EE5110/6110 Special Topics in Automation and Control

Introduction to Unmanned Aerial Vehicles

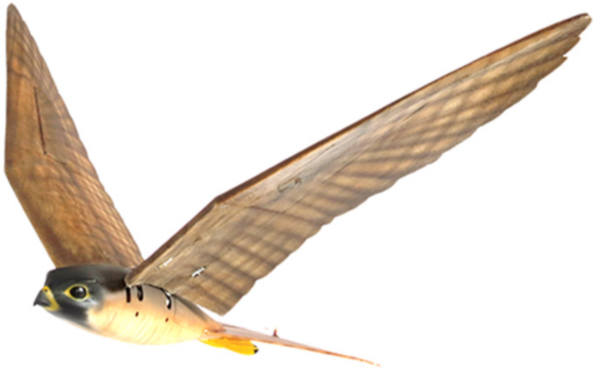
Ben M. Chen

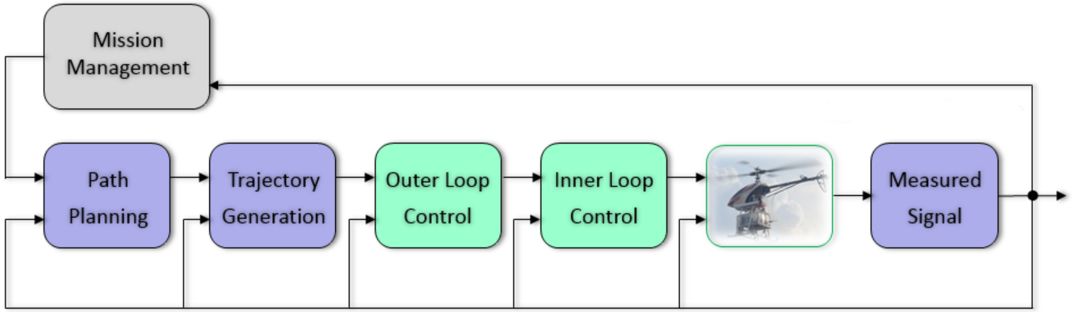
Professor & Provost's Chair
Department of Electrical & Computer Engineering

Office: E4-06-08, Phone: 6516-2289
Email: bmchen@nus.edu.sg   http://www.bmchen.net

Outline of the notes…

- Introduction to drones and applications
  - Types of drones
  - Applications
  - NUS UAV research highlights
- Overall structure of unmanned systems
  - Communication unit
  - Ground control systems
- Internal structure of unmanned systems
  - Avionic systems
  - Dynamic modeling
  - Flight control systems
  - Measurement systems
  - Path planning
  - Trajectory generation
  - Mission management





Continuous assessments…

EE5110

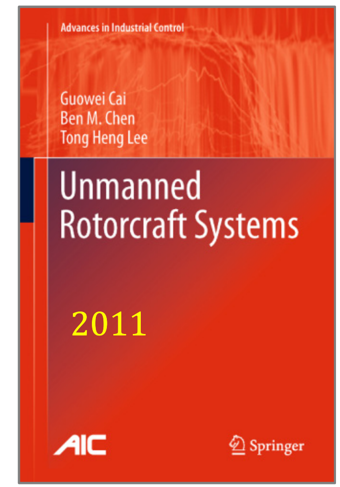
- EE5110 students are required to propose a technical solution for small-scale UAVs to navigate in a GPS-denied unknown environments, such as indoor and urban canyon. Considerations include the constrains of the UAV platforms, such as maximum payload, onboard computation resources, power consumption and communication bandwidth. The specification of the UAV will be provided. A technical report is expected to explain the solution. It should include the literature review, basic ideas, proof of concept and discussion.

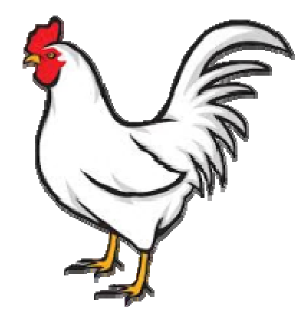
EE6110

- EE6110 students are required to finish a project to simulate navigation and control of a UAV in an indoor environment without GPS by using MATLAB or a robotics related simulation system, such as ROS. The dynamics model of the UAV will be provided in MATLAB and C++ together with the required indoor environment. The students can build the environment and place artificial markers in the environment by themselves.

What is a drone or UAV?

An unmanned aerial vehicle (UAV or drone) is an aircraft that is equipped with necessary data processing units, sensors, automatic control, and communications systems and is capable of performing autonomous flight missions without the interference of a human pilot.





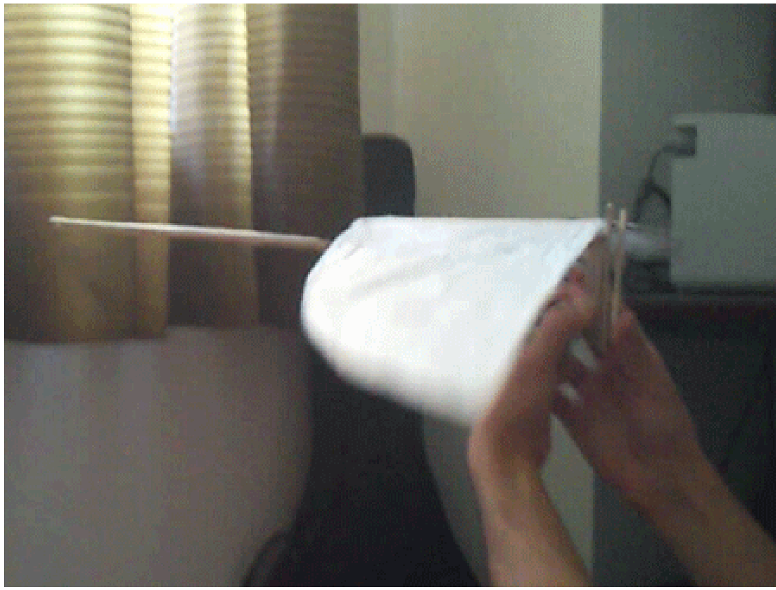
A drone?

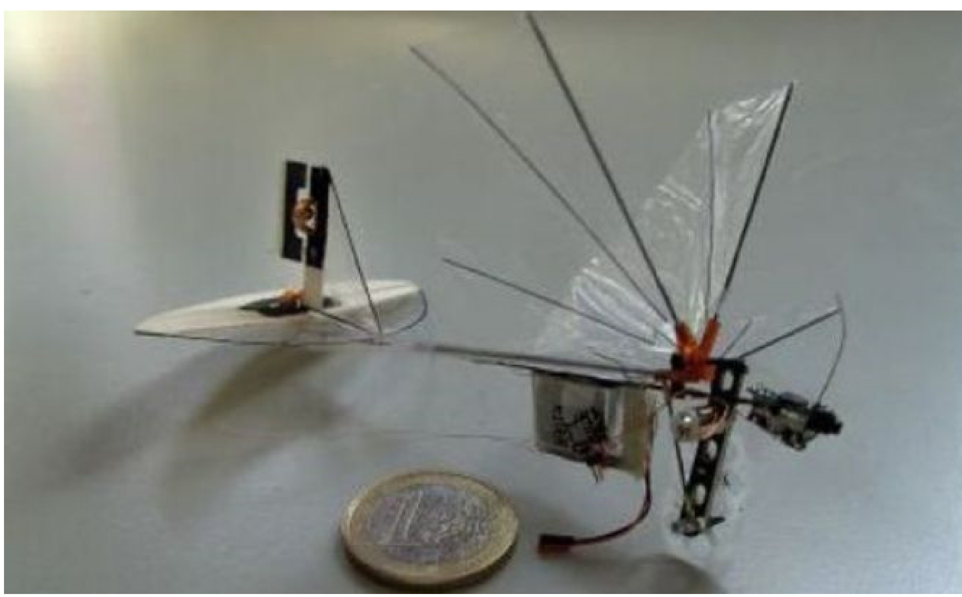
Drone types…

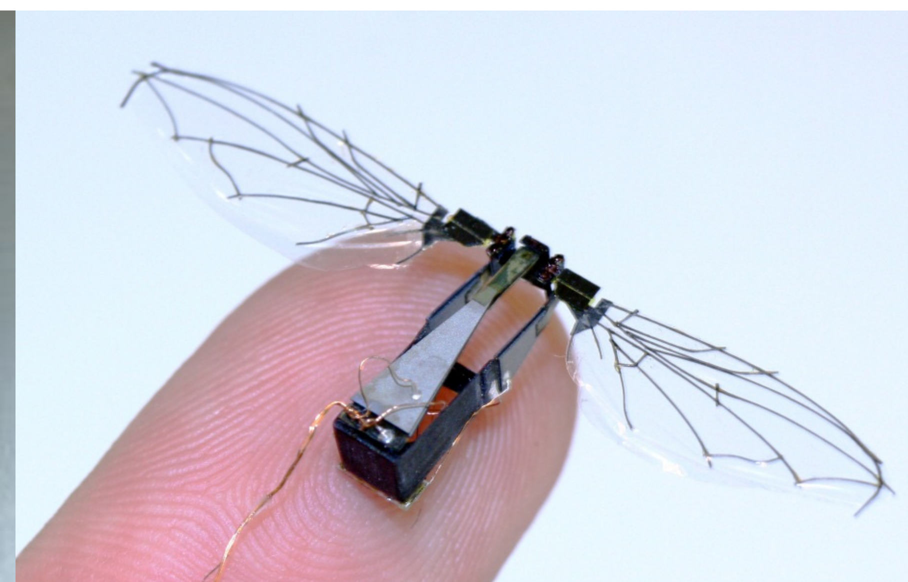
Fixed-wing UAVs…

Drone types...

Flapping-wing UAVs...







Drone types…

Rotorcraft UAVs…




Drone types…

Hybrid UAVs…

Drone applications…

GPS-based environments…

Search & rescue


Surveillance & patrolling


Forestation & agriculture


Aerial mapping

Drone applications…

GPS-based environments…

Drone applications…

GPS-denied environments…

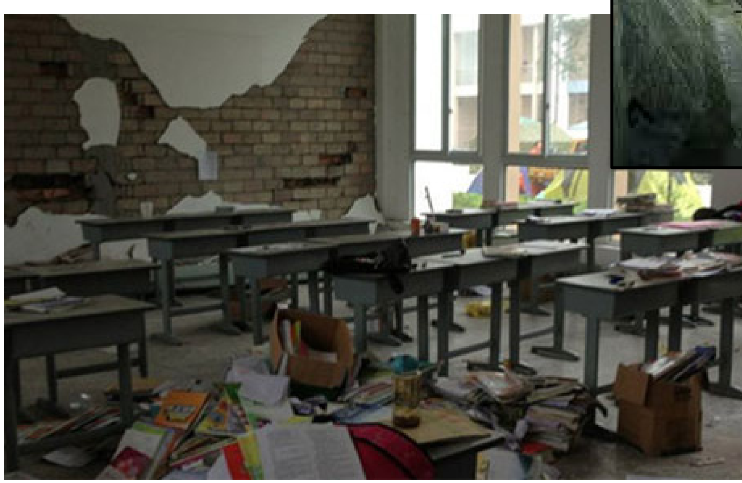
Urban canyon

&

inside forest

Indoor

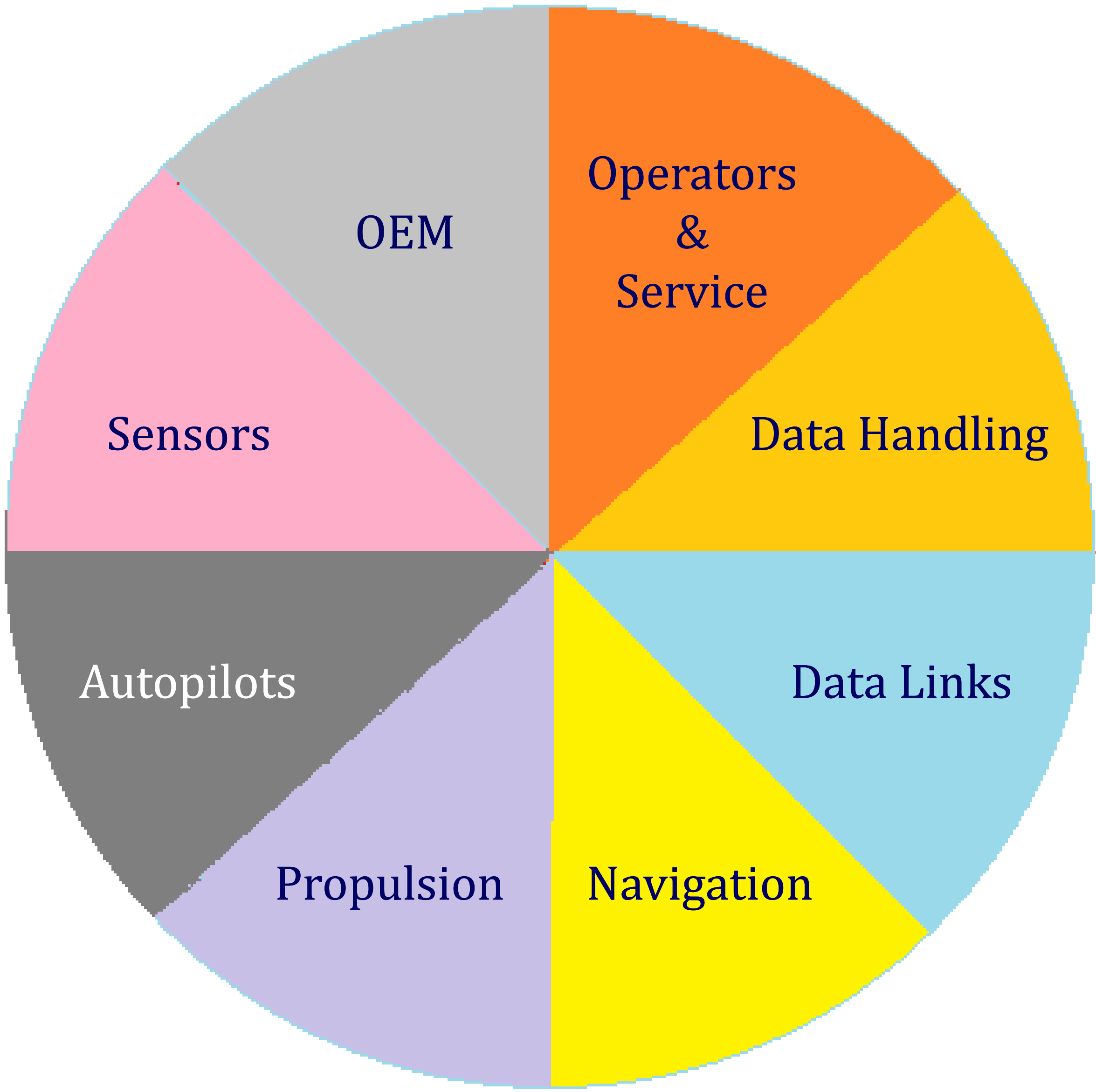
Tunnels



Under bridges

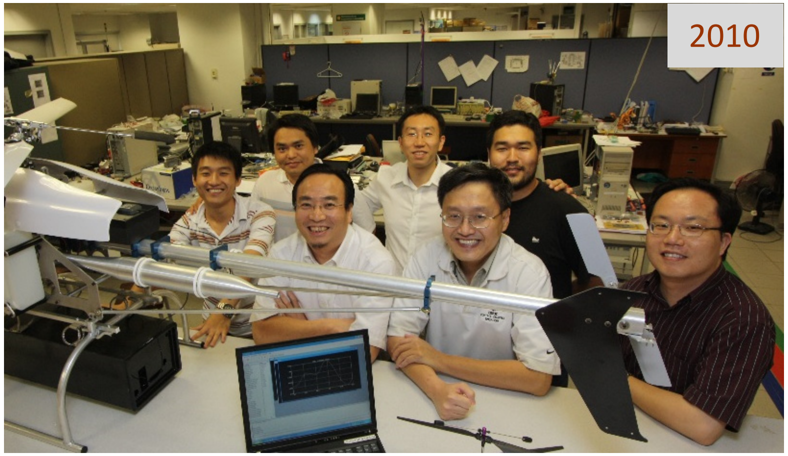
Drone industry…

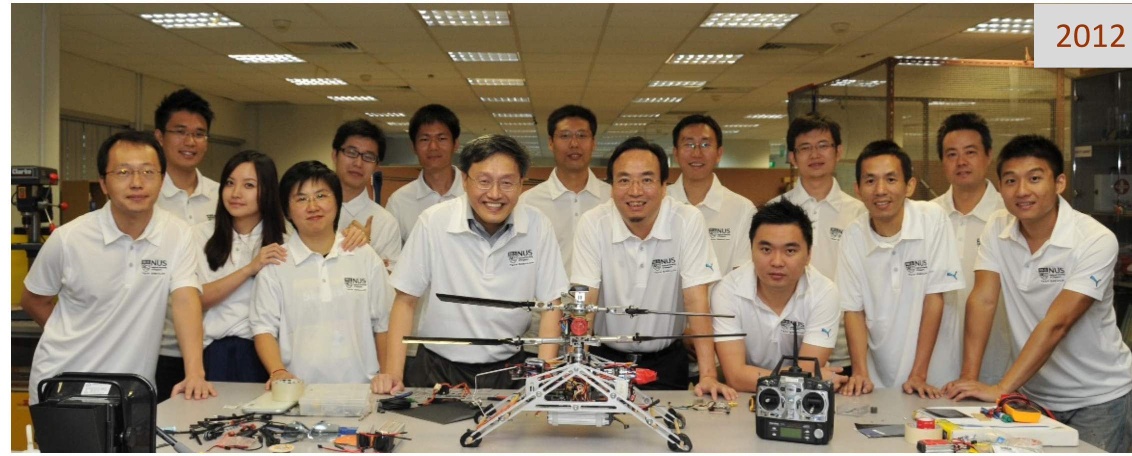
Market sectors…



- Military
- Research/education
- Mining
- Forestry
- Power & pipeline
- Media
- Real estate
- Agriculture
- Policing
- Film industry
- Oil & gas industry
- Ports
- Entertainment
- ……

NUS research team & unmanned systems platforms…





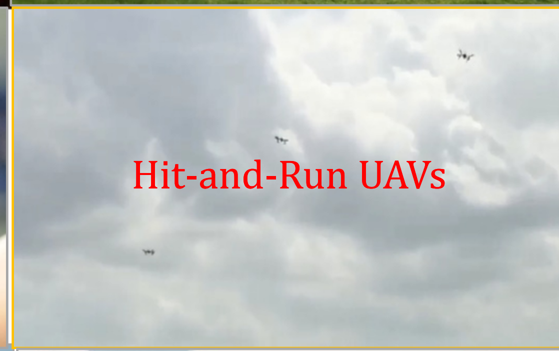
NUS unmanned systems research highlights…

NUS National University of Singapore

Indoor Navigation & Control

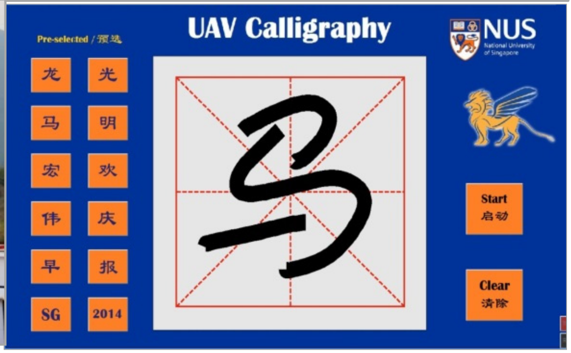
Firefighting

Inside Forest Navigation



Hybrid UAVs

Cargo Transportation



UAV Calligraphy

Awards won by NUS research team…

2003


DARPA UAVForge Finalist 2012


WCICA Best Paper Award 2010

2015


2015 IMAV 2015 2nd Place


2014 CCC Guan Zhaozhi Prize

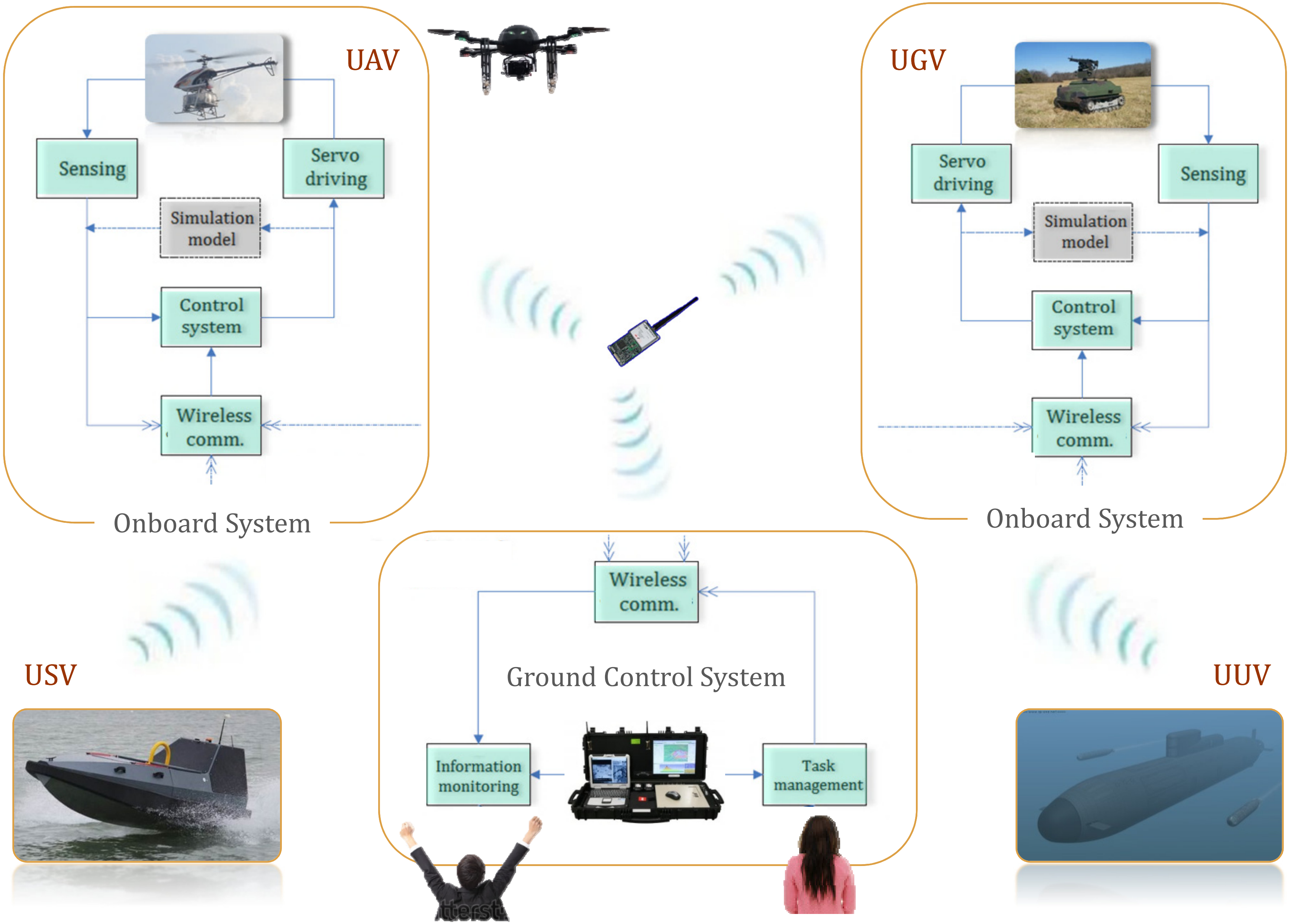
IMAV 2014 Champion


2011

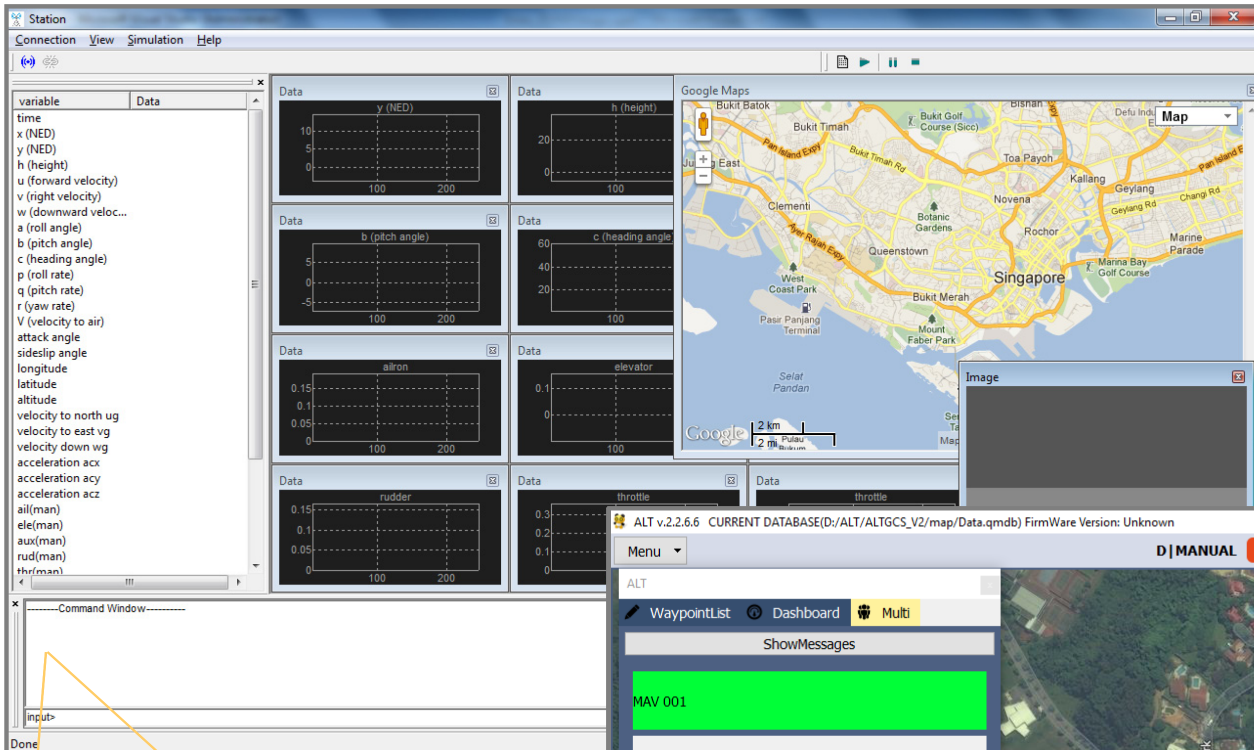
2013

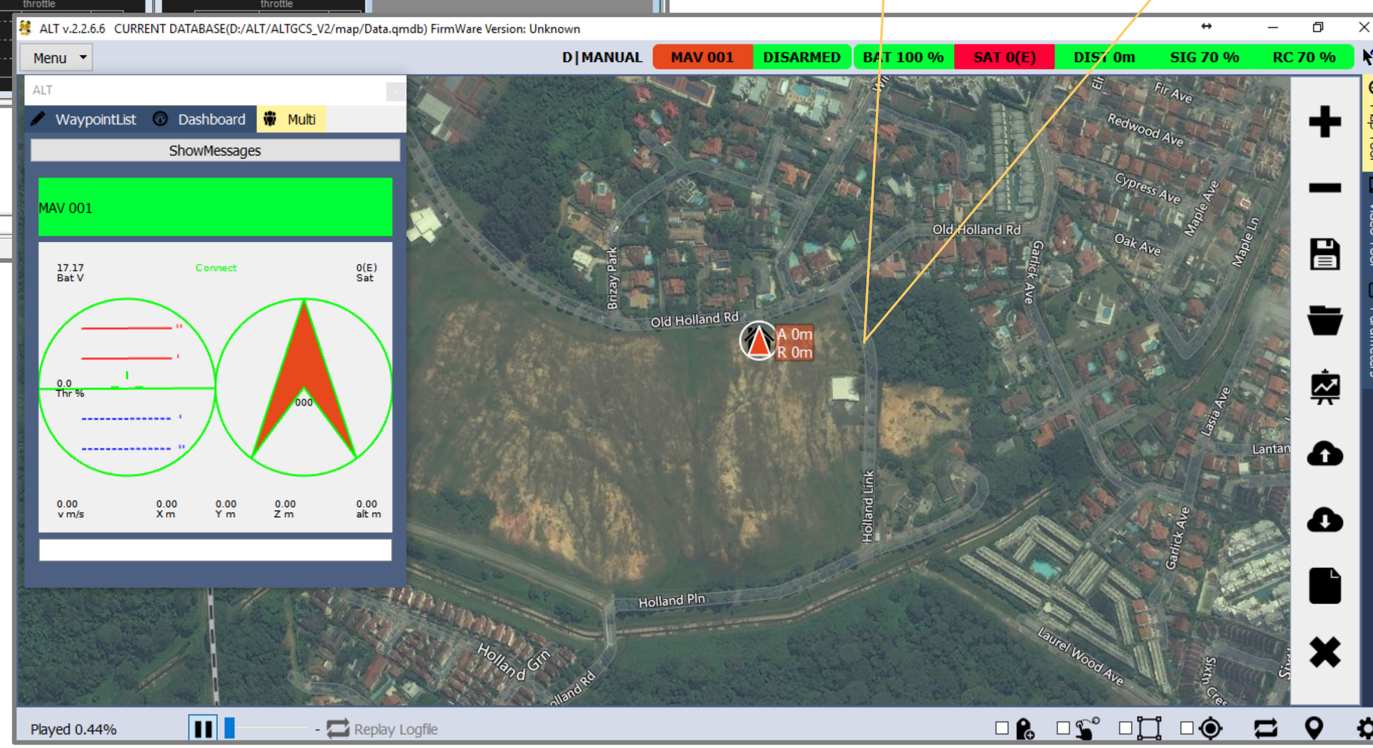

2014

Overall structure of unmanned systems…



Ground control systems...

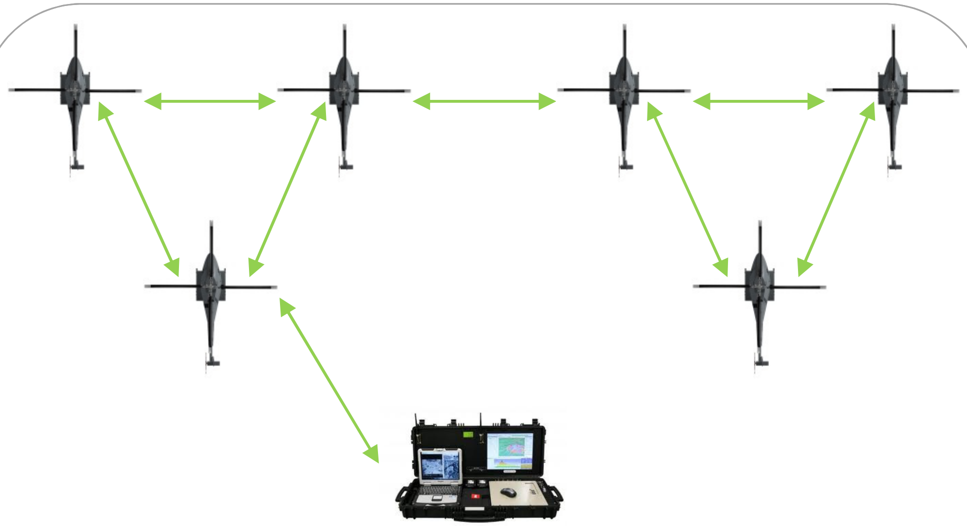




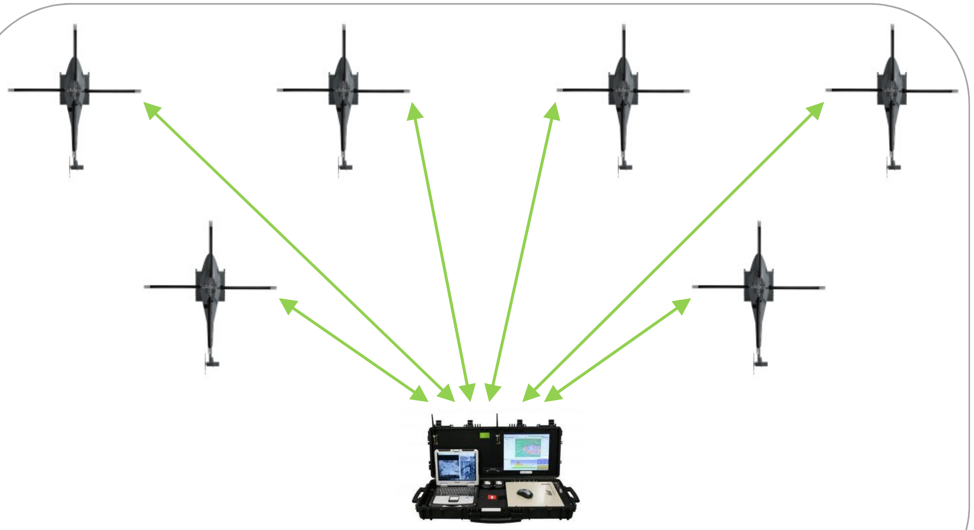
Online touch-screen path planning on the map

Command window to issue real-time commands

The communication units in the UAV system framework are deployed as interfaces between the UAV entity itself and external entities. The external entity can be the GCS for the ground operator, or another UAV entity for information exchange. With UAV to GCS communications, the operator can remotely control and monitor UAVs in operation. With inter-UAV communications, the UAV team can multiply their capability and effectiveness in cooperative tasks.

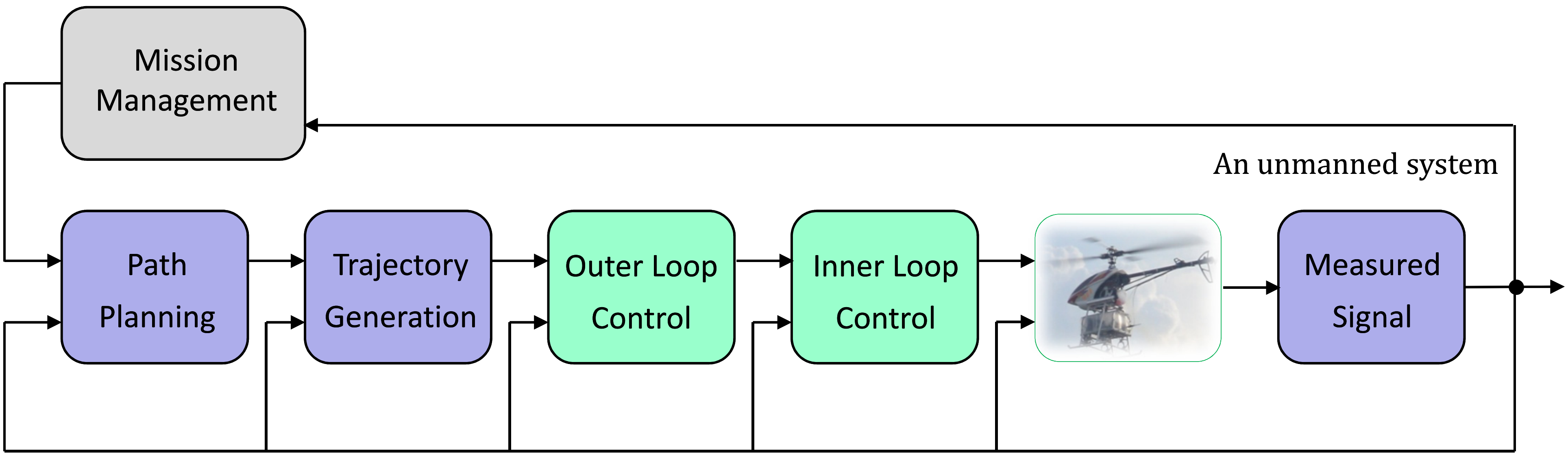


Decentralized Control



Centralized Control

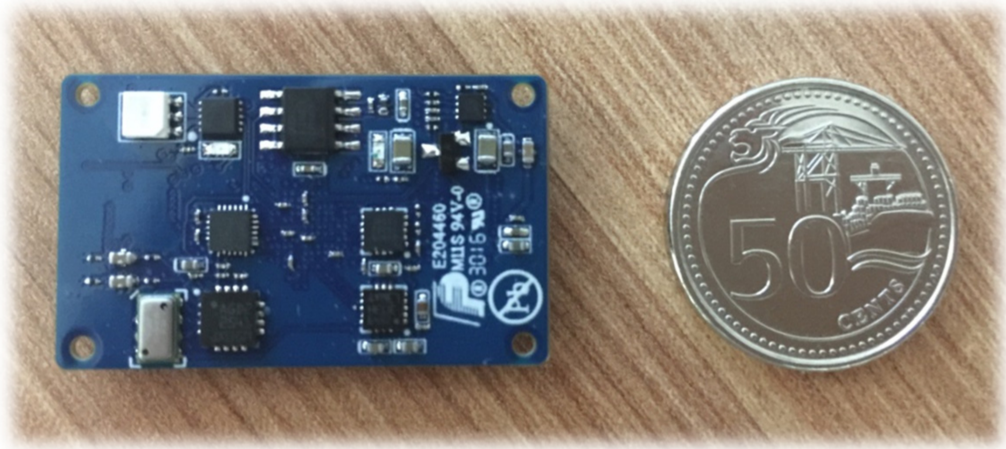
Internal structure of unmanned systems…

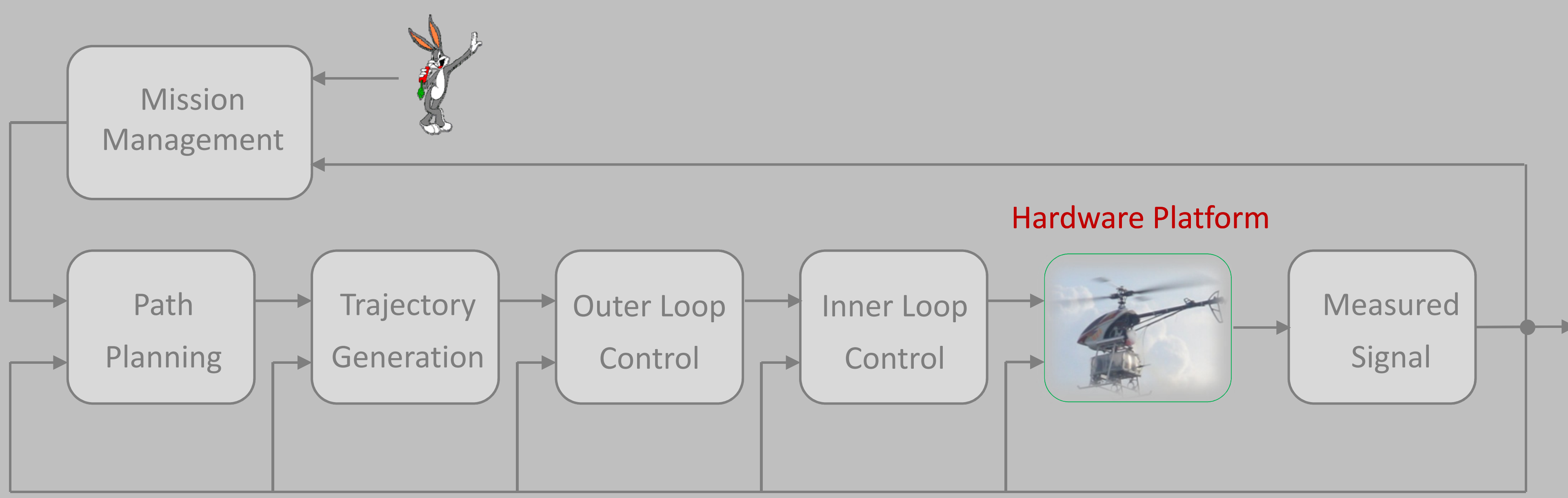


Hardware components & avionic systems…

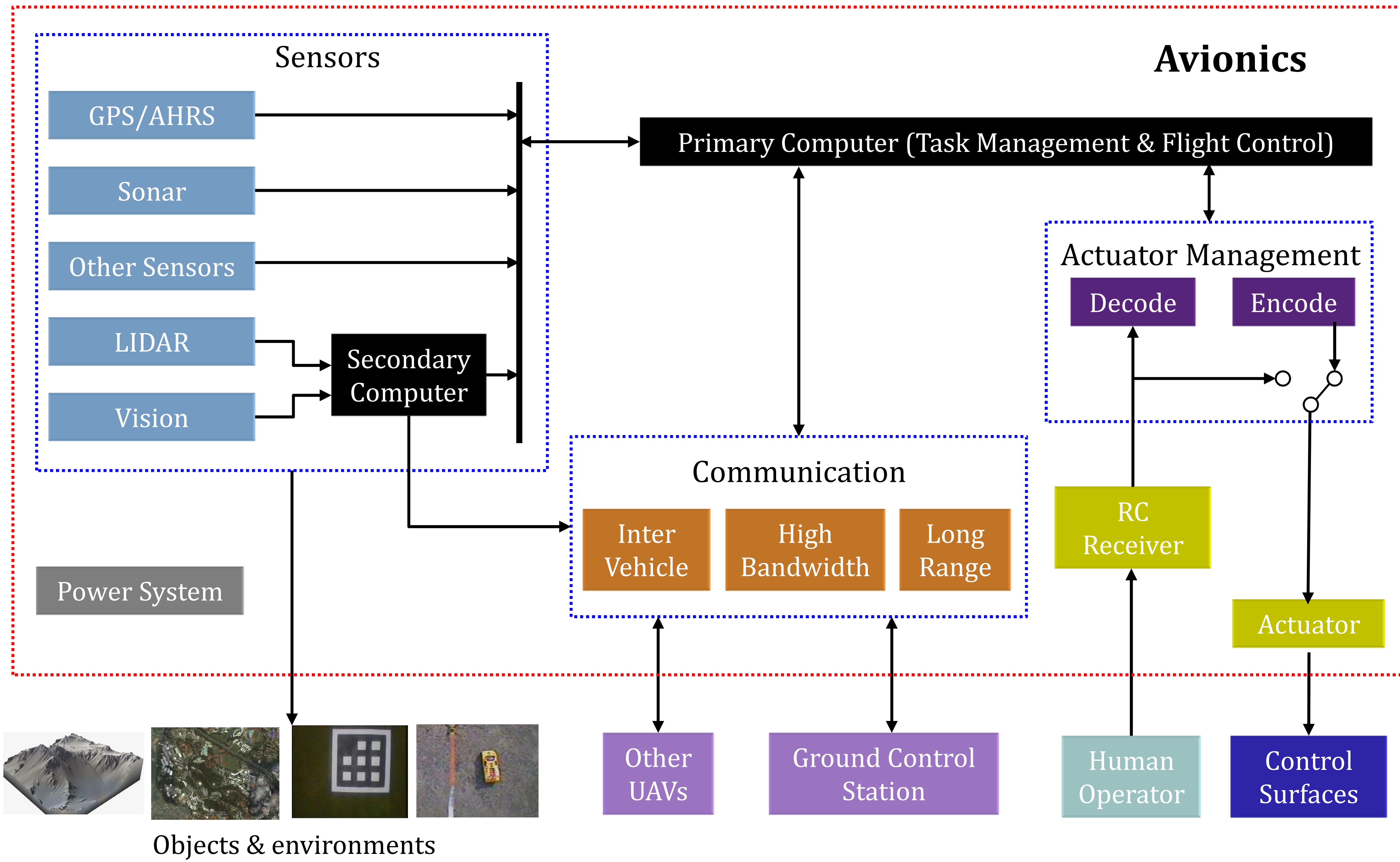
An NUS made micro quadrotor

An NUS made avionic system

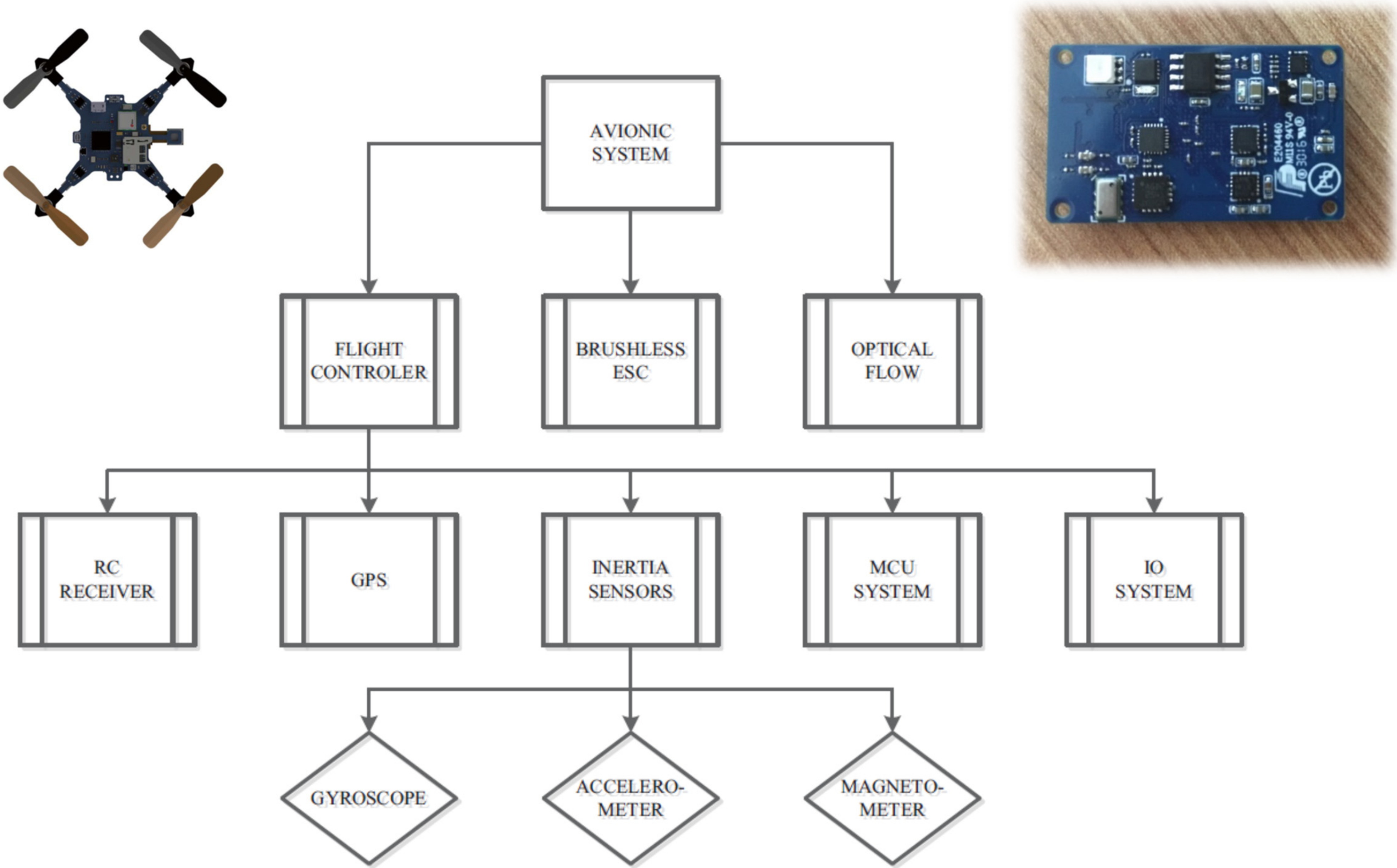




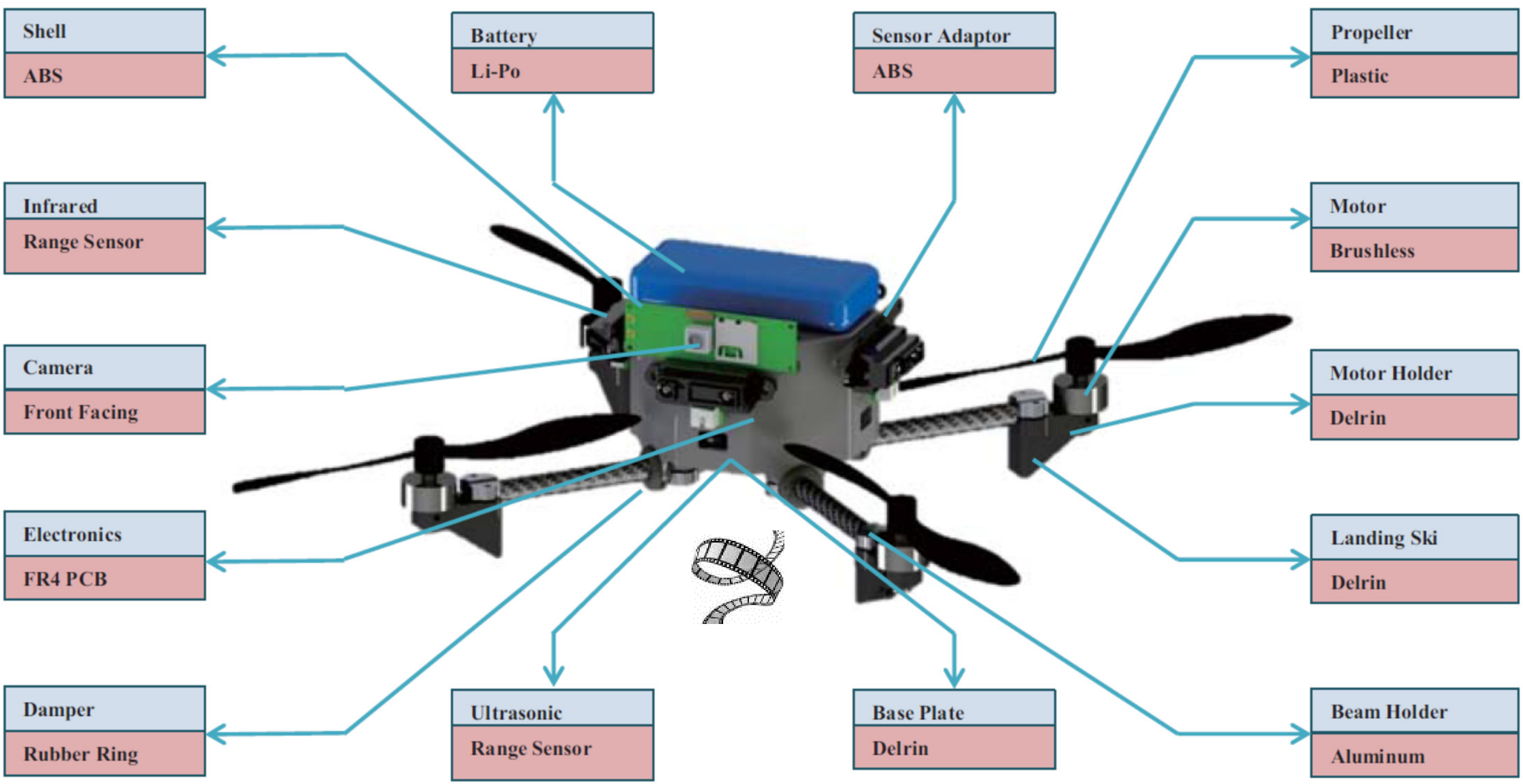
Essential hardware components of unmanned systems…



Hierarchy of an avionic system…



Component breakdown of a typical multi-rotor UAV…

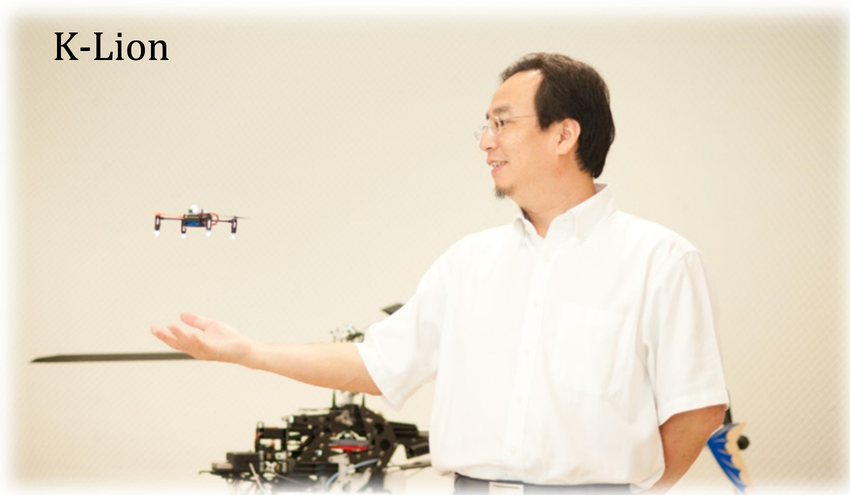


Multi-rotor UAV platforms…

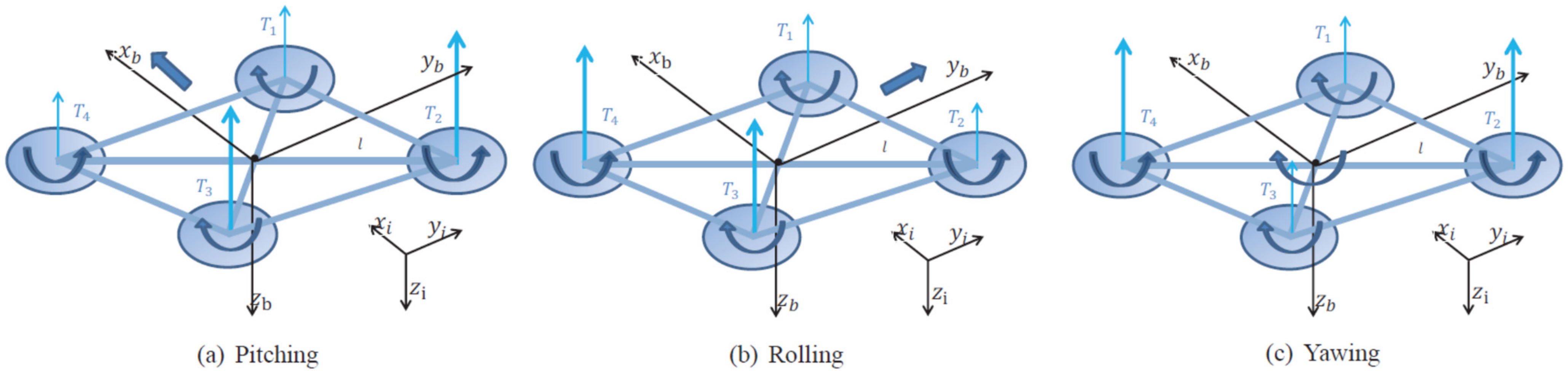
BlackLion 168

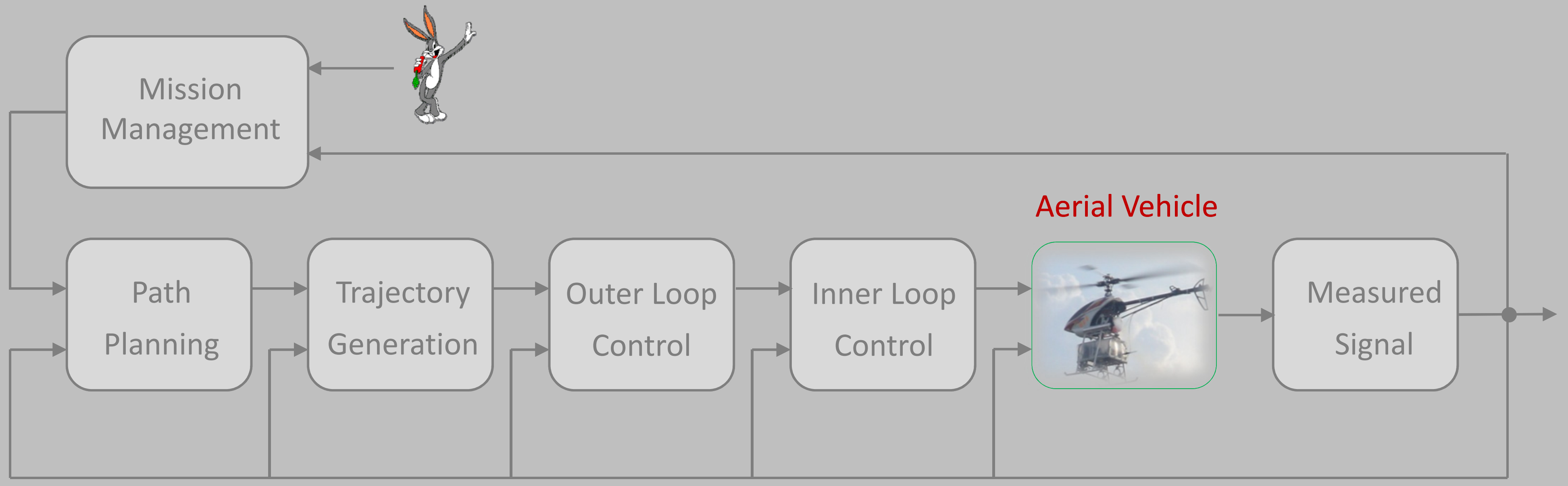
All Made in Singapore

BlackLion 068

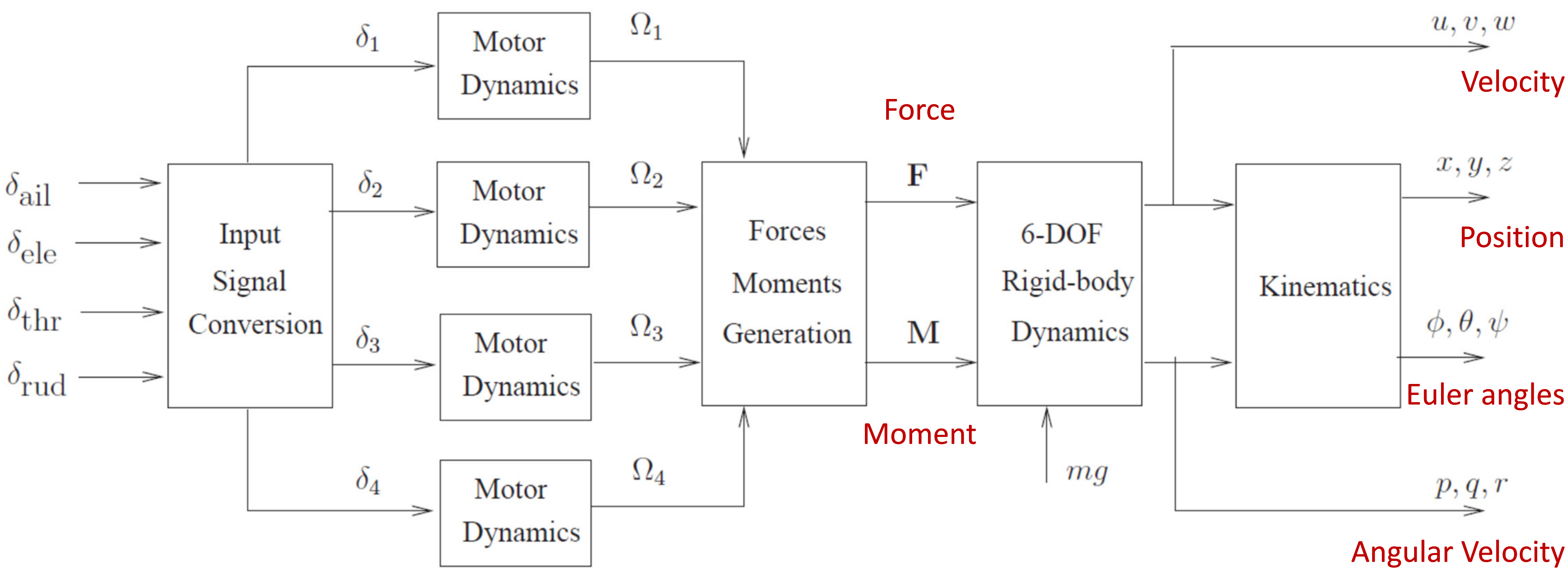


Dynamics modeling of a quadrotor UAV...

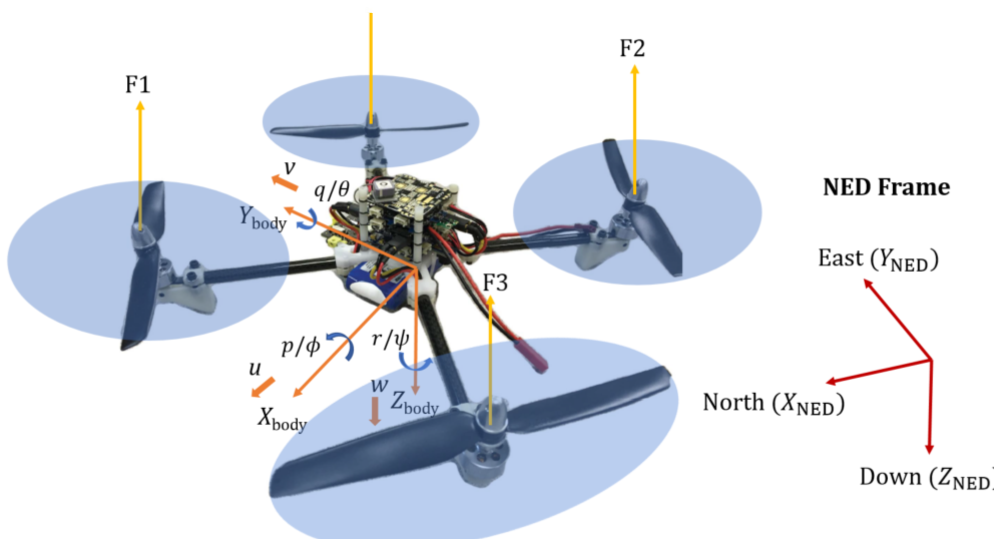




Dynamics model structure of the aerial vehicle…



- δ_{ail} aileron input
- δ_{ele} elevator input
- δ_{thr} collective pitch input
- δ_{rud} rudder input



Linearized model of the aerial vehicle…

With some appropriate simplifications, we can obtain a simplified linear model for a quadrotor UAV as follows

$$
\begin{aligned}
\dot{u} &= -g\theta \\
\dot{v} &= g\phi \\
\dot{w} &= \frac{1}{m}u_1 \\
\dot{p} &= J_{\mathrm{XX}}^{-1}u_2 \\
\dot{q} &= J_{\mathrm{YY}}^{-1}u_3 \\
\dot{r} &= J_{\mathrm{ZZ}}^{-1}u_4 \\
\dot{\phi} &= p \\
\dot{\theta} &= q \\
\dot{\psi} &= r
\end{aligned}
$$

$$
\begin{aligned}
u_1 &= F_\Sigma \\
u_2 &= M_1 \\
u_3 &= M_2 \\
u_4 &= M_3
\end{aligned}
$$

$$
\begin{pmatrix} F_\Sigma \\ M_1 \\ M_2 \\ M_3 \end{pmatrix} = \underbrace{\begin{pmatrix} -k_T & -k_T & -k_T & -k_T \\ dk_T & -dk_T & -dk_T & dk_T \\ dk_T & dk_T & -dk_T & -dk_T \\ k_Q & -k_Q & k_Q & -k_Q \end{pmatrix}}_{\Gamma} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix}
$$

ω_1, ω_2, ω_3, ω_4 are respectively the motor angular speeds and Γ is a constant matrix related to motor properties.

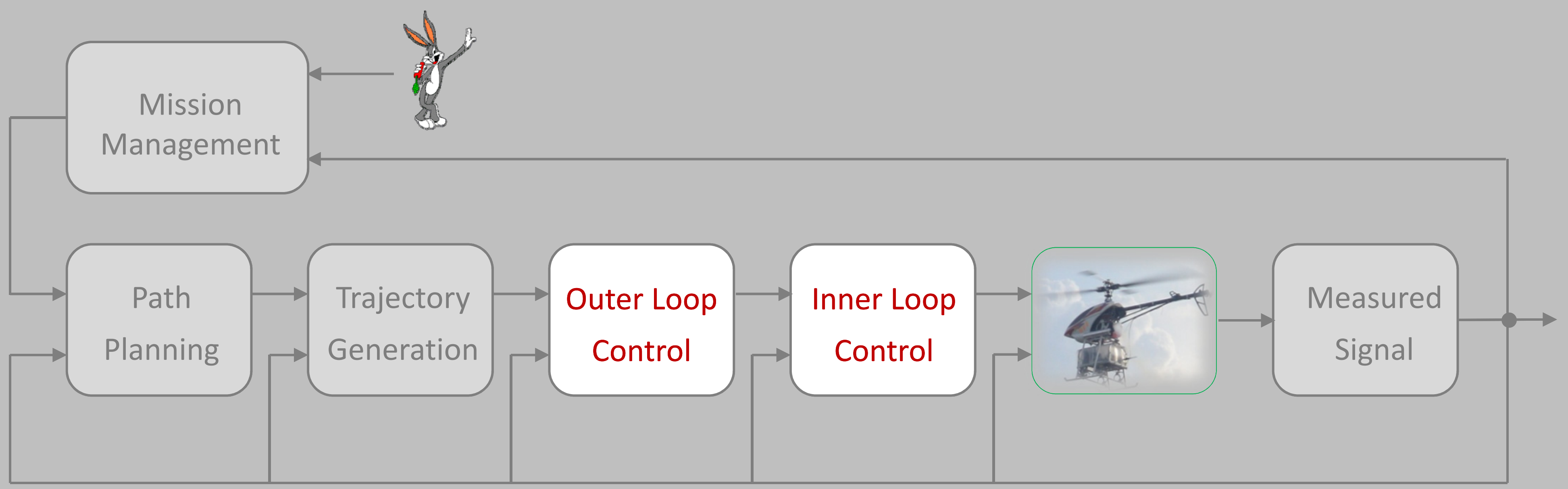
g is the gravity constant, m is the UAV mass.

J_{XX}	$[\mathrm{kgm}^2]$	Inertia moment of body about x-axis
J_{YY}	$[\mathrm{kgm}^2]$	Inertia moment of body about y-axis
J_{ZZ}	$[\mathrm{kgm}^2]$	Inertia moment of body about z-axis

Flight control systems…

Due to the nature of the time scale of the aircraft dynamics, i.e., the attitude response is much faster compared to that of the position and velocity, it is a common practice to separate the flight control system into two parts, i.e., the inner-loop and the outer-loop control. More specifically,

- Inner loop control system is to guarantee the stability of the aircraft attitude; and
- Outer loop control system is to control the aircraft position and velocity.



A brief view on the development of control techniques…

- **Classical control**

  PID control, developed in 1940s and used heavily for in industrial processes.

- **Optimal control**

  Linear quadratic regulator control, Kalman filter, H_2 control, developed in 1960s to achieve certain optimal performance.

- **Robust control**

  H_∞ control, developed in 1980s & 90s to handle systems with uncertainties and disturbances and with high performances.

- **Nonlinear control**

  Still on-going research topics, developed to handle nonlinear systems with high performances.

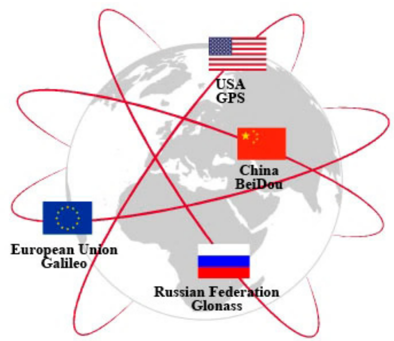
- **Intelligent control**

  Knowledge-based control, adaptive control, neural and fuzzy control, etc., developed to handle systems with unknown models.

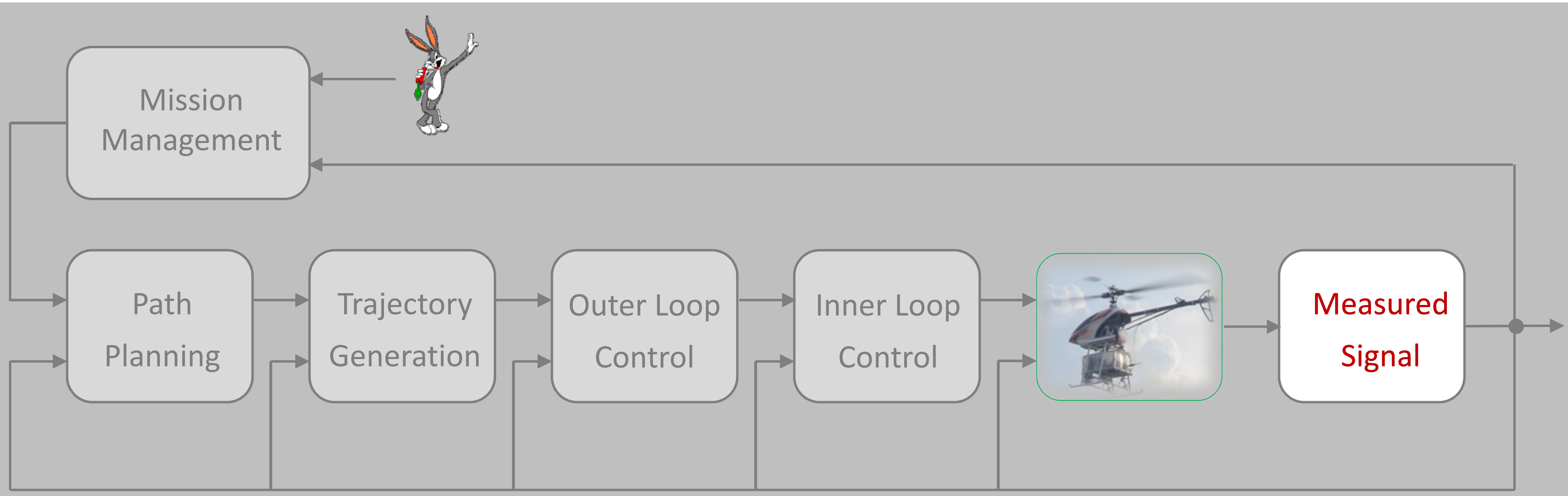
Measurement signals…

Measurement comes from inertial sensors (gyroscope, accelerometer, magnetometer) and

- External positioning systems **or**
  - GPS
  - Beidou
  - VICON
  - UWB



- Internal sensor systems
  - Vision
  - LiDAR
  - Radar
  - Sonar

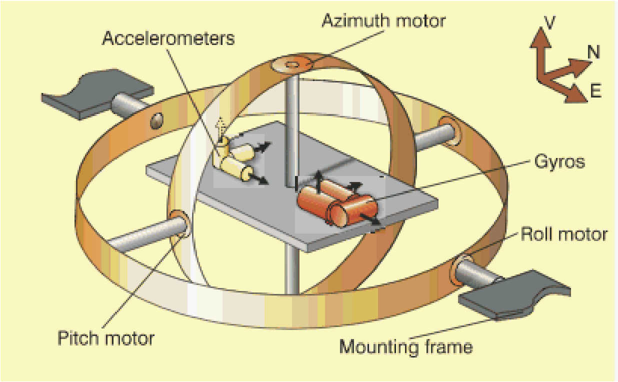


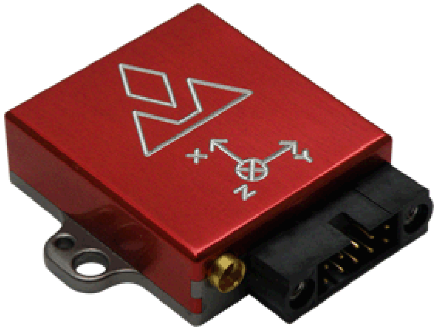
Inertial navigation system (INS)

An **inertial navigation system** (INS) is a navigation aid that uses a computer, motion sensors (accelerometers) and rotation sensors (gyroscopes) to continuously calculate via dead reckoning the position, orientation, and velocity (direction and speed of movement) of a moving object without the need for external references. It is used on vehicles such as ships, aircraft, submarines, guided missiles, and spacecraft. Other terms used to refer to inertial navigation systems or closely related devices include inertial guidance system, inertial instrument, **inertial measurement unit** (IMU).

An INS is capable of providing the following information of unmanned vehicles:

- Accelerations
- Velocities
- Rotating angles
- Heading angles

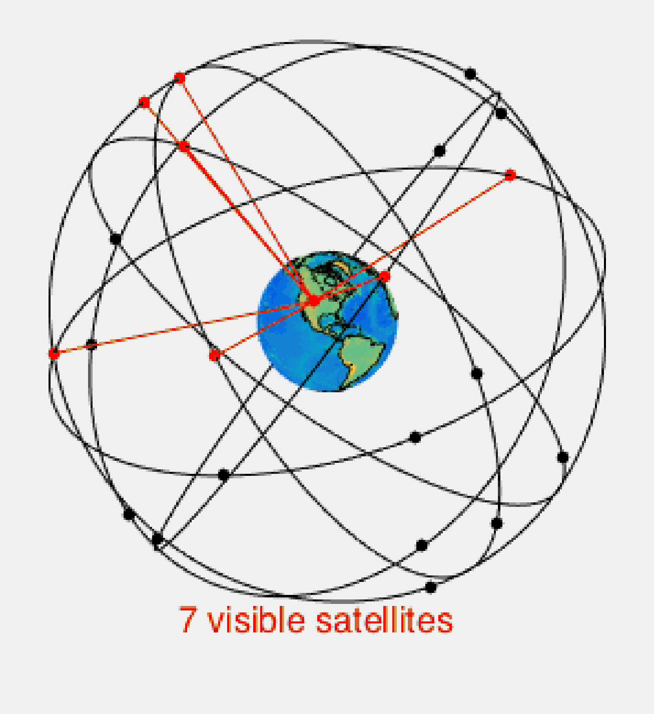




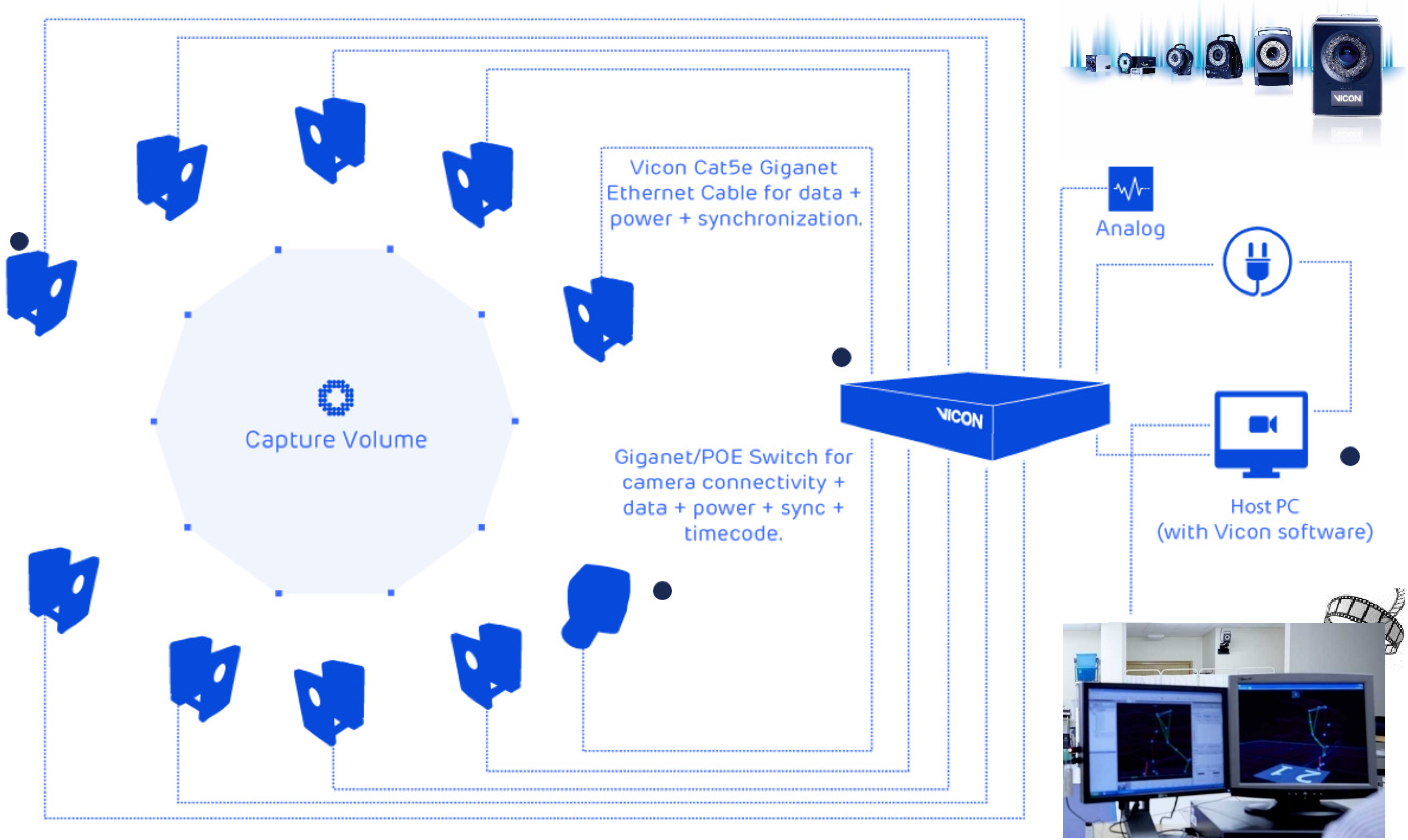
Global positioning system (GPS)

The **Global Positioning System** (**GPS**) is a space-based radio-navigation system owned by the US Government and operated by the US Air Force. It is a global navigation satellite system that provides geolocation and time information to a GPS receiver in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The GPS system provides critical positioning capabilities to all users around the world.

GPS is widely used by drones for outdoor applications nowadays. The incorporation of GPS receivers in drones allows for waypoint navigation. It allows a drone to autonomously fly to pre-programmed points, can instruct the drone how fast, how high, and where to fly.

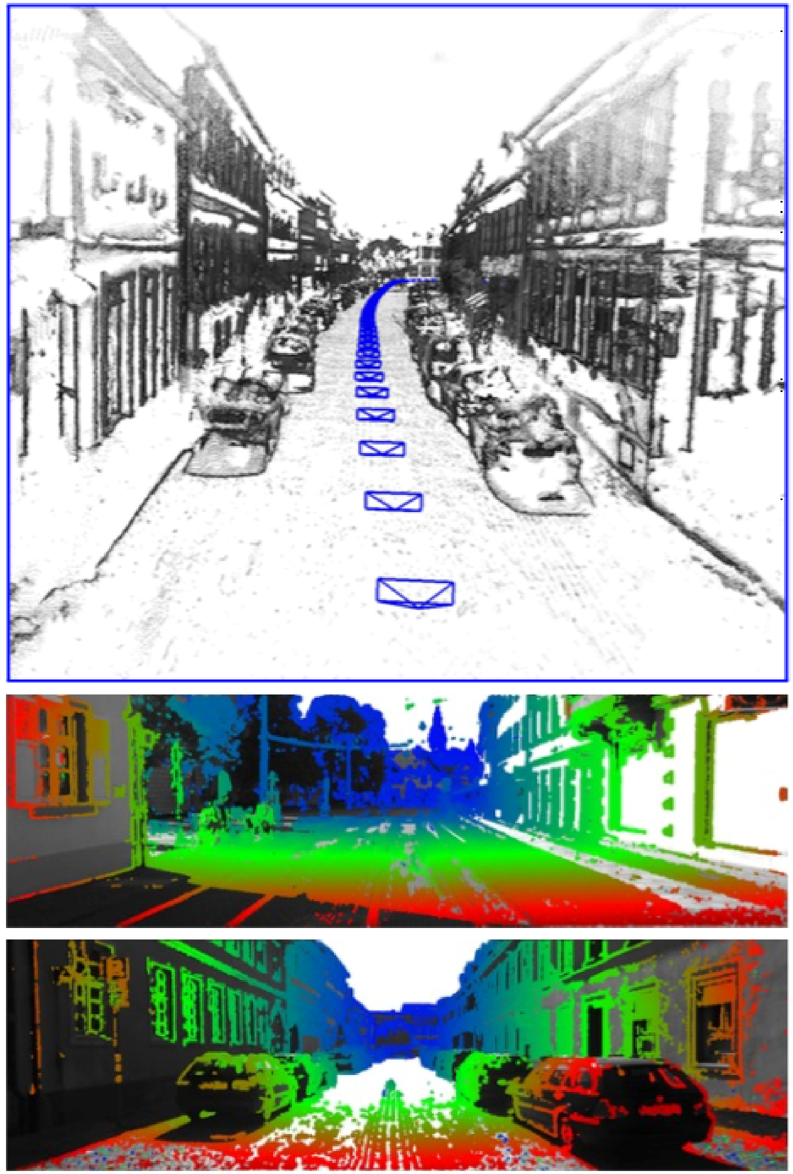


VICON: A motion capture positioning system…



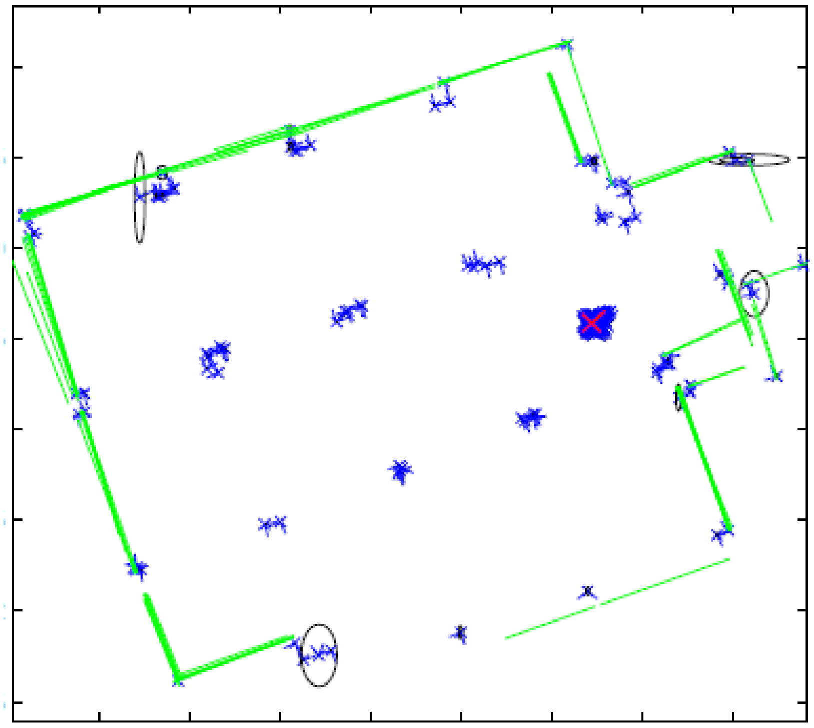
[Here is a link to YouTube for a tutorial on VICON…]

Sensor-based positioning systems... SLAM



Vision based 3D Dense SLAM

LiDAR based 2D SLAM



Topological SLAM techniques…

List of topological SLAM methods and their features/associated extraction algorithms…

- Line extractor (LiDAR)

  Split & merge, expectation maximization, Hough transform, RANdom Sample Consensus
- Haar wavelets (Vision)

  Fourier transform similar approach
- Edge based detector (Vision)
- Keypoint detectors

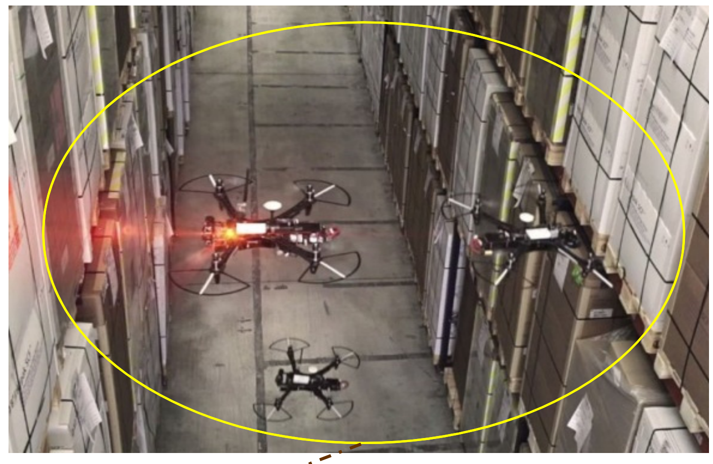
  Blob Detectors: Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Center Surround Extremas (SenSurE)

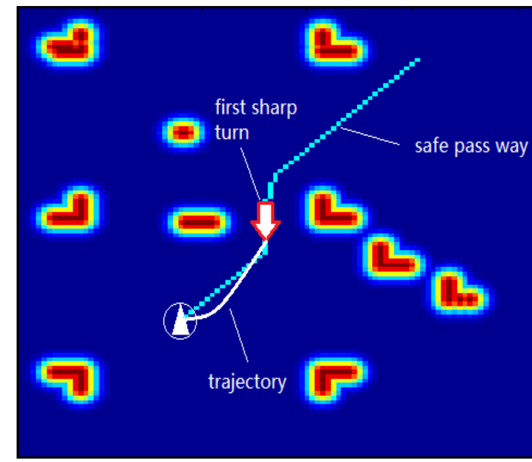
  Other Detectors: corner (KLT)
- Fingerprint of places

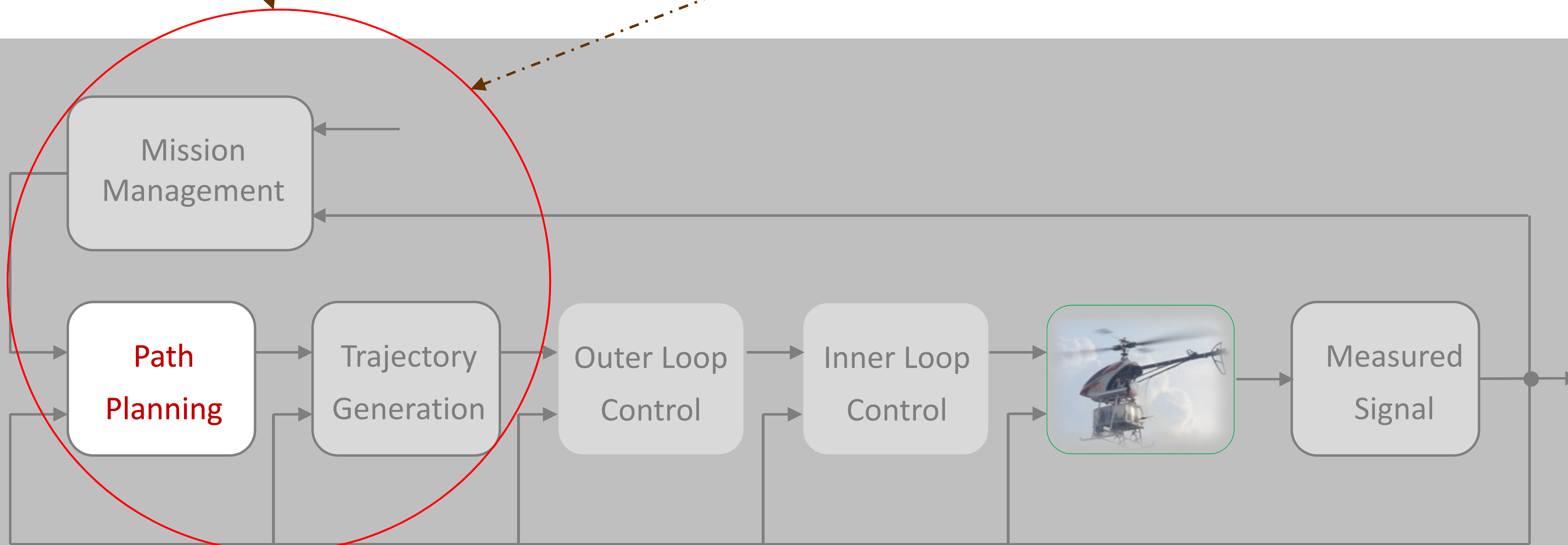
  FACT, DP-FACT
- Affine covariant region detector

  Harris-affine, Maximally Stable Extremal Regions (MSER)
- Bayesian surprise

Path planning...



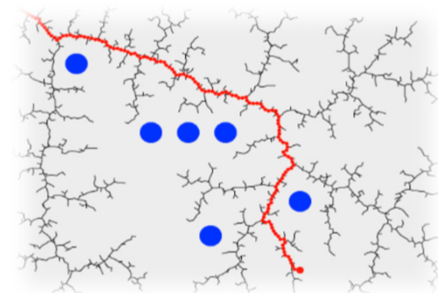


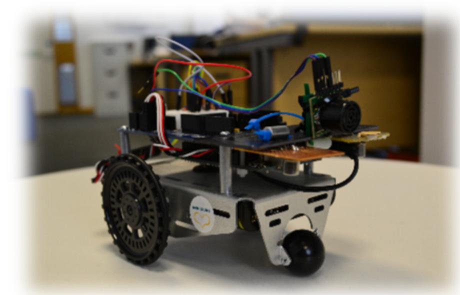


Motion planning...

Motion planning (also known as the **navigation problem**) is a term used in robotics or unmanned systems in general for the process of breaking down a desired movement task into discrete motions that satisfy movement constraints and possibly optimize some aspect of the movement, such as time and/or energy.

For example, consider navigating a UAV inside a room. The system should execute this task while avoiding walls and not hit the poles placed in the room. A motion planning algorithm would take a description of these tasks as input, and produce the speed and turning commands sent to the drone. It might address issues related to multiple UAVs, more complex tasks, different constraints (e.g., velocity, acceleration, heading), and uncertainty (e.g. imperfect models of the environment or UAV). Motion planning algorithms are widely used for ground robotic systems.





Motion planning…

Motion planning is to search an appropriate path for unmanned vehicles that would allow the vehicles to travel safely and efficiently among obstacles. It consists of two parts: the geometrical path planning and the trajectory generation or optimization.

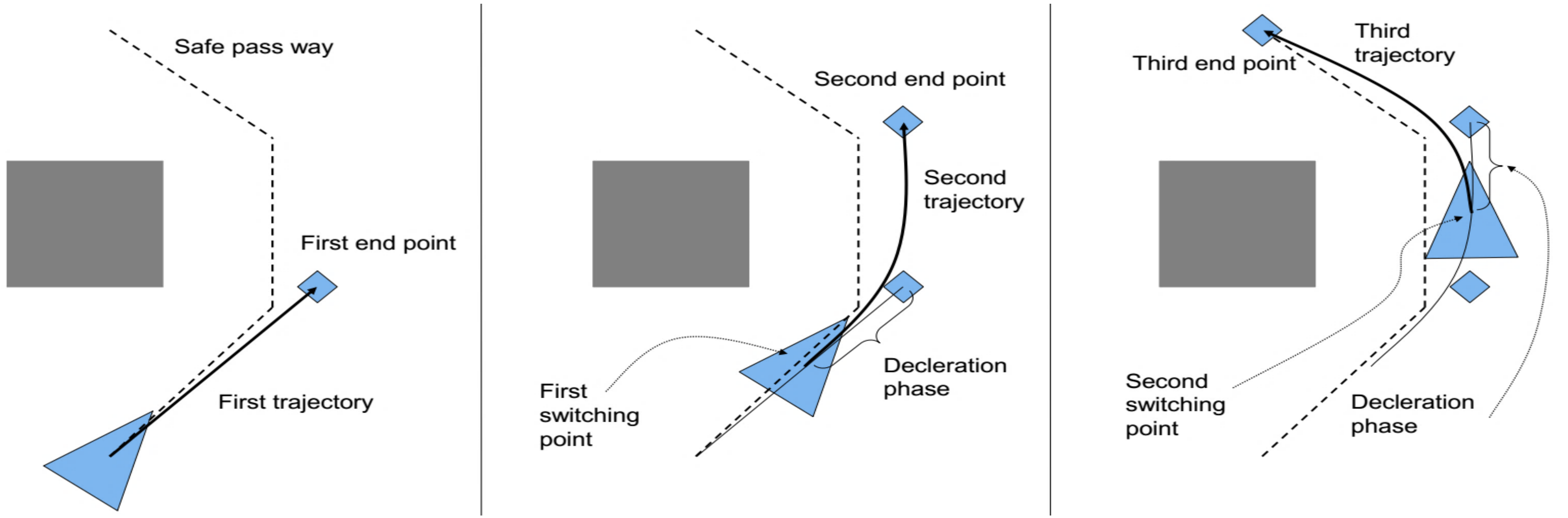
The **path planning** is done in the configuration space with all dynamics of the vehicles being ignored. This covers a lot of classical problems such as the piano mover's problem.

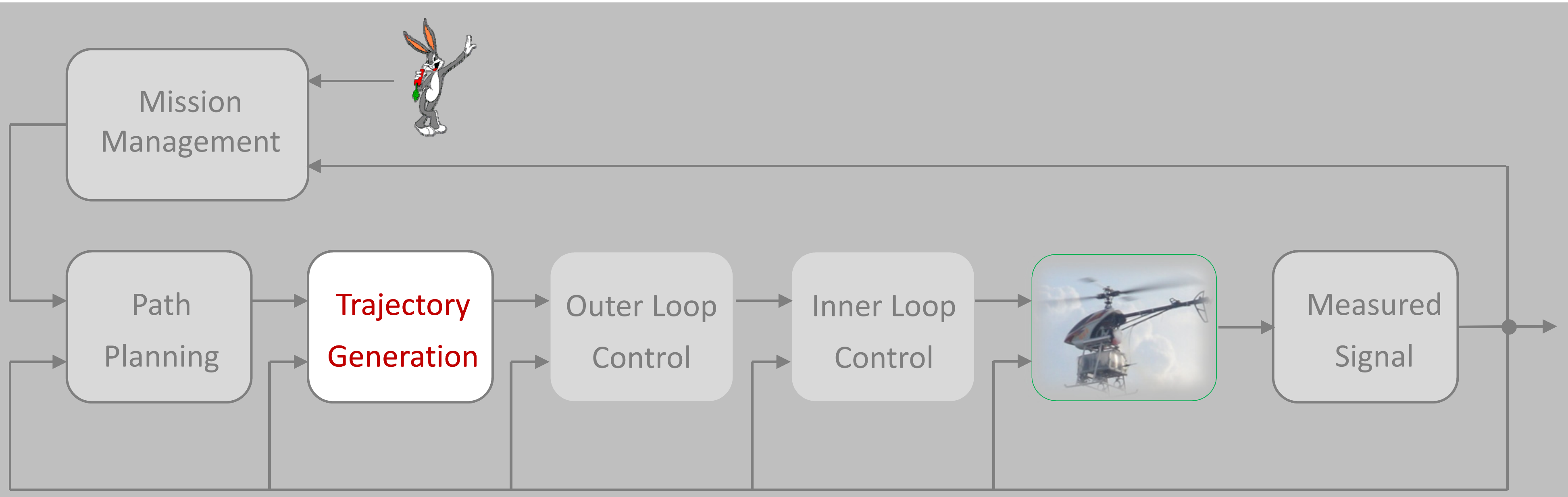
The **trajectory generation** is the process of designing a trajectory that minimizes (or maximizes) some measure of performance while satisfying a set of constraints of the dynamic model of the unmanned vehicles. Generally speaking, trajectory generation or optimization is a technique for computing an open-loop solution to an optimal control problem. It is often used for systems where computing the full closed-loop solution is either impossible or impractical.

Other path planning search techniques…

- **Dijkstra's algorithm**: Classic graph searching algorithm originated from dynamic programming
- **R* algorithm**: Optimized based on A* which depends less on the heuristic function
- **D* lite algorithm**: An incremental planning algorithm, build to handle a dynamic changing and unknown environment
- **JPS algorithm**: Jumping point search algorithm - an improved version of A*, created to handle the unnecessary zig-zag created by standard A* algorithm
- **PRM**: Probability roadmap, using random sampling to created connected graph, then use traditional graph search method to generate the map
- **RRT**: Rapidly-exploring random tree, combine the sampling and searching in the same algorithm, however it does not guarantee an optimal solution
- **RRT***: An improved version of RRT, guarantee an optimal solution
- **BIT***: Batch informed trees, a combination of random sampling algorithm with the A* algorithm

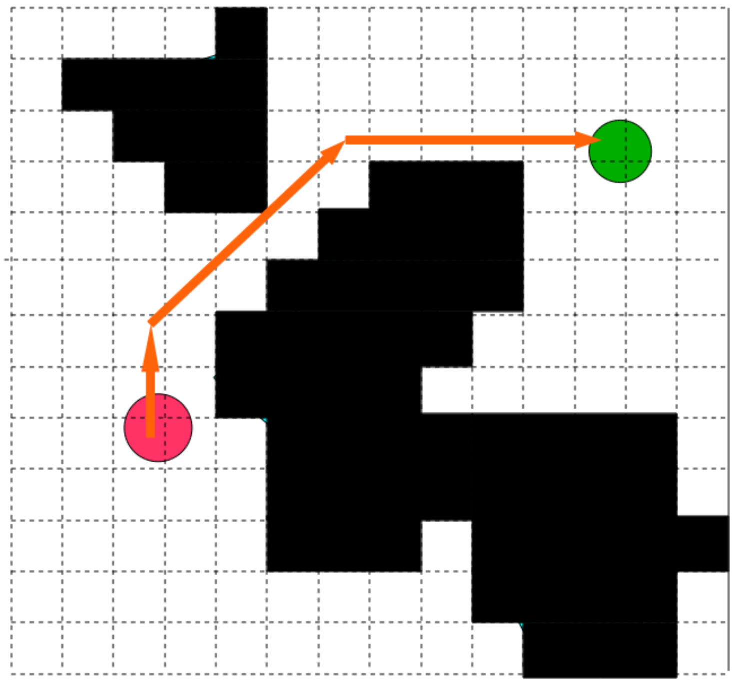
Trajectory generation...

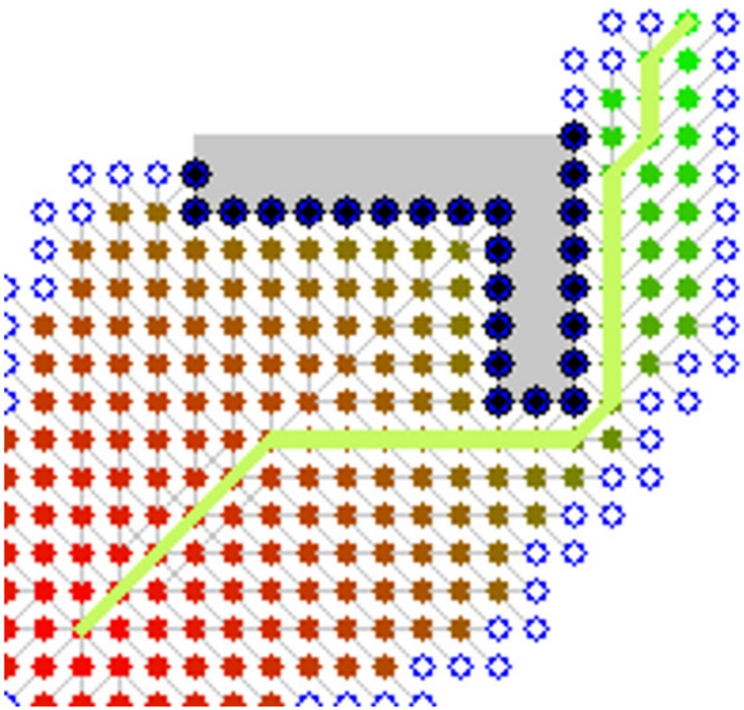




Trajectory generation for quadrotor drones…

With the A* algorithm, we are able to obtain a series of connected line segments shown as in the figures below





The next question one would ask: How can these line segments be realized (or used to guide) in a quadrotor drone?

Trajectory generation for quadrotor drones…

For a quadrotor drone to follow closely a pre-planned path, we need to specify a set of references to the outer-loop controller as depicted in the general unmanned system framework. For drones, the reference set should consist of the vehicle's

- 3-axis position references, x, y, z
- 3-axis velocity references, v_x, v_y, v_z
- 3-axis acceleration references, a_x, a_y, a_z

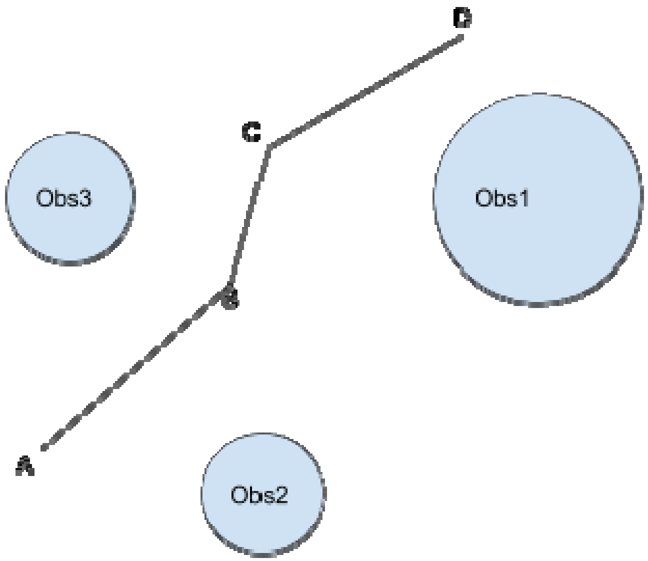
There are extra requirements on these references

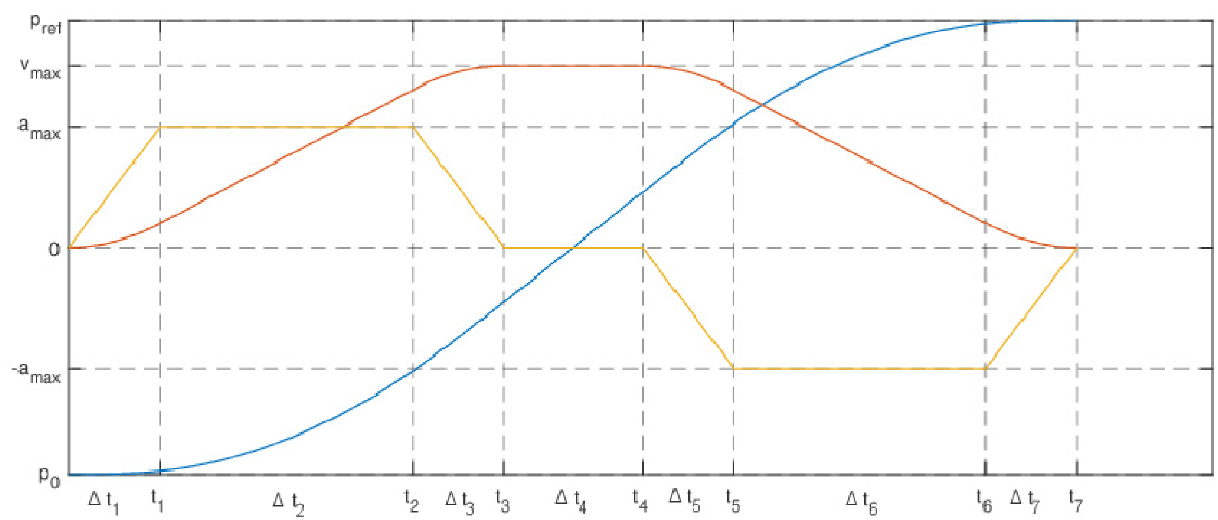
- All of the reference signals must be continuous
- The velocity and acceleration must be limited
- Some other constraints

Trajectory generation for quadrotor drones…

NUS National University of Singapore

The simplest way for the vehicle to travel on the segment path is to generate velocity profile along that line segment. For example,





red: velocity — yellow: acceleration — blue: position

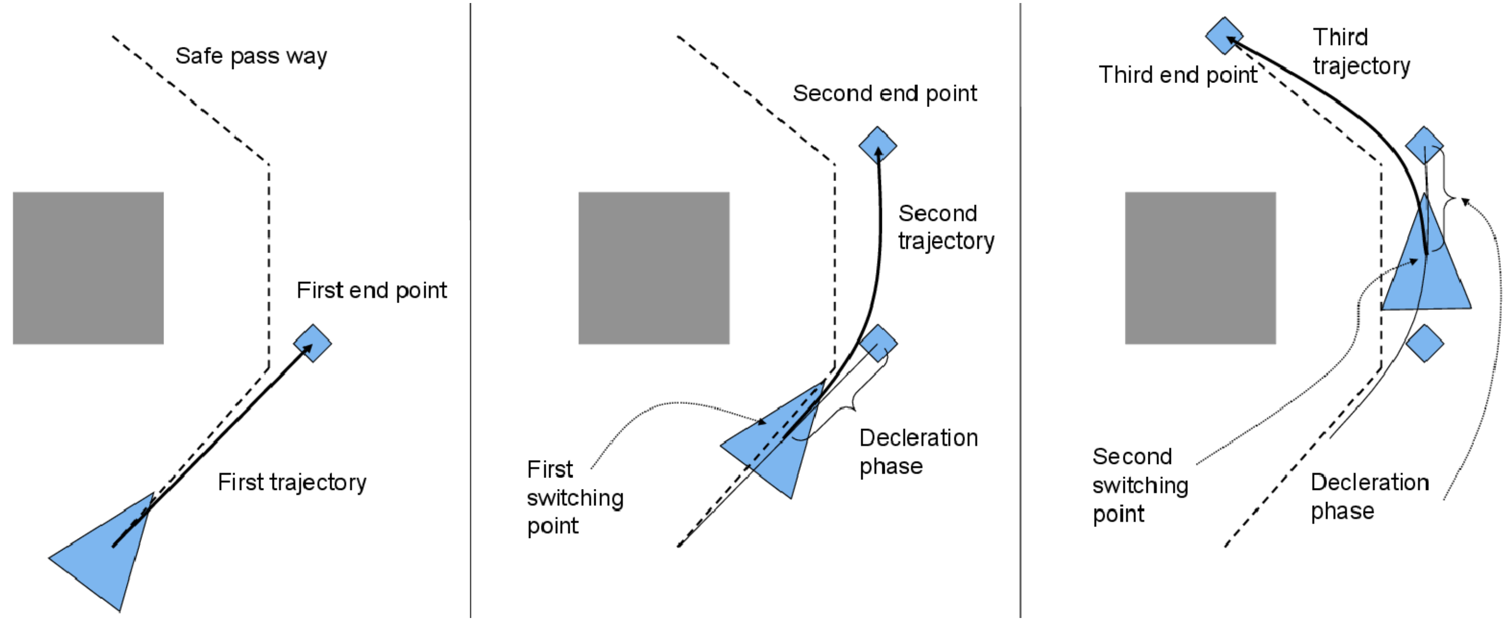
It is to generate trajectories for the vehicle to travel from A to D. For each line segment A–B, B–C, C–D, a velocity and acceleration profile shown above has to be generated.

The problem with this simple approach is that the drone will come to a full stop at the end of each line segment, such as point B and C.

Trajectory generation for quadrotor drones…

How can the drone fly from A→D as fast as possible without stops at the interim notes?

A possible solution is to ‘switch’ to the next line segment before it goes into full stop.



Step 1: Line segments are generated, vehicle flying towards the end of the 1st line segment.

Step 2: Before reaching the 1st end point to stop, switch to travel on the next line segment.

Step 3: By repeating the process in Step 2, vehicle could reach its target in a smoother way.

The difficulty is on maintaining the continuity of the reference signal for the entire path…

How to generate smooth trajectories?...

Trajectory generation techniques suitable for quadrotors…

- **Acceleration limited time optimal solution**

  Generate time minimal trajectory, under the velocity and acceleration constraint

- **Jerk limited time optimal solution**

  Time minimal trajectory, under the velocity, acceleration and jerk constraint

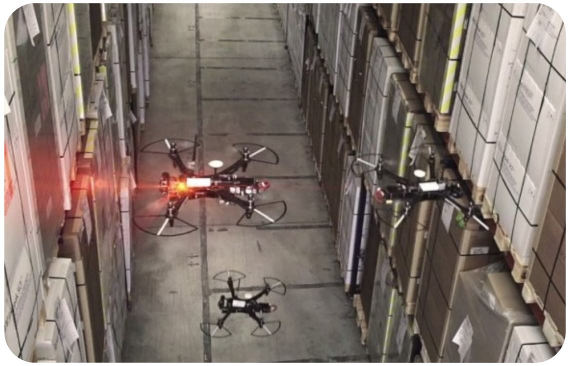
- **Polynomial spline based trajectory**

  Generate high quality, energy optimized trajectory, but requires the involving of numerical optimization

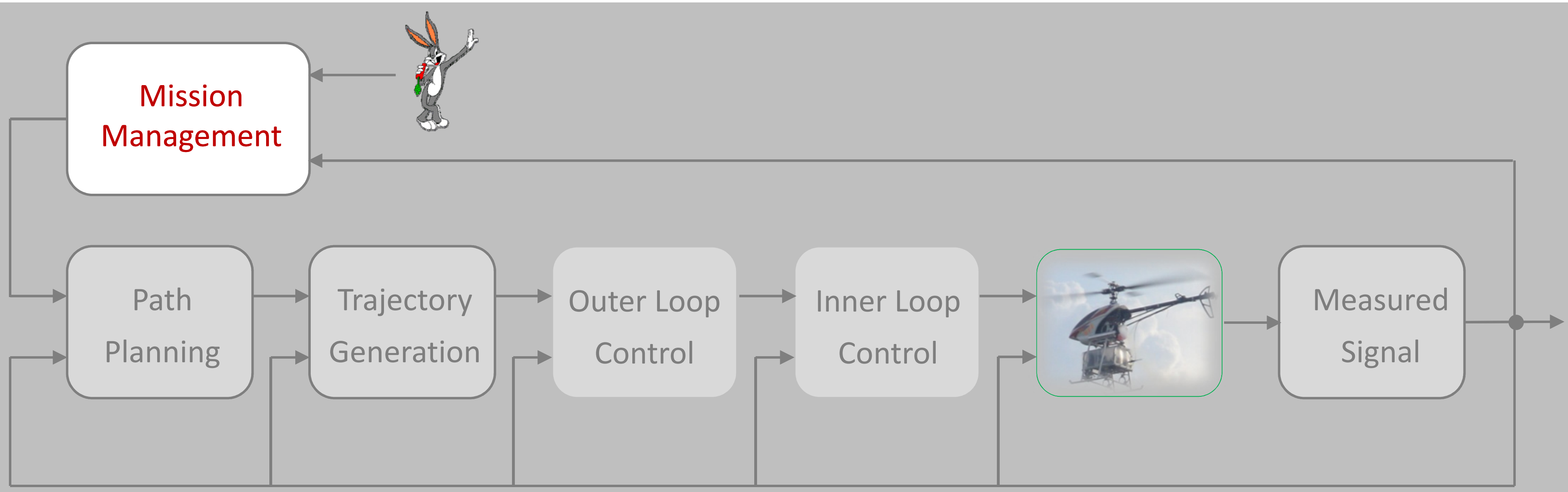
- **Dynamic programming**

  A search based trajectory generation method, very powerful for handling complex dynamics

Mission management…



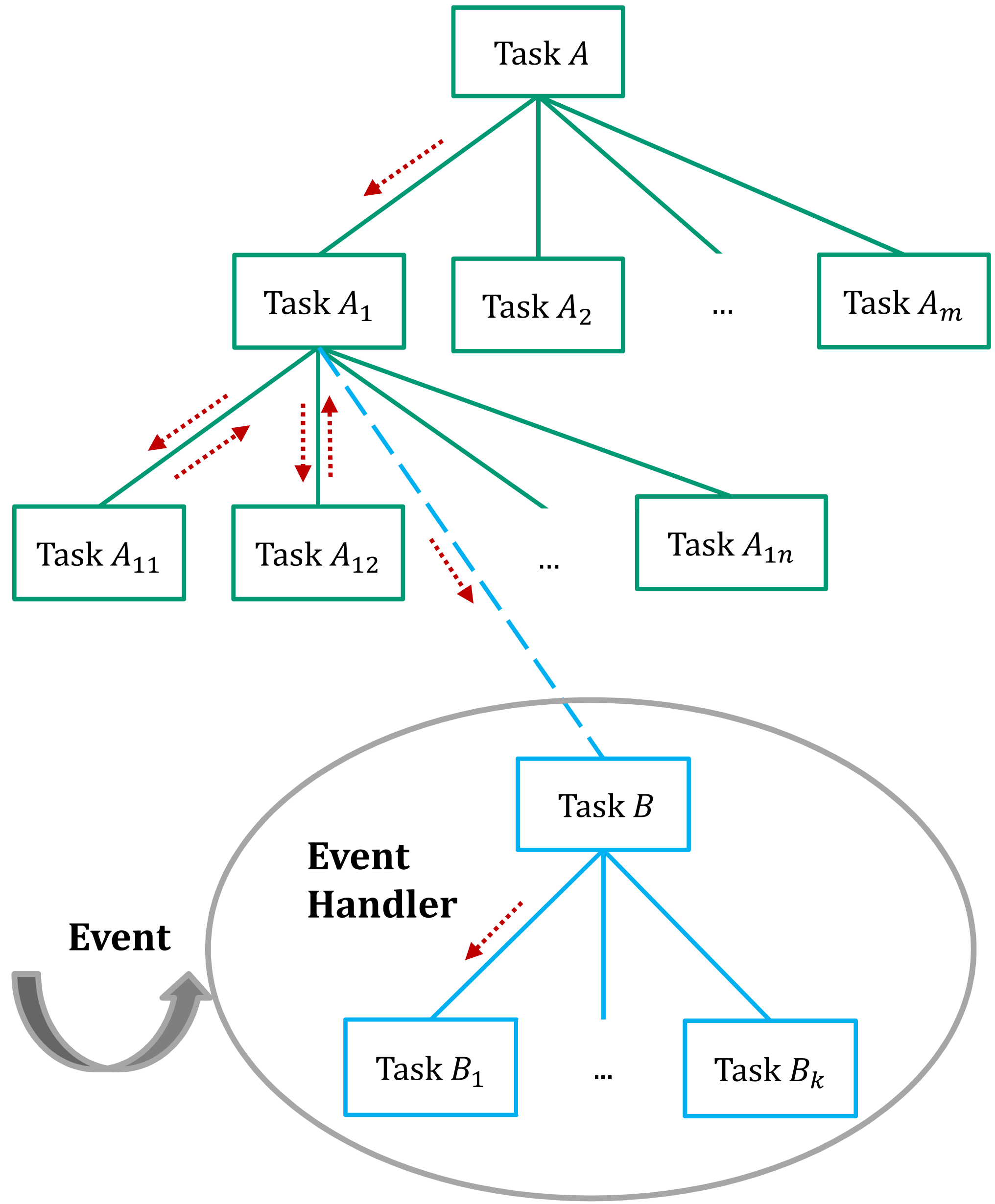
- State Machine
- Automata
- Deep learning? ……



Framework of a mission management…

For a task-based mission, we can use a tree-based framework. The tasks are organized into a **tree** and executed in a manner of **depth-first traversal**. Each leaf node task contains only one single **action**. In other words, executing that leaf node task is equivalent to executing the corresponding action.

When an **event** occurs, a new **task** (event handler) is inserted to the current tree node as a sub-tree. Some events require an immediate termination of mission after the event handler is executed (such as LAND once the battery is low). In that case, the remaining tasks are removed from the tree accordingly.



Acknowledgement…

… Special thanks to the whole NUS UAV Research Group

and particularly to

Lai Shupeng, Lan Menglu, Lin Feng, Li Kun, Phang Swee King, Tian Hongyu, Li Jiaxin, Wang Kangli

for their help in preparing lecture materials …

That's all, folks!

Thank You!