specifying a potential function provide a guidance solution in the presence of obstacles?



Because we can specify regions of *attractive* and *repulsive* potential.

- Attractive Potential

$$U_a = \frac{1}{2} K_a \rho_a^2$$

where, $\rho_a = \sqrt{(x - x_a)^2 + (y - y_a)^2}$ is the distance from the point of attraction (destination) to the current vehicle position and K_a is a constant.

- Repulsive Potential

$$U_r = \begin{cases} 0 & \rho > \rho_0 \\ \frac{1}{2} K_r \left(\frac{1}{\rho_r} - \frac{1}{\rho_0} \right)^2 & \rho \le \rho_0 \end{cases}$$

where, $\rho_r = \sqrt{(x - x_{\mathrm{obs}})^2 + (y - y_{\mathrm{obs}})^2}$ is the distance from the point of repulsion (obstacle centre) to the current vehicle position and ρ_0 is a tuning parameter defining the size of the obstacles repulsive effect.

Derive Guidance Accelerations

- Attractive Force

Consider the attractive potential first. To find the conservative forces necessary to traverse the potential field, we need to find the gradient of the potential function,

$$\begin{aligned}\nabla U_a &= \nabla \left(\frac{1}{2} K_a \rho_a^2 \right) \\ &= \nabla \left(\frac{1}{2} K_a \left((x - x_a)^2 + (y - y_a)^2 \right) \right)\end{aligned}$$

Then,

$$F_a = -\nabla U_a = -\left[\frac{\partial U_a}{\partial x} \; \frac{\partial U_a}{\partial y} \right]^{\mathrm{T}}$$

and

$$\begin{aligned}F_{a_x} &= -K_a (x - x_a) \\ F_{a_y} &= -K_a (y - y_a)\end{aligned}$$

Derive Guidance Accelerations

- Repulsive Force

Now the repulsive potential. We only need to apply the gradient inside the i^{th} obstacle boundary.

$$\begin{aligned}\nabla U_r &= \nabla \left(\frac{1}{2} K_{r_i} \left(\frac{1}{\rho_{r_i}} - \frac{1}{\rho_0} \right)^2 \right) \\ &= \nabla \left(\frac{1}{2} K_{r_i} \left(\frac{1}{\sqrt{(x - x_{\text{obs}_i})^2 + (y - y_{\text{obs}_i})^2}} - \frac{1}{\rho_0} \right)^2 \right)\end{aligned}$$

Following some manipulation, it can be shown that

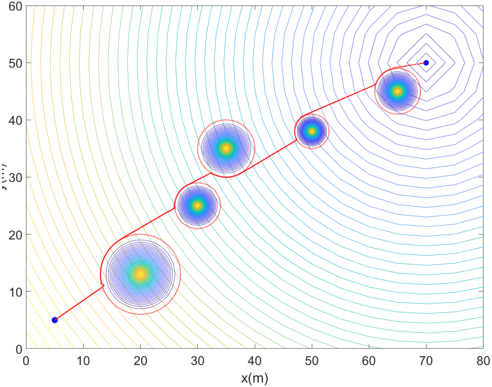
$$\begin{aligned}F_{r_x} &= K_{r_i} \left(\frac{1}{\rho_{r_i}} - \frac{1}{\rho_0} \right) \frac{(x - x_{\text{obs}_i})}{\rho_{r_i}^3} \\ F_{r_y} &= K_{r_i} \left(\frac{1}{\rho_{r_i}} - \frac{1}{\rho_0} \right) \frac{(y - y_{\text{obs}_i})}{\rho_{r_i}^3}\end{aligned}$$

Derive Guidance Accelerations

- Finally, we have,

$$F = F_a + \sum_{i=0}^{N_{\text{obs}}} F_{\text{obs}_i}$$

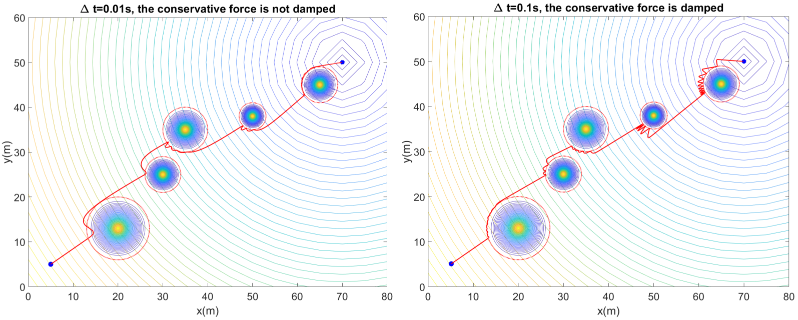
- A typical example



Known issues . . .

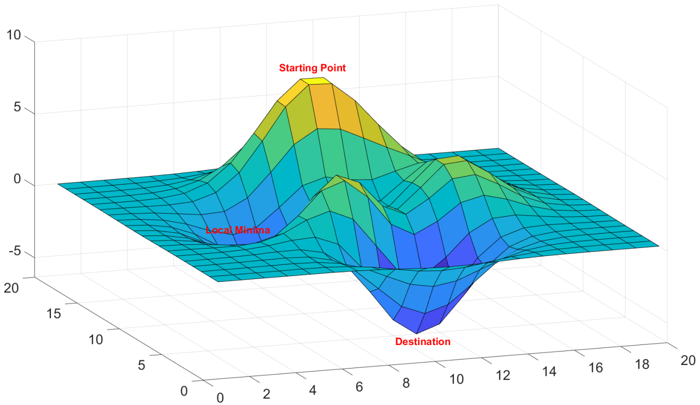
There are a few known issues with the potential function method,

- Chattering. This is where the vehicle oscillates around the final destination point, never coming to rest. Although the potential method should provide asymptotic behaviour, practical constraints can induce this effect. Too big step-size or accelerations may also cause oscillations in the trajectory.

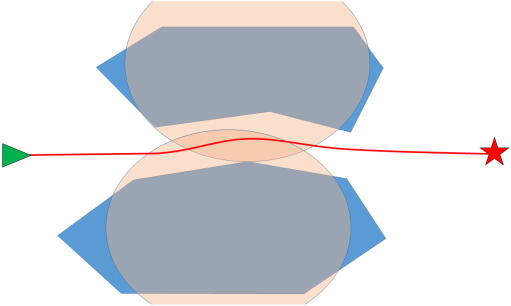


Known issues . . .

- Stagnation point. If the vehicle lands on the stagnation point of the potential field (i.e. the gradient of the potential field is zero in all directions) it cannot move towards the destination without additional information.



- Canyon path. If two obstacles of complex shape are close together, it can be difficult to define a suitable potential field to encourage path traversal between the obstacles.



Reference: *Mass-spring-damper model.*
Reference: *Table of Laplace transforms.*

Summary

- We focus on the path planning in a 2D space.
- All obstacles are assumed to be static.
- Normally the double integrator model is employed.
- Guidance methods can be continuous or discrete.
- The selection of methods is upon the real problem.
- Parameters are required tuning in implementation.

Term project 1

In term project 1, you are expected using APF guidance to navigate a drone through an obstacle field.

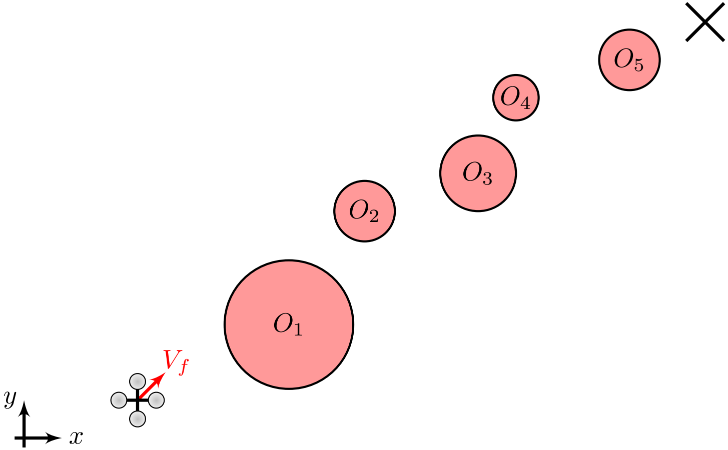


Figure TP1: Drone in an obstacle field

Term project 1

- Your task is to implement a path planning algorithm using the potential function method for the scenario shown in Figure TP1.2 above, with the following obstacle locations.

Obstacle	Position	Radius	K_r
1	(20,13)	7	500
2	(30,25)	3	250
3	(35,35)	6	400
4	(50,38)	4	300
5	(65,45)	4	300

  The initial drone position is $(5, 5)$ and the desired end-point is located at $(70, 50)$.

- Add the following additional force for each direction

$$F_{x_v} = -K_v \ v_x, \quad F_{y_v} = -K_v v_y$$