CS491/691: Introduction to Aerial Robotics

Topic: Robot Motion Planning

Dr. Kostas Alexis (CSE)

What is Motion Planning?

- **Determining where to go based on a set of goals and objectives**

What is Motion Planning?

Hawk Navigation

Cheetah running

Eagle hunting

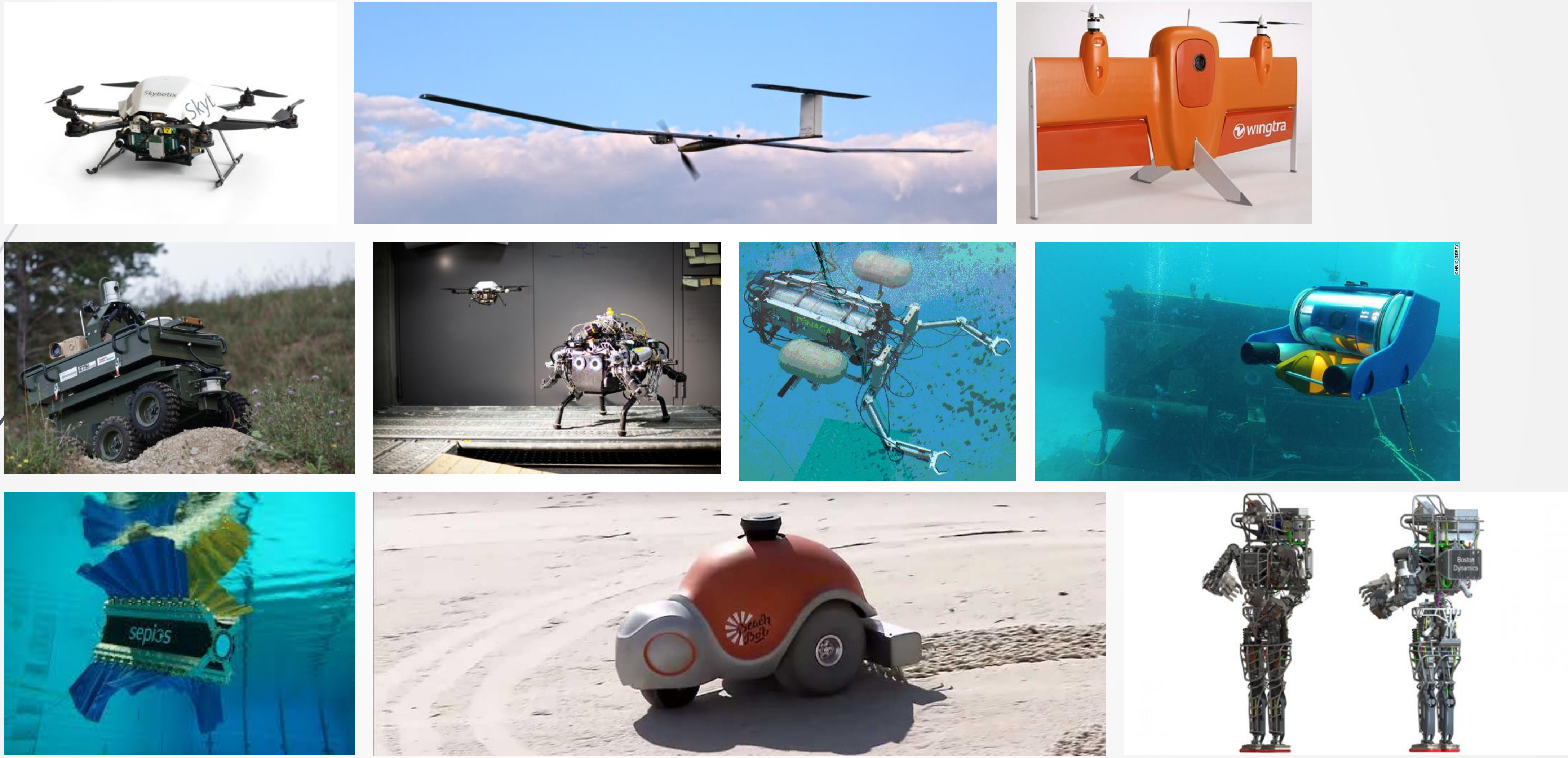
Nadia Comaneci, First “10”, 1976

What should we assume?

- Perfect Sensors?
  - What information
  - Uncertainty
- Perfect Control?
  - What controls?
  - Uncertainty
- Perfect Thinking?
  - Knowledge of the world? Complete?
  - Processing the world? Everything?

What else?

Robots



Trends in Robotics/Motion Planning

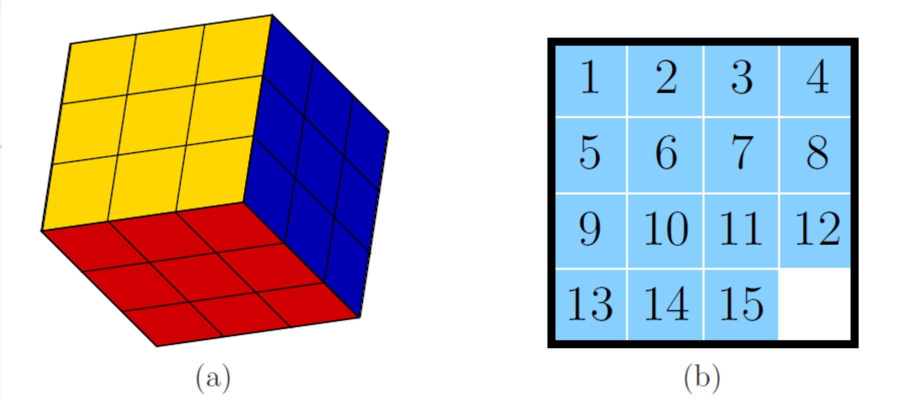
- **Classical Robotics (mid-70's)**
  - Exact models
  - No sensing necessary
- **Reactive Paradigm (mid-80's)**
  - No models
  - Relies heavily on good sensing
- **Hybrids (since 90's)**
  - Model-based at higher levels
  - Reactive at lower levels
- **Probabilistic Robotics (since mid-90's)**
  - Seamless integration of models and sensing
  - Inaccurate models, inaccurate sensors

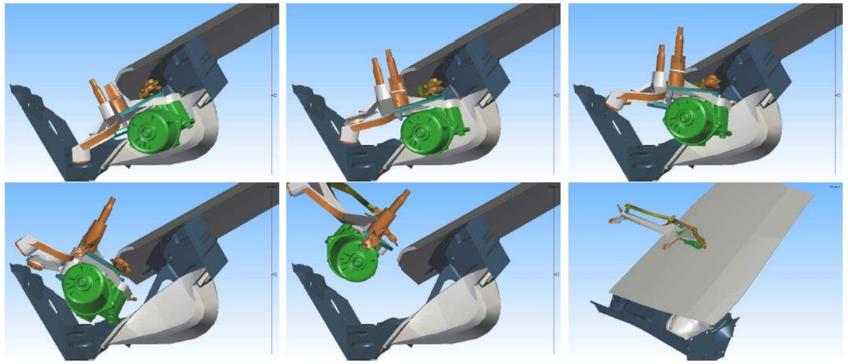
Copied from G. Hager http://voronoi.sbp.ri.cmu.edu/~motion

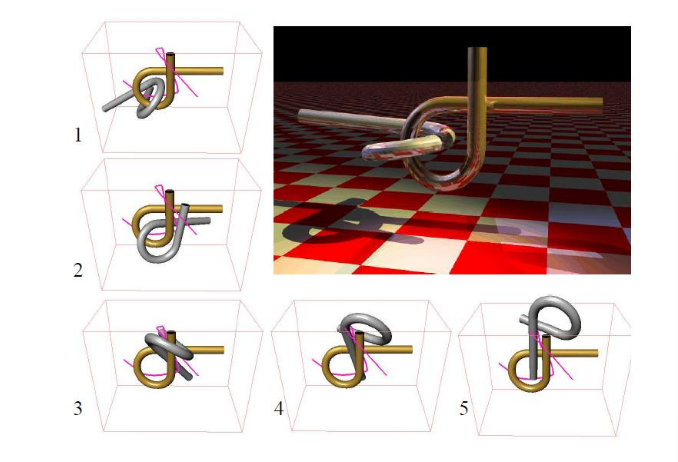
Overview

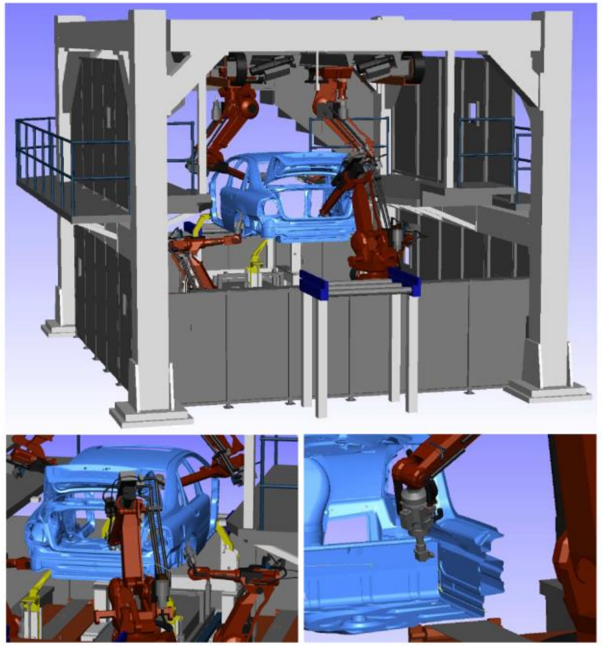
- **Planning tasks**
  - Navigation
  - Coverage
  - Localization
  - Mapping
- **Properties of the robot**
  - Degrees of freedom
  - Holonomic or not
  - Kinematic vs dynamic
- **Properties of Algorithms**
  - Optimality
  - Computational complexity
  - Completeness
- **Properties of Algorithms**
  - Optimality
  - Computational complexity
  - Completeness
    - Resolution completeness
    - Probabilistic completeness
  - Online vs offline
  - Sensor vs offline
  - Sensor-based vs not
  - Feedback or not

Indicative Examples

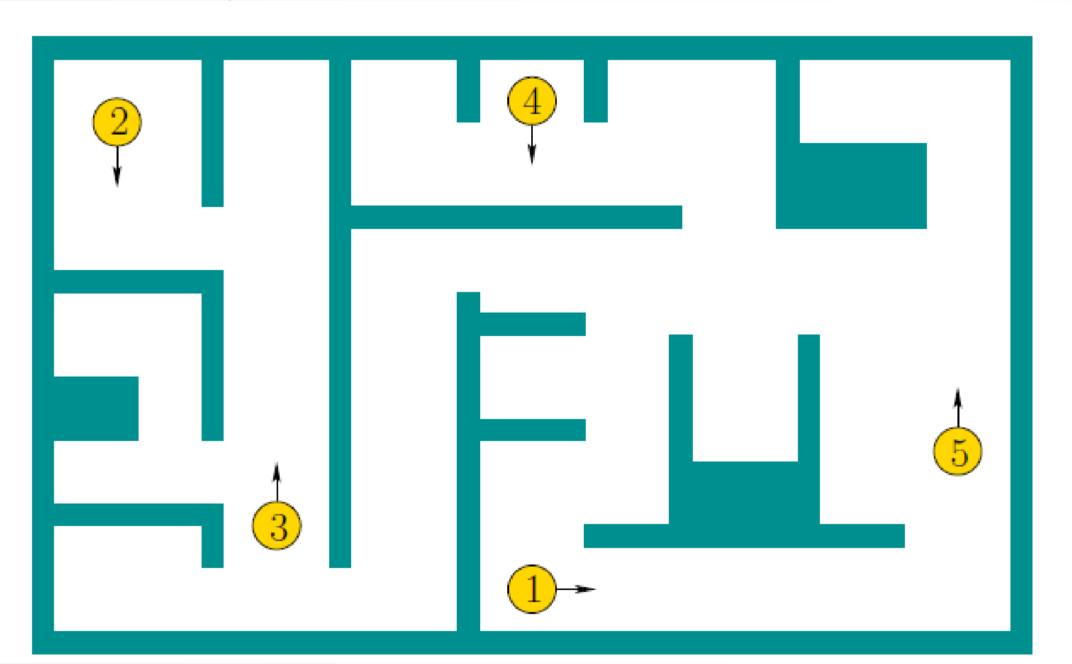






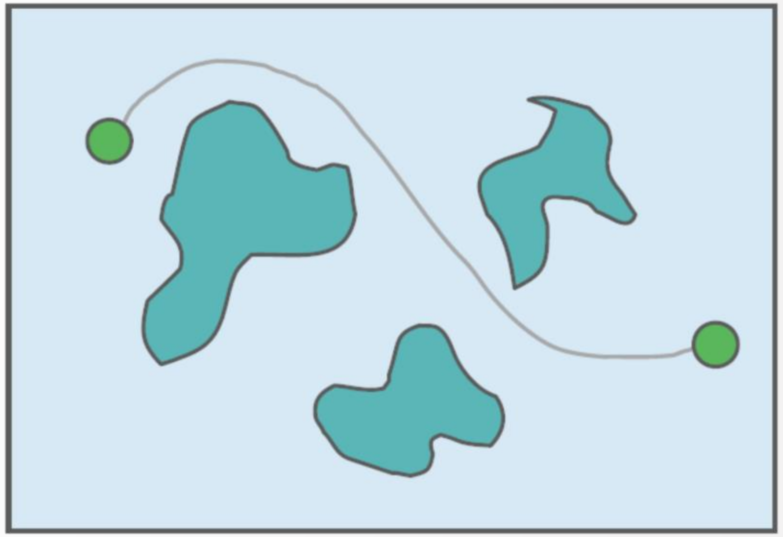


Example of a world (and a robot)



Fundamental Problem of Path Planning

- **Problem Statement:**
  - Compute a continuous sequence of collision-free robot configurations connecting the initial and goal configurations.



- Geometry of the environment
- Geometry and kinematics of the robot
- Initial and goal configurations

→ **Path Planner** →

Collision-free path

Fundamental Problem of Path Planning

- **Problem Statement:**
  - Compute a continuous sequence of collision-free robot configurations connecting the initial andgoal configurations.
- **Motion Planning Statement for collision-free navigation**
  - If W denotes the robot's workspace, and WO_i denotes the i-th obstacle, then the robot's free space, W_{free}, is defined as: $W_{free} = W - (\cup WO_i)$ and a path c is $c: [0,1] \rightarrow W_{free}$, where $c(0)$ is the starting configuration q_{start} and $c(1)$ is the goal configuration q_{goal}.

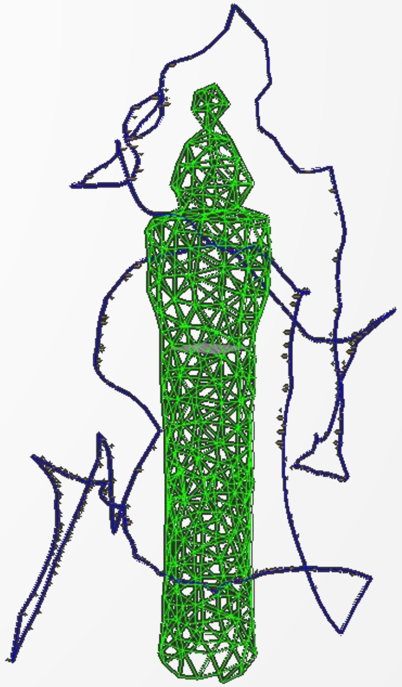

ETH zürich
Autonomous
Systems Lab
Continuous-Time Trajectory Optimization
for Online UAV Replanning
Helen Oleynikova, Michael Burri, Zachary Taylor, Juan Nieto,
Roland Siegwart and Enric Galceran

Coverage Path Planning Problem

- **Problem Statement:**
  - Consider a 3D structure to be inspected and a system with its dynamics and constraints and an integrated sensor, the limitations of which have to be respected. The 3D structure to be inspected is represented with a geometric form and the goal is to calculate a path that provides the set of camera viewpoints that ensure full coverage subject to the constraints of the robot and the environment.

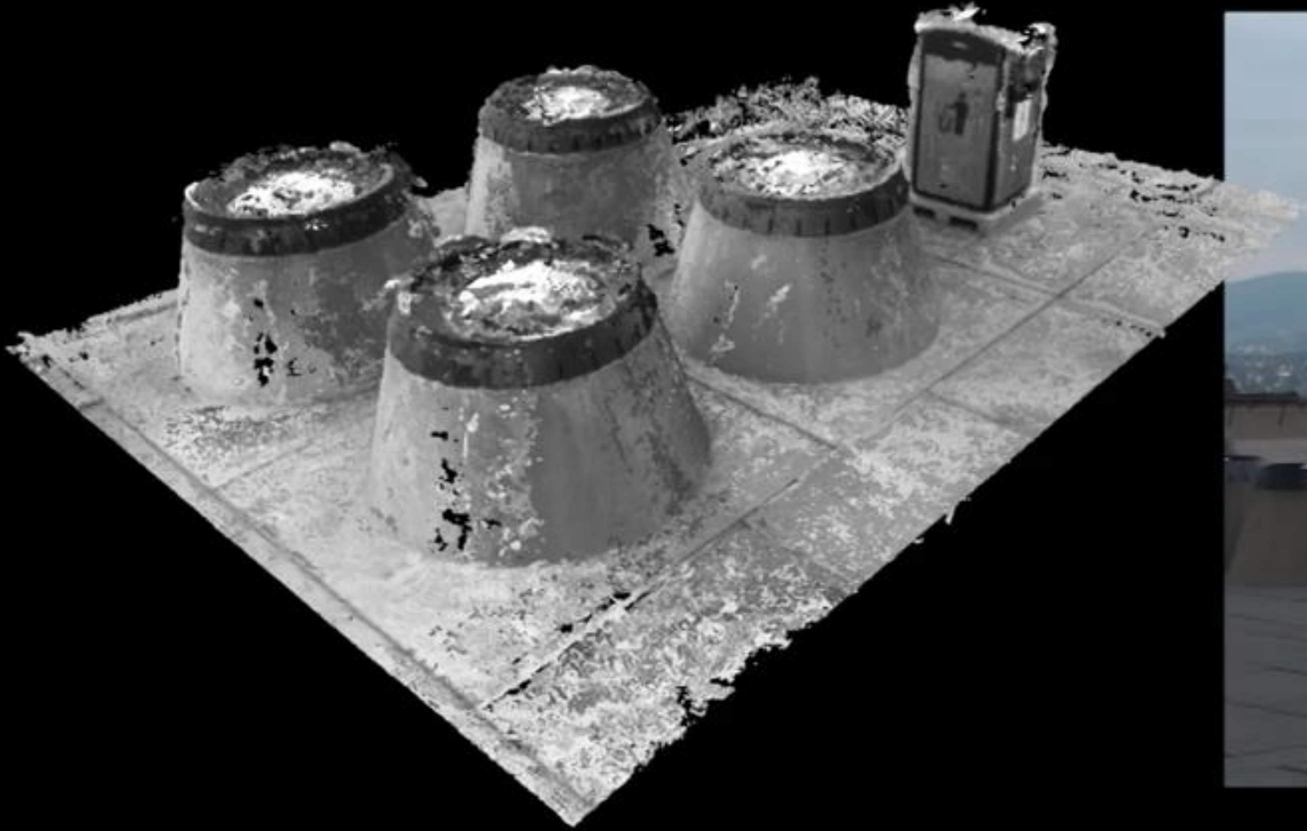
- Geometry of the environment
- Geometry and kinematics of the robot
- Structure to be inspected

→ **Path Planner** → Full coverage path



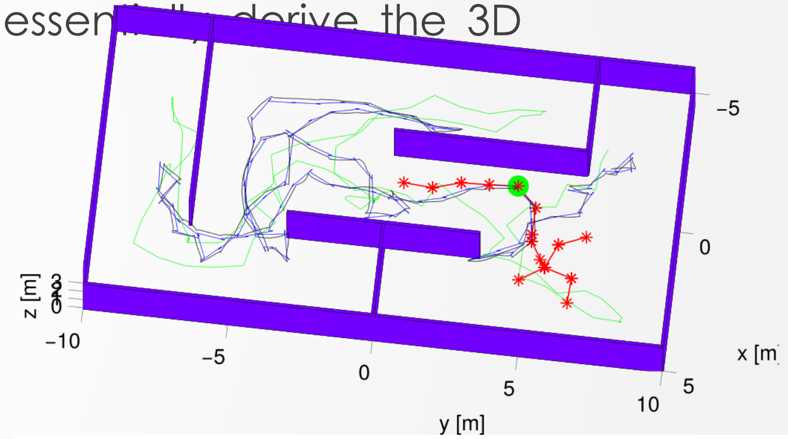
Three-dimensional Coverage Path Planning via Viewpoint Resampling and Tour Optimization using Aerial Robots

A. Bircher, K. Alexis, M. Kamel, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart



Exploration of Unknown Environments

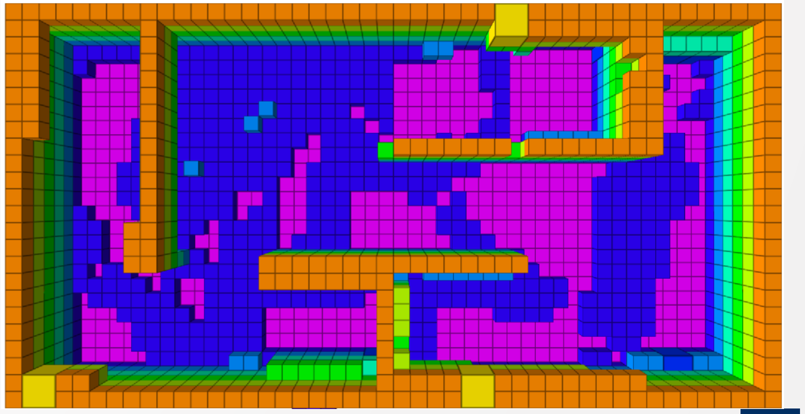
- **Problem Statement:**
  - Consider a 3D bounded space V unknown to the robot. The goal of the autonomous exploration planner is to determine which parts of the initially unmapped space are free V_{free} or occupied V_{occ} and essentially derive the 3D geometric model of the world.



- No knowledge of the environment
- Online 3D Mapping

Path Planner

Efficient exploration per step of execution.



BVS: Holonomic Robot

- Explicit solutions to the problem of point-to-point navigation of a holonomic vehicle operating within an obstacle-free world are straightforward. More specifically, a 6-degrees of freedom (DOF) vehicle that can be approximated to assume only small roll and pitch angles can be approximated using a very simple Boundary Value Solver (BVS). Considering an approximate state vector **ξ = [x,y,z,ψ]** (where x,y,z are the 3 position states and ψ the yaw angle), the path from the state configuration **ξ0** to **ξ1** is given by:

$$\xi(s) = s\xi_1 + (1-s)\xi_0, \ \xi \in [0,1]$$

And considering a limitation on the possible rate of change of the yaw angle \dψ/*dt*|*max* and the maximum linear velocity *υmax*, the execution time is:

$$t_{ex} = \max(d/\upsilon_{max}, \|\psi_1 - \psi_0\|/\dot{\psi}_{\max}$$

with d used to denote the Euclidean distance.

BVS: Holonomic Robot

File: HoverModeMain.py

```
#       __HOVERMODEMAIN__
#       This is the main file to execute examples of the Hover mode
#
#       Authors:
#       Kostas Alexis (kalexis@unr.edu)

from HoverFunctions import *
from PlottingTools import plot3
import numpy as np
import time
import sys

pi = np.pi
verbose_flag = 0
plot_flag = 1

point_0 = np.array([0,0,0,0])
point_1 = np.array([10,10,10,pi/4])

class ExecutionFlags(object):
    """
        Execution flags
    """
    def __init__(self, verbose_flag, plot_flag):
        self.verbose = verbose_flag
        self.plot = plot_flag

class VehicleParameters(object):
    """
        Vehicle Parameters
    """
```

- http://www.kostasalexis.com/holonomic-vehicle-bvs.html

BVS: Nonholonomic Robot

- Dubins airplane is an extension of the classical Dubins car model for the 3D case of an airplane. The specific implementation provided here relies on the formulation presented in:

  (b)Mark Owen, Randal W. Beard and Timothy W. McLain, "Implementing Dubins Airplane Paths on Fixed-Wing UAVs"

  and essentially (as described in this paper) corresponds to a modification of the initial model proposed by *Lavalle et al.* so that it becomes more consistent with the kinematics of a fixed-wing aircraft. Dubins airplane paths are more complicated than Dubins car paths because of the altitude component. Based on the difference between the altitude of the initial and final configurations, Dubins airplane paths can be classified as low, medium, or high altitude gain. While for medium and high altitude gain there are many different Dubins airplane paths, this implementation selects the pat that maximizes the average altitude throughout the maneuver.

BVS: Nonholonomic Robot

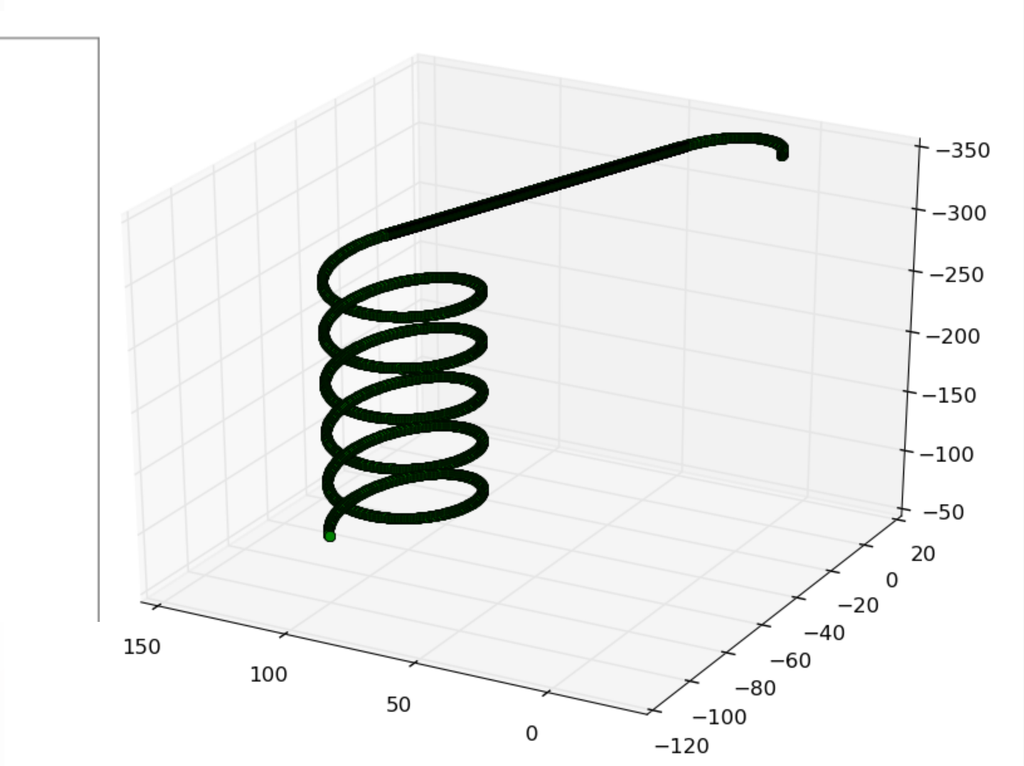
Main File: **DubinsAirplaneMain.py**

```
#       __DUBINSAIRPLANEMAIN__
#       This is the main file to execute examples of the Dubins Airplane mode
#       that supports 16 cases of possible trajectories
#
#       Authors:
#       Kostas Alexis (konstantinos.alexis@mavt.ethz.ch)

from DubinsAirplaneFunctions import *
from PlottingTools import plot3
import numpy as np
import time
import sys

pi = np.pi
dubins_case = 0
verbose_flag = 0
plot_flag = 1

class ExecutionFlags(object):
    """
        Execution flags
    """
    def __init__(self, verbose_flag, plot_flag):
        self.verbose = verbose_flag
```



- http://www.kostasalexis.com/dubins-airplane.html

Open Source Code

- **Open Source Code:**
  - Structural Inspection Planner:
    - https://github.com/ethz-asl/StructuralInspectionPlanner
  - Next-Best-View Planner:
    - https://github.com/ethz-asl/nbvplanner

- **Associated Datasets:**
  - Structural Inspection Planner:
    - https://github.com/ethz-asl/StructuralInspectionPlanner/wiki/Example-Results
  - Next-Best-View Planner:
    - https://github.com/ethz-asl/nbvplanner/wiki/Example-Results
  - Solar-powered UAV Sensing & Mapping:
    - http://projects.asl.ethz.ch/datasets/doku.php?id=fsr2015

What do I need to know? How do I start?

- **Motion Planning algorithms**
  - Deep understanding of motion planning algorithms.
- **Robot Dynamics**
  - Capability to model the robot dynamics and solve state-space sampling problems (require boundary value solvers) or control-space sampling.
- **Robot Localization & Mapping**
  - At the very minimum, understand Simultaneous Localization & Mapping as well as sensor modeling.

- **How do I start?**
  - Use the Open-Sourced code!
  - Use RotorS: https://github.com/ethz-asl/rotors_simulator
  - Learn ROS and one of the C++/Python
  - Contact us

Find out more

- http://www.kostasalexis.com/autonomous-navigation-and-exploration.html
- http://www.kostasalexis.com/holonomic-vehicle-bvs.html
- http://www.kostasalexis.com/dubins-airplane.html
- http://www.kostasalexis.com/collision-free-navigation.html
- http://www.kostasalexis.com/structural-inspection-path-planning.html

- http://ompl.kavrakilab.org/
- http://moveit.ros.org/
- http://planning.cs.uiuc.edu/

References

- A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, "Structural Inspection Path Planning via Iterative Viewpoint Resampling with Application to Aerial Robotics", IEEE International Conference on Robotics & Automation, May 26-30, 2015 (ICRA 2015), Seattle, Washington, USA
- Kostas Alexis, Christos Papachristos, Roland Siegwart, Anthony Tzes, "Uniform Coverage Structural Inspection Path-Planning for Micro Aerial Vehicles", Multiconference on Systems and Control (MSC), 2015, Novotel Sydney Manly Pacific, Sydney Australia. 21-23 September, 2015
- K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: planning and control", Autonomous Robots, Springer US, DOI: 10.1007/s10514-015-9485-5, ISSN: 0929-5593, http://dx.doi.org/10.1007/s10514-015-9485-5
- A. Bircher, K. Alexis, U. Schwesinger, S. Omari, M. Burri and R. Siegwart "An Incremental Sampling–based approach to Inspection Planning: the Rapidly–exploring Random Tree Of Trees", accepted at the Robotica Journal (awaiting publication)
- A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, "Three-dimensional Coverage Path Planning via Viewpoint Resampling and Tour Optimization for Aerial Robots", Autonomous Robots, Springer US, DOI: 10.1007/s10514-015-9517-1, ISSN: 1573-7527
- A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, "Receding Horizon "Next-Best-View" Planner for 3D Exploration", IEEE International Conference on Robotics and Automation 2016 (ICRA 2016), Stockholm, Sweden (Accepted - to be presented)

Thank you!

Please ask your question!