Aerial Robotics

FIELD AND SERVICE ROBOTICS

UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

www.prisma.unina.it

- Aerial robotics has become a new frontier in field and service robotics

- Really useful?

- Business opportunity

Delivering?

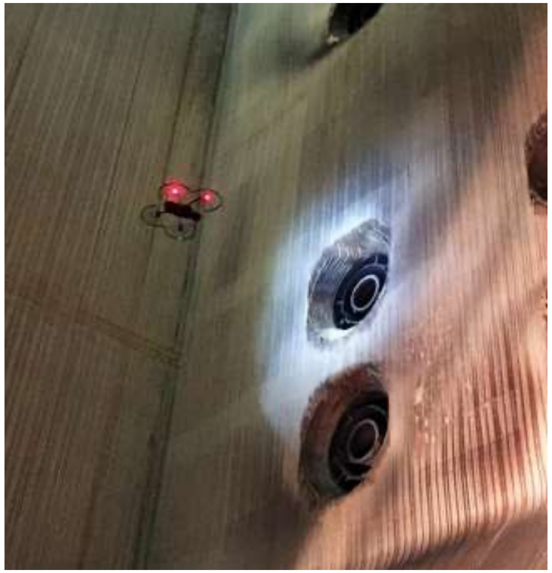
Industry?

Marketing?

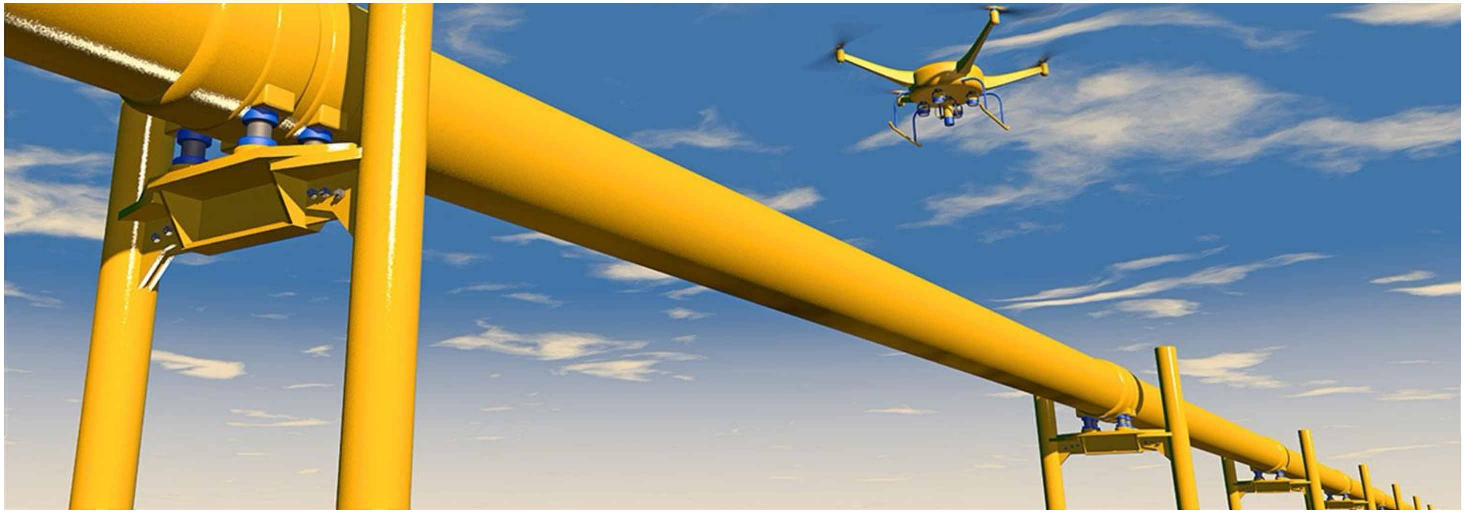
- Aerial robotics applications
  - Transportation
  - Manipulation
  - Inspection and maintainance (I&M)
  - Reparing
  - Montiroing
  - Patrolling
  - Exploration
  - Intervention in devastated areas
  - …

- Oil and gas facilities
  - What is possible:
    - 3D mapping of existing plants
    - Visual inspection
    - Thermographic inspection
    - Direct measurements (e.g., thickness)
    - First intervention and monitoring
  - What is needed:
    - ATEX compliant solutions
    - Power remote-handling for maintenance and (dis)assembling

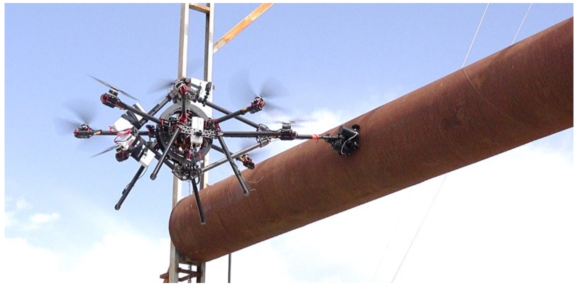
- Drone for inspection in confined spaces



- Drone for long pipe inspection



- Drone for thickness measurements



- History
  - Leonard Da Vinci's notes
  - First autonomous flight of an unmanned aerial vehicle (UAV) on 1896
    - Number 5 vehicle, designed by Samuel P. Langley, that travelled 400 meters
    - Number 6 vehicle travelled 1600 meters
  - In 1916, during the First World War, the Aerial Target vehicle was radio controlled (RC)
  - Hewitt-Sperry vehicle in 1916

History

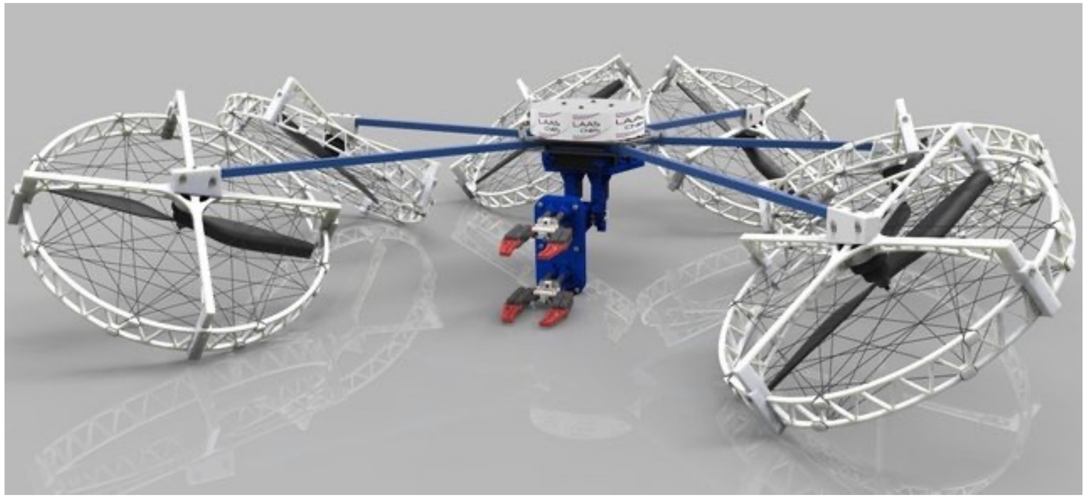
- Before the Second World War the airplanes were converted into autonomous vehicles controlled by autopilots
  - The first society, the *Radioplane Company*, specialized in building was founded by Reginald Denny
  - 1500 radio-controlled helicopters were built for the Second World War
- Technological development, smaller processors, lightweight sensor measurement systems, global navigations systems and so on boosted UAVs growth
  - In Japan, Yamaha Motor company developed an unmanned helicopter for irrigation in 1983
  - An autonomous UAV flying with GPS was out in 1988
    - It observed volcanic activities at Usu-zan in 2000

- Acronyms
  - UAV – Unmanned Aerial Vehicle
  - UAS – Unmanned Aerial System
  - RPV – Remotely Piloted System
  - ROA – Remotely Operated Aircraft
  - UVS – Unmanned Vehicle System

- Fixed-wing UAVs
  - They require a track to take off and land
  - High robustness, high cruise velocity

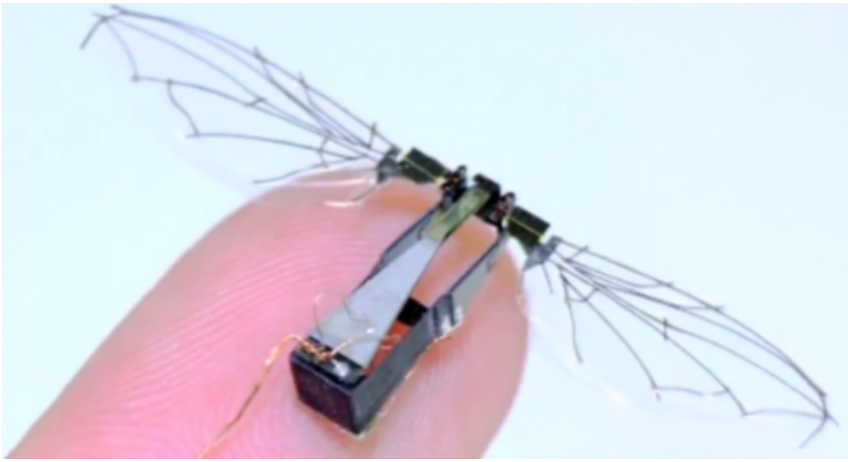
- Rotary-wing UAVs
  - Vertical take off and landing (VToL) vehicles
  - Hovering
  - High manoeuvrability
  - Helicopters, quadrotor, hexarotors, ….

- Tilted (passive) or tilting (active) configurations



- Blimp
  - Lighter-than-air
  - High robustness
  - Low velocity

- Flapping wings
  - Bio-inspired



- Taxonomy about dimension and flight time
  - HALE (High Altitude Long Endurance)
  - MALE (Medium Altitude Long Endurance)
  - Tactical UAVs
  - Small and man-portable UAVs



- MAVs (Micro Aerial Vehicles)

- Performance of aerial robots depends on dimensions and flying mechanisms (wings, rotors, …)
- Effects of dimensions on a quadrotor
  - Blade velocity $v \propto \sqrt{R}$
  - Lift force $F \propto R^3$
  - Mass m $\propto R^3$
  - Inertia $I \propto R^5$
  - Angular acceleration $\dot{\omega} \propto \frac{1}{R}$

- Propulsion systems
  - Jet
  - Internal combustion
  - Spacecraft propulsion
  - Electrical
- Flight regimes
  - Hovering
    - Tailsitters robot
  - Cruising flight

Physical interaction of the UAVs with the environment

- Physical interaction while flying is relevant for many applications
  - Interaction with other flying objects
    - Refuelling
    - Object transportation
    - Capture of drones for security reasons
    - Docking
    - Maintenance and repairing

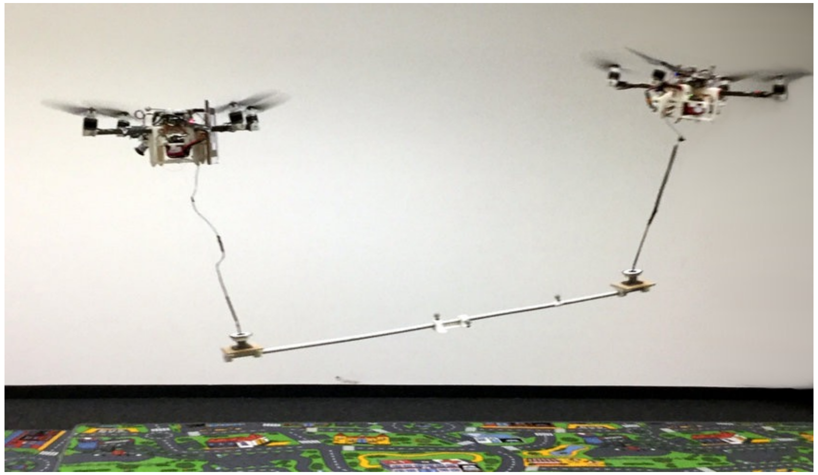
Physical interaction of the UAVs with the environment

- Interaction with ground objects
  - Mobile objects
    - Picking mobile targets
    - Landing on mobile platforms
  - Fixed objects
    - Picking fixed targets
    - Cleaning
    - Contact inspection
    - Manipulation

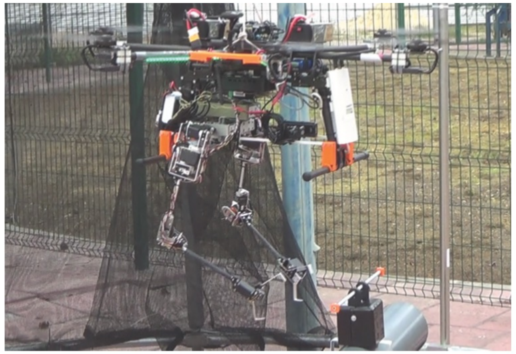
Physical interaction of the UAVs with the environment

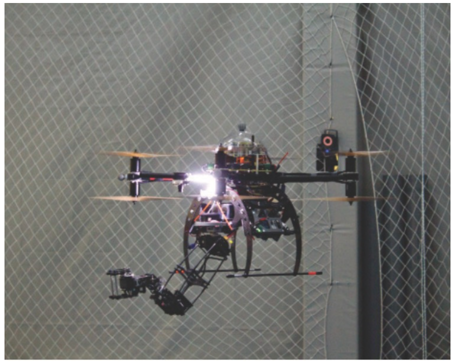
- Problems
  - Stability of the aircraft during the interaction
  - Accuracy
    - Trajectory tracking
    - Positioning with respect to targets
    - Grasping
    - Physical contact with objects
  - Aerodynamic perturbations due to proximity to surfaces
  - Payload required to carry devices for physical interaction
  - Required flight time
  - Reactivity to cancel the effect of wind perturbations flying close to objects
  - Planning

- Aerial manipulation is a special case of physically interacting UAVs with the environment
- Aerial manipulation can be achieved through
  - Flying hands
    - Object or rigid tool directly attached to the UAV
    - The object is linked or tethered



- Unmanned aerial manipulators (UAMs)



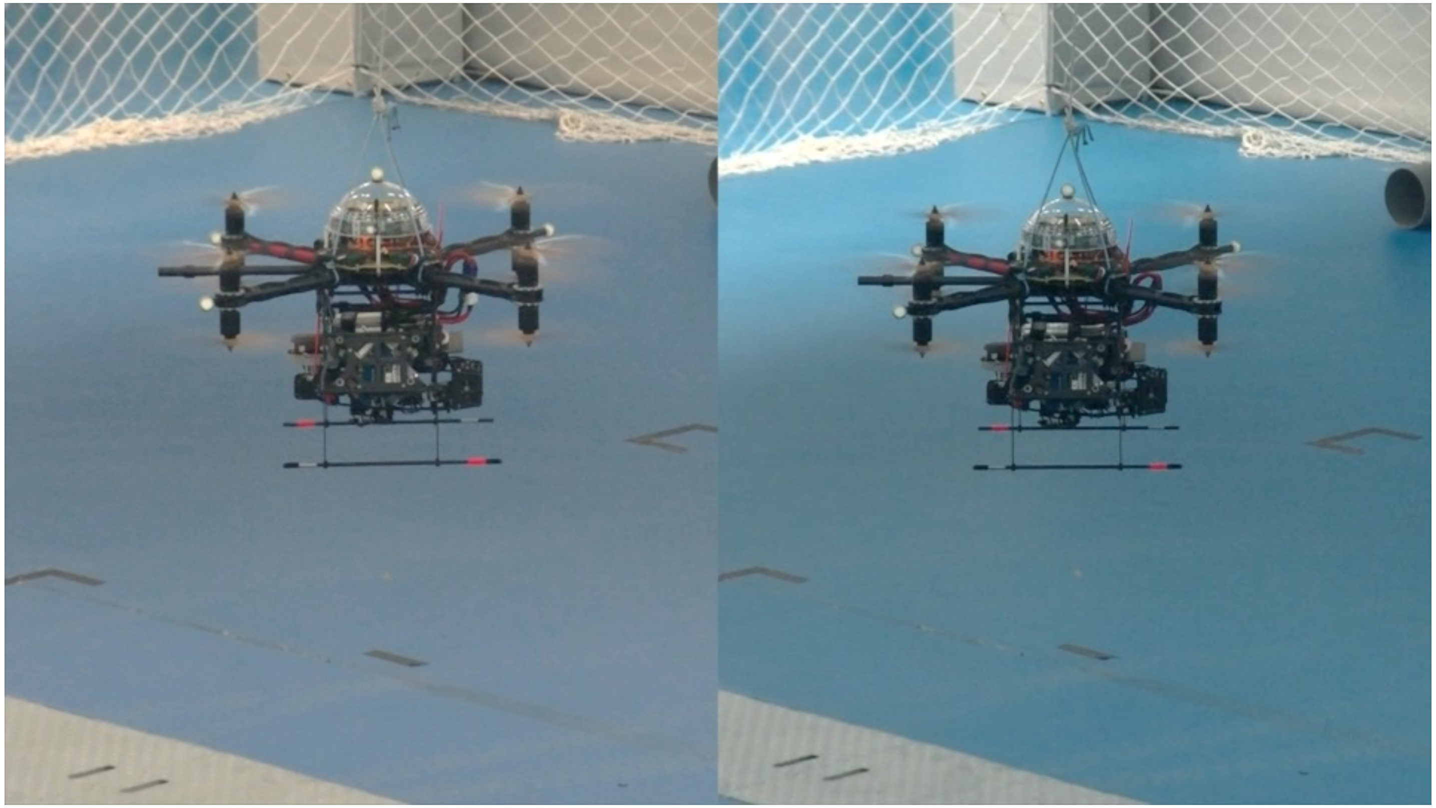


- UAMs are constituted by the following elements
  - Floating base (UAV)
  - Robotic arm
  - Gripper or multi-fingered hand (end-effector)
  - Sensors (cameras, laser scanners,…)
- UAMs can be classified in several ways
  - Number of DoFs of the arm, or number of arms on the device (single arm, dual-arm,…)
  - Mechanism of the joints
  - Control modality of the motors
  - Resulting configuration
- The mounted robot arm creates issues due to the coupling dynamics
- Two ways to control the UAM
  - Centralized approach
    - The UAV and the robotic arm are seen as a unique entity
  - Decentralized approach
    - The UAV and the robotic arm are seen as independent systems

Reactiveness of the proposed estimator
Interaction with a human operator

www.youtube.com/watch?v=iHKtHF0LF-w

Research videos



www.youtube.com/watch?v=4l_2Pa_xOUM

Research videos

www.youtube.com/watch?v=5hR-xYMbk50

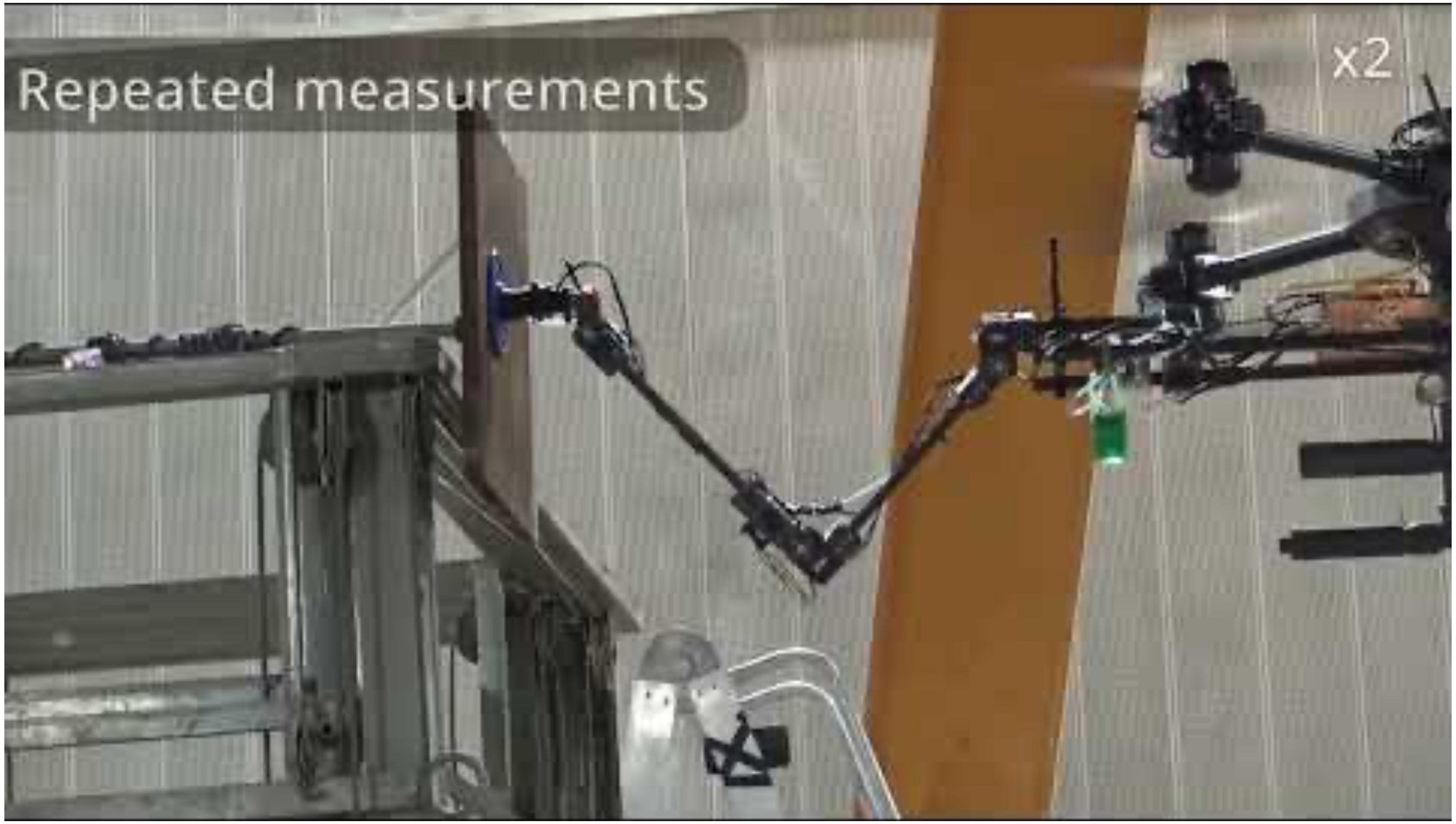
Development of a Control Framework to Autonomously Install Clip Bird Diverters on High-Voltage Lines

Simone D'Angelo, Francesca Pagano, Fabio Ruggiero, Vincenzo Lippiello

PRISMA Lab
Department of Electrical Engineering and Information Technology
University of Naples Federico II
www.prisma.unina.it

https://www.youtube.com/watch?v=nffDgxhORdc

Research videos



https://www.youtube.com/watch?v=ljDkv3oef_s

Research projects @ PRISMA Lab

www.aerotrain-etn.eu

www.aerial-core.eu

www.hyfliers.eu

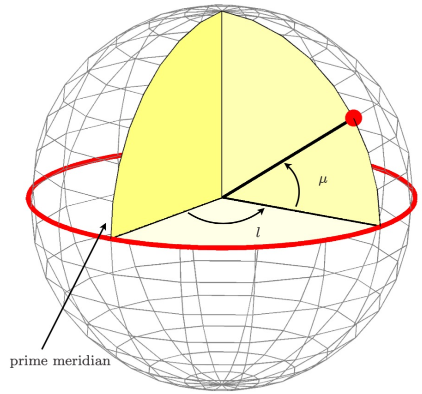
http://aeroarms-project.eu

http://cordis.europa.eu/project/rcn/106964_it.html

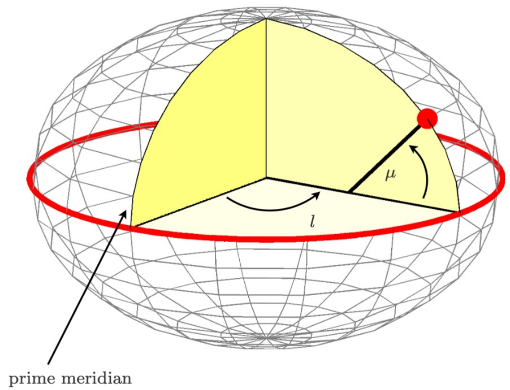
www.arcas-project.eu

http://airobots.dei.unibo.it

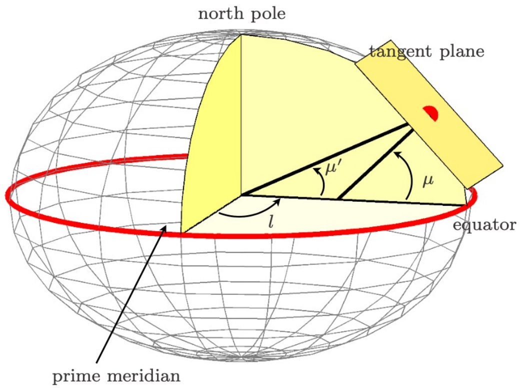
- Latitude - longitude – altitude
  - Latitude: north—south
  - Longitude: east – west



- Geographic latitude
  - Not referred to the earth's centre
    - Except for the poles and points on the prime meridian



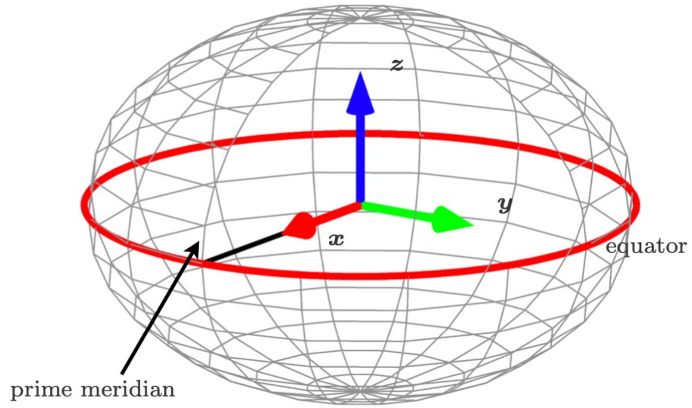
- Geocentric latitude



- Earth-centred inertial frame
  - The frame has its origin in the earth's centre
  - The orientation is fixed
  - The x-y plane lies o the equator plane
  - z-axis points to the north pole
  - The frame is not subject to any acceleration and it can be considered as inertial

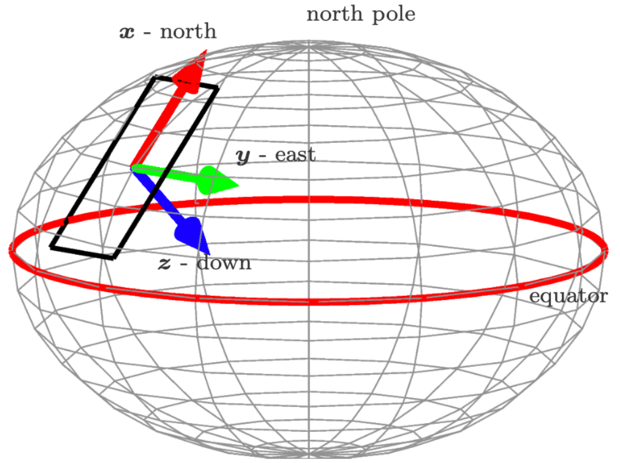
Earth centred earth-fixed frame

- Earth-centred earth-fixed frame
  - The frame has its origin in the earth's center
  - The orientation follows the earth's rotation
  - The x-y plane lies o the equator plane
  - z-axis points to the north pole
  - The frame is not inertial, but it can be assumed to be so
  - GPS gives positions in this frame



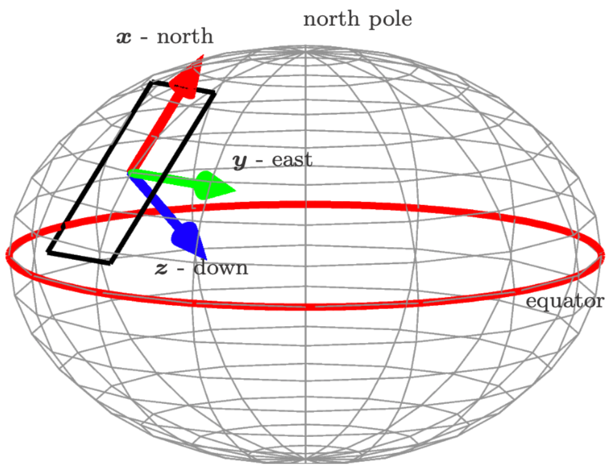
NED and ENU frames

- North-East-Down (NED) frame
  - It is local to the aera of interest
  - x-axis points to the north
  - y-axis points to the east
  - z-axis points down
  - Opposite to East-North-Up (ENU)



- Body frame
  - The origin is at any point of the robot
    - Usually it is located at the center of mass
    - Orientation of the axes is arbitrary
    - It moves together with the robot

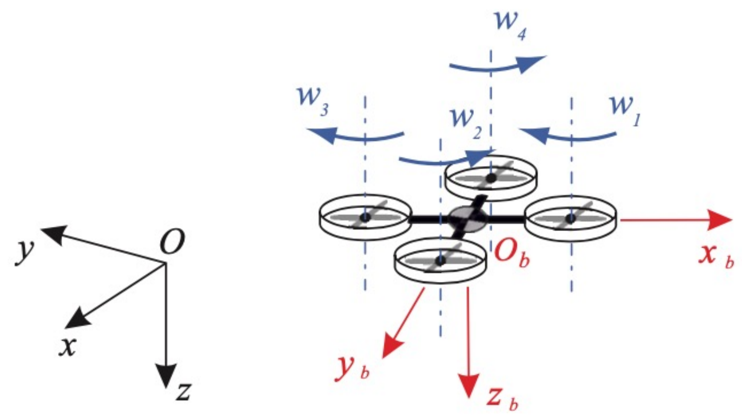
- North-East-Down (NED) frame
  - It is local to the aera of interest
  - x-axis points to the north
  - y-axis points to the east
  - z-axis points down
  - Opposite to East-North-Up (ENU)



- Body frame
  - The origin is at any point of the robot
    - Usually it is located at the center of mass
    - Orientation of the axes is arbitrary
    - It moves together with the robot

Definitions

- Frames
  - World frame (North-East-Down configuration)
  - Body frame (North-East-Down configuration)
- Pose
  - $p_b = [x \quad y \quad z]^T \in \mathbb{R}^3$ position
  - $R_b = [x_b \quad y_b \quad z_b] \quad \in SO(3)$ attitude
    - $SO(3)$ is the special orthogonal group of dimension three, made by all the three by three rotation matrices
    - The attitude can be also expressed trough a minimal representation given by the ZYX Euler angles (roll-pitch-yaw)
      - $\eta_b = [\varphi \quad \theta \quad \psi]^T \in \mathbb{R}^3$
- Inputs
  - The real inputs are the velocity for each propeller ω_i



$$R_b(\eta_b) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_p si - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

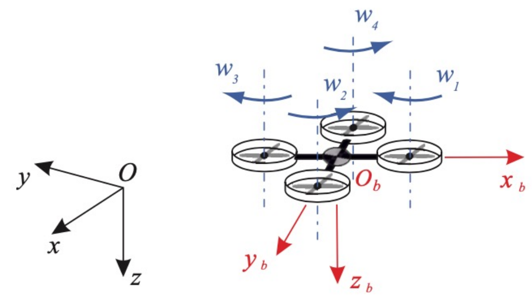
- Inputs
  - Each propeller produces a thrust $T_i > 0$
  - Considering a quadrotor, $i = 1, \ldots, 4$, the total thrust is $u_T = \sum_{i=1}^{4} T_i > 0$
    - The total thrust is directed along the z_b-axis of the body frame to compensate for gravity and control the vertical motion
    - The horizontal movements are controlled by directing the total thrust vector $u_T z_b$ in the appropriate direction (thrust vectoring control), thereby resulting in longitudinal and lateral force components
  - If we do not consider quadrotors, we can assume a control force $f^b = [f_x \quad f_y \quad f_z]^T \in \mathbb{R}^3$
    - In the quadrotor case, $f^b = [0 \quad 0 \quad -u_T]^T$
  - The control torques $\tau^b = [\tau_x \quad \tau_y \quad \tau_z]^T \in \mathbb{R}^3$ are generated in different ways according to the UAV configuration
    - For instance, in case of helicopters, τ^b are produced by using a tail rotor and swash plates (to change the angle of the main rotor blades)
    - They are exploited to control the UAV's attitude which in turn controls the UAV horizontal movement
  - The quadrotors are underactuated mechanical systems with control inputs $u_T > 0$ and $\tau^b \in \mathbb{R}^3$
  - We need to connect the control inputs with the real inputs of the system
    - To this purpose, we neglect the parasitic forces and moments induced by control inputs coupling, rotors dynamics, gyroscopic effects, and other small aerodynamic effects

Inputs

- With reference to the figure, the torques generated by the single propellers thrust are

$$\tau_x = l(T_4 - T_2)$$
$$\tau_y = l(T_1 - T_3)$$
$$\tau_z = -Q_1 + Q_2 - Q_3 + Q_4$$

  - $l > 0$ distance of each propeller to the UAV's centre
  - Q_i fan torque due to air drag



- The relation between the single propellers thrust and the velocity is approximated by

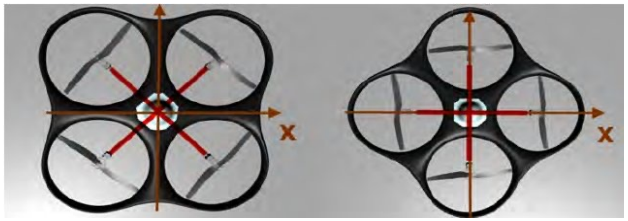
$$T_i = c_T \omega_i^2$$
$$Q_i = c_Q \omega_i^2$$

  - $c_T > 0$ thrust constant
  - $c_Q > 0$ drag factor
  - This is true if we are far from obstacles and considering a quasi-static analysis
- Therefore, the sought relation is

$$\begin{bmatrix} u_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & -lc_T & 0 & lc_T \\ lc_T & 0 & -lc_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix}}_{\text{Allocation matrix}} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

Allocation matrix

- The allocation matrix for a UAV with the propellers in a flat configuration is $G_q \in \mathbb{R}^{4\times n}$, with n the number of propellers
  - Notice that, if $n > 4$, but the propellers are co-planar, you have a redundancy in how to split the velocities among all the propellers, but the system is still underactuated since the n co-planar propellers do not give you the possibility to have more real inputs to the system than 4
  - To get $G_q \in \mathbb{R}^{6\times n}$ and avoid the underactuation the UAV must have either non co-planar propellers (tilted configurations) or motorised propellers (actively tilting configurations)
- The allocation matrix is invertible for quadrotors but it should be checked for different configurations



Dynamic model of a UAV

- The dynamic model of the UAV can be written in the body frame as

$$\begin{cases} m\ddot{p}_b^b = -mS(\omega_b^b)\dot{p}_b^b + mgR_b^T e_3 + f^b + f_e^b \\ \dot{R}_b = R_b S(\omega_b^b) \\ I_b \dot{\omega}_b^b = -S(\omega_b^b) I_b \omega_b^b + \tau^b + \tau_e^b \end{cases}$$

  - In the quadrotor case

$$\begin{cases} m\ddot{p}_b^b = -mS(\omega_b^b)\dot{p}_b^b + mgR_b^T e_3 - u_T e_3 + f_e^b \\ \dot{R}_b = R_b S(\omega_b^b) \\ I_b \dot{\omega}_b^b = -S(\omega_b^b) I_b \omega_b^b + \tau^b + \tau_e^b \end{cases}$$

  - $e_3 = [0 \quad 0 \quad 1]^T$
  - $I_b \in \mathbb{R}^{3\times 3}$ is the diagonal inertia matrix for the angular part and referred to the body frame
  - $m > 0$ mass of the UAV
  - g gravity acceleration
  - $f_e^b, \tau_e^b \in \mathbb{R}^r$ external or unmodelled terms expressed in the body frame

Coordinate-free dynamic model of a UAV

- For many control purposes, the linear part should be expressed in the world frame
- Recall that $\dot{p}_b = R_b \dot{p}_b^b$ and then $\ddot{p}_b = R_b \ddot{p}_b^b + \dot{R}_b \dot{p}_b^b = R_b \ddot{p}_b^b + R_b S(\omega_b^b)\dot{p}_b^b$
  - Multiply both sides of $m\ddot{p}_b^b = -mS(\omega_b^b)\dot{p}_b^b + mgR_b^T e_3 + f^b + f_e^b$ by R_b

$$mR_b\ddot{p}_b^b = -mR_b S(\omega_b^b)\dot{p}_b^b + mgR_b R_b^T e_3 + R_b f^b + R_b f_e^b$$

  - Substituting $R_b \ddot{p}_b^b = \ddot{p}_b - R_b S(\omega_b^b)\dot{p}_b^b$ into the previous one yields

$$m\ddot{p}_b - \cancel{mR_b S(\omega_b^b)\dot{p}_b^b} = \cancel{-mR_b S(\omega_b^b)\dot{p}_b^b} + mgR_b R_b^T e_3 + R_b f^b + R_b f_e^b$$

- Therefore, we obtain the coordinate-free dynamic model of a UAV

$$\begin{cases} m\ddot{p}_b = mge_3 + R_b f^b + f_e \\ \dot{R}_b = R_b S(\omega_b^b) \\ I_b \dot{\omega}_b^b = -S(\omega_b^b) I_b \omega_b^b + \tau^b + \tau_e^b \end{cases}$$

  - In the quadrotor case

$$\begin{cases} m\ddot{p}_b^b = mge_3 - u_T R_b e_3 + f_e \\ \dot{R}_b = R_b S(\omega_b^b) \\ I_b \dot{\omega}_b^b = -S(\omega_b^b) I_b \omega_b^b + \tau^b + \tau_e^b \end{cases}$$

RPY dynamic model of a UAV

- The dynamic model of the UAV can be written also in terms of the RPY angles as [1]

$$\begin{cases} m\ddot{p}_b = mge_3 + R_b f^b + f_e \\ M(\eta_b)\ddot{\eta}_b = -C(\eta_b,\dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b + Q^T(\eta_b)\tau_e^b \end{cases}$$

  - In the quadrotor case

$$\begin{cases} m\ddot{p}_b = mge_3 - u_T R_b e_3 + f_e \\ M(\eta_b)\ddot{\eta}_b = -C(\eta_b,\dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b + Q^T(\eta_b)\tau_e^b \end{cases}$$

  - $M(\eta_b) = Q^T(\eta_b) I_b Q(\eta_b) \in \mathbb{R}^{3\times3}$ symmetric and positive definite provided that $\theta \neq \pm\frac{\pi}{2}$
  - $C(\eta_b,\dot{\eta}_b)$= $Q^T(\eta_b) S(Q(\eta_b)\dot{\eta}_b) I_b Q(\eta_b) + Q^T(\eta_b) I_b \dot{Q}(\eta_b) \in \mathbb{R}^{3\times3}$ with $S(*) \in \mathbb{R}^{3\times3}$ the skew-symmetric operator
  - $Q(\eta_b) \in \mathbb{R}^{3\times3}$ is the transformation matrix such that $\omega_b^b = Q(\eta_b)\,\dot{\eta}_b$, with $\omega_b^b \in \mathbb{R}^3$ the angular velocity of the body frame with respect to the world frame expressed in the body frame

$$Q = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix}$$

  - $Q^T(\eta_b)\tau^b$ is an example of kinetostatic duality

[1] A. Ollero, B. Siciliano (Eds.), "Aerial robotic manipulation," Springer, 2019.

- Sketch of the proof

$$\omega_b^b = Q(\eta_b)\,\dot{\eta}_b \Longrightarrow \dot{\eta}_b = Q(\eta_b)^{-1}\omega_b^b, \text{ provided that } \theta \neq \pm\frac{\pi}{2}$$

$$\ddot{\eta}_b = \dot{Q}^{-1}\omega_b^b + Q^{-1}\dot{\omega}_b^b$$

  - We replace the coordinate-free quadrotor dynamic model for the angular in $\dot{\omega}_b^b$

$$\ddot{\eta}_b = \dot{Q}^{-1}\omega_b^b + Q^{-1}I_b^{-1}\left(-S\left(\omega_b^b\right)I_b\omega_b^b + \tau^b + \tau_e^b\right)$$

$$\ddot{\eta}_b = \dot{Q}^{-1}Q\,\dot{\eta}_b + Q^{-1}I_b^{-1}\left(-S(Q\,\dot{\eta}_b)I_b Q\,\dot{\eta}_b + \tau^b + \tau_e^b\right)$$

  - We multiply both sides of the last expression by $M(\eta_b) = Q^T I_b Q$

$$M(\eta_b)\,\ddot{\eta}_b = Q^T I_b Q\,\dot{Q}^{-1}Q\,\dot{\eta}_b + Q^T I_b Q\,Q^{-1}I_b^{-1}\left(-S(Q\,\dot{\eta}_b)I_b Q\,\dot{\eta}_b + \tau^b + \tau_e^b\right)$$

$$M(\eta_b)\,\ddot{\eta}_b = Q^T I_b Q\,\dot{Q}^{-1}Q\,\dot{\eta}_b + Q^T\left(-S(Q\,\dot{\eta}_b)I_b Q\,\dot{\eta}_b + \tau^b + \tau_e^b\right)$$

  - The following expression holds $\dot{Q} = -Q\dot{Q}^{-1}Q \Longrightarrow \dot{Q}^{-1}Q = -Q^{-1}\dot{Q}$
  - Therefore

$$M(\eta_b)\,\ddot{\eta}_b = Q^T I_b \dot{Q}\quad\dot{\eta}_b + Q^T\left(-S(Q\,\dot{\eta}_b)I_b Q\,\dot{\eta}_b + \tau^b + \tau_e^b\right)$$

from which it is possible to have the model seen before

RPY dynamic model of a UAV in a compact form

- Stacking $\zeta = [p_b^T \quad \eta_b^T]^T$ from the RPY dynamic model it is possible to obtain the RPY dynamic model of a UAV in a compact form

$$M_\zeta \ddot{\zeta} + C_\zeta \dot{\zeta} + g_\zeta = \Lambda u + \xi_e$$

  - $M_\zeta = \begin{bmatrix} mI_3 & O_{3\times3} \\ O_{3\times3} & M(\eta_b) \end{bmatrix} \in \mathbb{R}^{6\times6}$ symmetric and positive definite provided that $\theta \neq \pm\frac{\pi}{2}$
  - $C_\zeta = \begin{bmatrix} O_{3\times3} & O_{3\times3} \\ O_{3\times3} & C(\eta_b, \dot{\eta}_b) \end{bmatrix} \in \mathbb{R}^{6\times6}$
  - $g_\zeta = \begin{bmatrix} mge_3 \\ 0_3 \end{bmatrix} \in \mathbb{R}^6$
  - $\Lambda = \begin{bmatrix} R_b & O_{3\times3} \\ O_{3\times3} & Q^T(\eta_b) \end{bmatrix} \in \mathbb{R}^{6\times6}$ and $u = \begin{bmatrix} f^{b^T} & \tau^{b^T} \end{bmatrix}^T$
    - $\Lambda = \begin{bmatrix} R_b e_3 & O_{3\times3} \\ 0_3 & Q^T(\eta_b) \end{bmatrix} \in \mathbb{R}^{6\times4}$ and $u = \begin{bmatrix} u_T & \tau^{b^T} \end{bmatrix}^T$ in the quadrotor case
  - $\xi_e = \begin{bmatrix} f_e \\ Q^T(\eta_b)\tau_e^b \end{bmatrix} \in \mathbb{R}^6$

Quadrotor flat outputs

- The quadrotor is a differentially flat system
  - (x, y, z, ψ) are the flat outputs
- Therefore, it is reasonable to plan the UAV path/time law on the basis of the flat outputs
- Notice that the full attitude of the UAV cannot be specified, but it depends on the position and the yaw and their time derivatives
  - This is due to the system underactuation
  - To obtain the desired free attitude, other UAVs as the tilting propellers drones must be employed

- For the quadrotor described bye the RPY dynamic model, the state is $(\zeta, \dot{\zeta})$
- The flat outputs are

$$\sigma = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix}$$

- The relation between the state and the flat outputs is

$$x = \sigma_1$$
$$y = \sigma_2$$
$$z = \sigma_3$$
$$\phi = atan2(\beta_a, \beta_b)$$
$$\theta = atan2(\beta_c, \beta_d)$$
$$\psi = \sigma_4$$

  - $\beta_a = -\ddot{\sigma}_1 \sin \sigma_4 + \ddot{\sigma}_2 \cos \sigma_4$
  - $\beta_b = \sqrt{\beta_c^2 + \beta_d^2}$
  - $\beta_c = -\ddot{\sigma}_1 \cos \sigma_4 + \ddot{\sigma}_2 \sin \sigma_4$
  - $\beta_d = -\ddot{\sigma}_3 + g$

- Then, for the velocity

$$\dot{x} = \dot{\sigma}_1$$
$$\dot{y} = \dot{\sigma}_2$$
$$\dot{z} = \dot{\sigma}_3$$
$$\dot{\psi} = \dot{\sigma}_4$$

- For the roll and pitch velocity, recall $\dot{\eta}_b = Q^{-1}\omega_b^b$ and $\dot{R}_b = R_b S(\omega_b^b) \Longrightarrow S(\omega_b^b) = R_b^T \dot{R}_b$
  - The inverse of the skew-symmetric operator is the vee operator

$$\omega_b^b = S(\omega_b^b)^{\vee}$$

    - Example

$$\omega_b^b = \begin{bmatrix} \omega_{b,x}^b \\ \omega_{b,y}^b \\ \omega_{b,z}^b \end{bmatrix} \text{ from which } S(\omega_b^b) = \begin{bmatrix} 0 & -\omega_{b,z}^b & \omega_{b,y}^b \\ \omega_{b,z}^b & 0 & -\omega_{b,x}^b \\ -\omega_{b,y}^b & \omega_{b,x}^b & 0 \end{bmatrix} \text{ therefore if } S(\omega_b^b) = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & 4 \\ 2 & -4 & 0 \end{bmatrix} \text{ then } \omega_b^b = \begin{bmatrix} -4 \\ -2 \\ 1 \end{bmatrix}$$

  - Then, $\dot{\eta}_b = Q^{-1}\left(R_b^T \dot{R}_b\right)^{\vee}$ by taking the RPY angles from the above expressions

Quadrotor flat outputs

- Finally, for the inputs

$$u_T = m\sqrt{\ddot{\sigma}_1^2 + \ddot{\sigma}_2^2 + (\ddot{\sigma}_3 - g)^2}$$

$$\tau_b^b = I_b(\dot{R}_b^T \dot{R}_b + R_b^T \ddot{R}_b)^{\vee} + R_b^T \dot{R}_b I_b (R_b^T \dot{R}_b)^{\vee}$$

- Therefore, it is reasonable to plan a UAV path/time law on the basis of its flat outputs
- Notice that the dynamics is included within the inputs
  - Therefore, these are valid control inputs as long as we are in ideal conditions and only feed-forward can be sufficient
- It is not possible to plan in advance the quadrotor's attitude, that is instead related to the position
  - This is due because of the underactuation of the quadrotor
  - To obtain the desired free attitude, other UAVs must be considered (tilted or tilting propellers drone)

Hierarchical controller of a quadcopter

- We start from the RPY quadcopter dynamic model, neglecting external disturbances

$$\begin{cases} m\ddot{p}_b = mge_3 - u_T R_b e_3 \quad \textbf{(1)} \\ M(\eta_b)\ddot{\eta}_b = -C(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b \end{cases}$$

- The controller's objective is to obtain two linear subsystems (one for the linear and the other for the angular part) coupled by a non-linear interconnection term
- To this purpose, notice that the angular part is feedback linearizable

$$\tau^b = I_b Q(\eta_b)\tilde{\tau} + Q(\eta_b)^{-T} C(\eta_b, \dot{\eta}_b)\dot{\eta}_b$$

  - $\tilde{\tau} \in \mathbb{R}^3$ is a virtual control input
  - Singularity when $\theta = \pm\frac{\pi}{2}$
- Substituting this control-law within the angular part of the RPY quadcopter dynamic model yields

$$\ddot{\eta}_b = \tilde{\tau}$$

  - $M(\eta_b) = Q^T(\eta_b) I_b Q\ (\eta_b)$ has been exploited

- Consider the tracking error for the attitude

$$e_\eta = \begin{bmatrix} e_\phi \\ e_\theta \\ e_\psi \end{bmatrix} = \eta_b - \eta_{b,d} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} - \begin{bmatrix} \phi_d \\ \theta_d \\ \psi_d \end{bmatrix} \qquad (2)$$

$$\dot{e}_\eta = \begin{bmatrix} \dot{e}_\phi \\ \dot{e}_\theta \\ \dot{e}_\psi \end{bmatrix} = \dot{\eta}_b - \dot{\eta}_{b,d} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \begin{bmatrix} \dot{\phi}_d \\ \dot{\theta}_d \\ \dot{\psi}_d \end{bmatrix}$$

  - $\eta_{b,d}, \dot{\eta}_{b,d}, \ddot{\eta}_{b,d} \in \mathbb{R}^3$ the desired attitude, its desired velocity and acceleration, respectively
    - *How to get the desired orientation if the planner gives us only the desired yaw? Wait some slides, now suppose to have it (!)*
- Consider the following definition, that is is like the virtual desired acceleration oriented as the desired attitude, regulated by the total thrust, plus gravity

$$\mu_d = [\mu_x \quad \mu_y \quad \mu_z]^T = -\frac{1}{m} u_T R_b(\eta_{b,d}) e_3 + g e_3 \ \in \mathbb{R}^3$$

- From (2), we have

$$\eta_b = e_\eta + \eta_{b,d}$$

  - That is $\phi = e_\phi + \phi_d$, $\theta = e_\theta + \theta_d$, and $\psi = e_\psi + \psi_d$
  - We can substitute this expression into $R_b(\eta_b)$ of (1) in the previous slide

Hierarchical controller of a quadcopter

$$(3)\quad \begin{cases} \ddot{p}_b = \mu_d + \dfrac{1}{m} u_T \delta(\eta_{b,d}, e_\eta) \\ \ddot{\eta}_b = \tilde{\tau} \end{cases}$$

- $\delta(\eta_{b,d}, e_\eta)$ is the interconnection term between the linear and the angular part

$$\delta(\eta_{b,d}, e_\eta) = \begin{bmatrix} s_{\phi_d} s_{\psi_d} - s_\phi s_\psi - c_\phi c_\psi s_\theta + c_{\phi_d} c_{\psi_d} s_{\theta_d} \\ c_\psi s_\theta - s_{\phi_d} c_{\psi_d} + c_{\phi_d} s_{\psi_d} s_{\theta_d} - c_\phi c_\psi s_\theta \\ c_{\phi_d} c_{\theta_d} - c_\phi c_\theta \end{bmatrix}$$

- The above expressions have been obtained exploiting this property

$$\sin(\alpha + \beta) = \sin\alpha + 2\sin\frac{\beta}{2}\cos\left(\alpha + \frac{\beta}{2}\right)$$
$$\cos(\alpha + \beta) = \cos\alpha - 2\sin\frac{\beta}{2}\sin\left(\alpha + \frac{\beta}{2}\right)$$

- Now, forgetting for one moment the underactuation, we consider μ_d and $\tilde{\tau}$ our inputs in (3)

- Consider the following definitions

$$e_p = p_b - p_{b,d}$$
$$\dot{e}_p = \dot{p}_b - \dot{p}_{b,d}$$

  - $p_{b,d}, \dot{p}_{b,d}, \ddot{p}_{b,d} \in \mathbb{R}^3$ the desired position, its desired velocity and acceleration, respectively

- It is thus possible to rewrite (3) as follows

$$\textbf{(4)} \quad \begin{cases} \begin{bmatrix} \dot{e}_p \\ \ddot{e}_p \end{bmatrix} = A \begin{bmatrix} e_p \\ \dot{e}_p \end{bmatrix} + B(\mu_d - \ddot{p}_{b,d}) + \frac{1}{m} u_T \Delta \\ \begin{bmatrix} \dot{e}_\eta \\ \ddot{e}_\eta \end{bmatrix} = A \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix} + B(\tilde{\tau} - \ddot{\eta}_{b,d}) \end{cases}$$

  - with

$$A = \begin{bmatrix} O & I \\ O & O \end{bmatrix} \in \mathbb{R}^{6\times 6}, B = \begin{bmatrix} O \\ I \end{bmatrix} \in \mathbb{R}^{6\times 3}, \Delta = \begin{bmatrix} 0 \\ \delta(\eta_{b,d}, e_\eta) \end{bmatrix} \in \mathbb{R}^6$$

  - As said, the inputs for this system are μ_d and $\tilde{\tau}$ that must bring to zero the error for the linear and angular parts

- We can design the following controller

$$(5) \quad \begin{cases} \mu_d = -K_p \begin{bmatrix} e_p \\ \dot{e}_p \end{bmatrix} + \ddot{p}_{b,d} \\ \tilde{\tau} = -K_e \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix} + \ddot{\eta}_{b,d} \end{cases}$$

  - $K_p \in \mathbb{R}^{3\times 6}$ chosen such that $A_p = A - BK_p$ is Hurwitz
  - $K_e \in \mathbb{R}^{3\times 6}$ chosen such that $A_e = A - BK_e$ is Hurwitz
- Substituting (5) into (4) yields the following loop equations

$$\begin{cases} \begin{bmatrix} \dot{e}_p \\ \ddot{e}_p \end{bmatrix} = A_p \begin{bmatrix} e_p \\ \dot{e}_p \end{bmatrix} + \dfrac{1}{m} u_T \Delta \\ \begin{bmatrix} \dot{e}_\eta \\ \ddot{e}_\eta \end{bmatrix} = A_e \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix} \end{cases}$$

  - To show the asymptotic stability of the errors equal to zero, the interconnection term requires the use of the perturbation theory

- Now, we have to solve the pending questions
  - How is it possible to compute the full desired attitude and its time derivatives if the planner give us only the desired position and the desired yaw?
  - How is it possible to compute the desired thrust from μ_d?
- The solution comes from the definition of the flat outputs

$$
\begin{aligned}
u_T &= m\sqrt{{\mu_x}^2 + {\mu_y}^2 + (\mu_z - g)^2} \\
\varphi_d &= \sin^{-1}\left(\frac{m}{u_T}(\mu_y \cos\psi_d - \mu_x \sin\psi_d)\right) \\
\theta_d &= \tan^{-1}\left(\frac{\mu_x \cos\psi_d + \mu_y \sin\psi_d}{\mu_z - g}\right)
\end{aligned}
\qquad (6)
$$

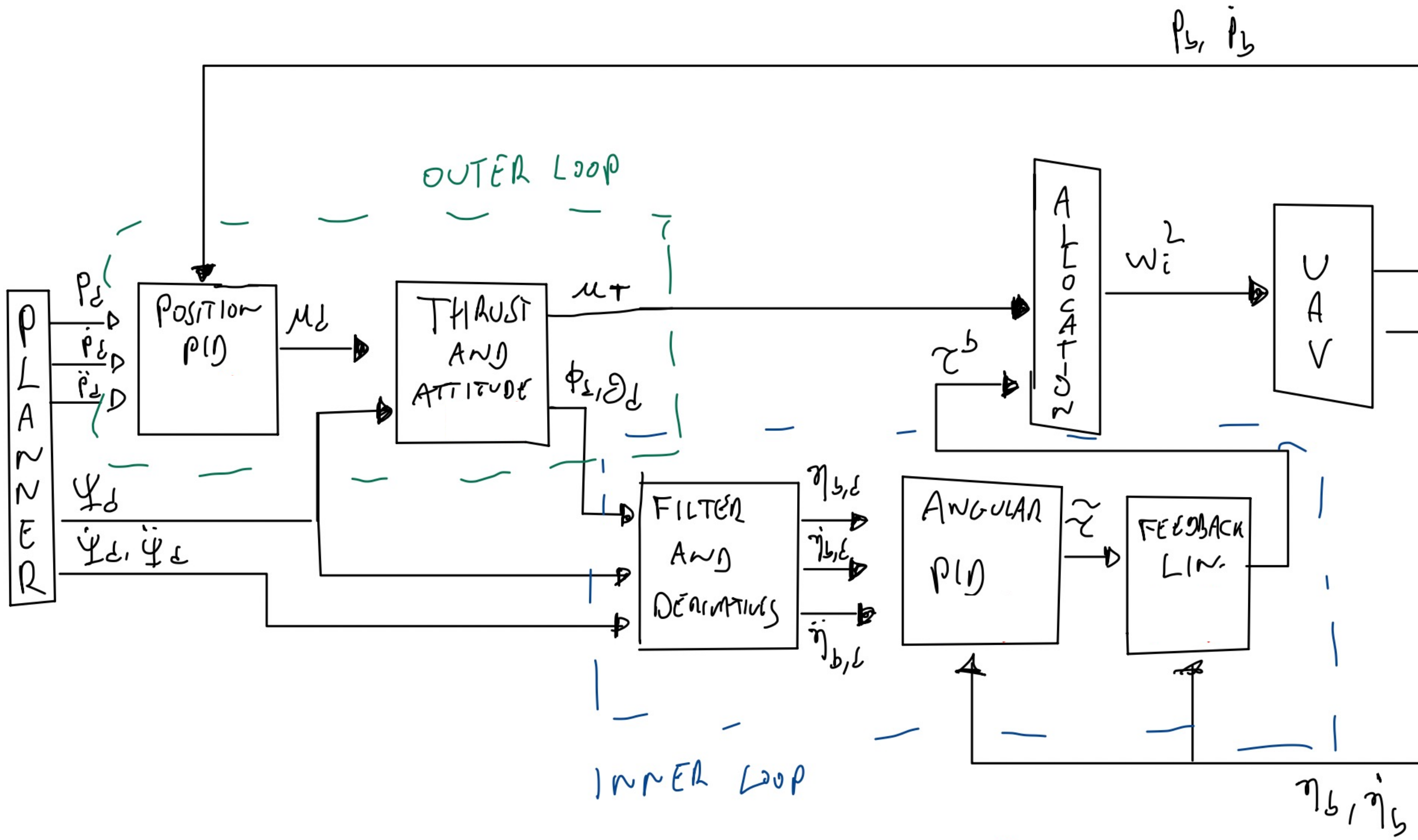
  - Notice that there is a singularity when $\mu_z = g$, therefore the planner should be designed such that $|\mu_z| < g$

Hierarchical controller of a quadcopter

- Outer loop
  - Given the desired linear position, velocity, and acceleration from the planner, it is possible to compute $\mu_d = -K_p \begin{bmatrix} e_p \\ \dot{e}_p \end{bmatrix} + \ddot{p}_{b,d}$
    - This control law can be modified by adding an integral action of the position error without destroying the asymptotic stability property
    - This is needed to add robustness and increase tracking accuracy
  - Then, we find the total thrust and the reference for the roll and the pitch angles
- Inner loop
  - Notice that (6) gives only the roll and the pitch angles
  - In order to retrieve the desired angular velocity and acceleration, we must numerically derivate these values
    - Therefore, a second-order low-pass digital filter should be employed to both reduce noise and compute the time derivatives of such angles
  - After $\tilde{\tau}$, it is possible to compute τ^b from the initial feedback linearization
  - Having the total thrust and τ^b , it is then possible to compute the propellers' velocities
- Considerations
  - Such a two-loop architecture is said to be hierarchical [2]
  - It is also justified because the angular dynamics are faster than linear ones (time-scale separation)
  - The controller is not suitable for acrobatic flights since $\theta \neq \pm\frac{\pi}{2}$ to avoid singularities

[2] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, D. Nakazawa, "Autonomous flying robots. Unmanned aerial vehicles and micro aerial vehicles," Springer, 2010

Hierarchical controller of a quadcopter



Geometric controller of a quadcopter

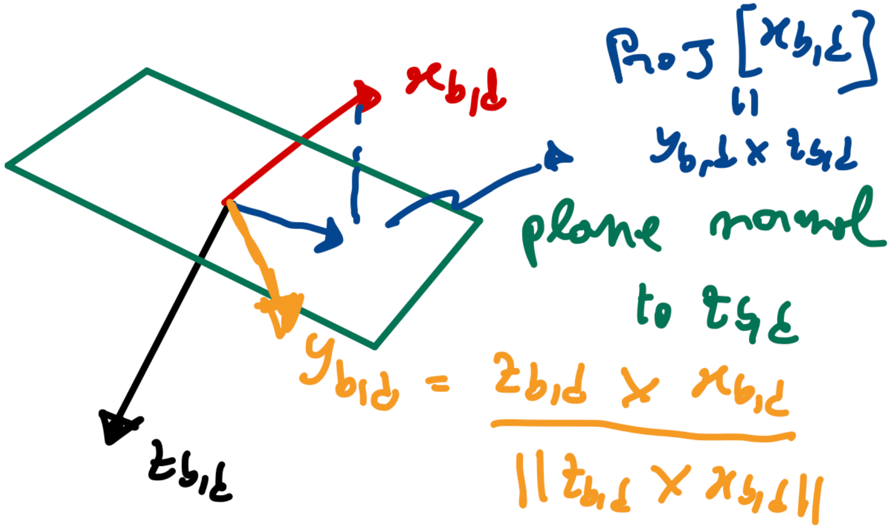
- We start from the coordinate-free quadcopter dynamic model, neglecting external disturbances [3]

$$\begin{cases} m\ddot{p}_b^b = mge_3 - u_T R_b e_3 \\ \dot{R}_b = R_b S(\omega_b^b) \\ I_b \dot{\omega}_b^b = -S(\omega_b^b) I_b \omega_b^b + \tau^b \end{cases}$$

- As in the hierarchical controller, the translational motion is only dependent on the term $u_T R_b e_3$
  - The magnitude of the total thrust, u_T, is directly controlled, but the direction of the total thrust $R_b e_3$ is determined by the third body-fixed axis
  - To change the direction of the total thrust, the attitude should be changed accordingly
- We choose the total thrust, u_T, and the desired reduced attitude, $R_{b,d} e_3$, for the third body-fixed axis such that they stabilize the zero equilibrium of the tracking error for the translational dynamics
- The remaining first two columns of $R_{b,d}$, representing the direction of the body-fixed axes, are chosen to follow the desired direction
  - The control torque, τ^b , is designed to follow the resulting desired attitude $R_{b,d}$ obtained by $x_{b,d}$ and $z_{b,d} = R_{b,d} e_3$

[3] T. Lee, M. Leok, N.H. McClamroch, "Geometric tracking control of quadrotor UAV on SE(3)," 49th IEEE Conference on Decision and Control, 2010

- Once $z_{b,d} = R_{b,d}e_3$ is defined, the desired $x_{b,d}$ might not be orthogonal to it
  - $x_{b,d}$ must be project into the plane orthogonal to $z_{b,d}$



$$R_{b,d} = \begin{bmatrix} S(y_{b,d})z_{b,d} & \underbrace{\frac{S(z_{b,d})x_{b,d}}{\|S(z_{b,d})x_{b,d}\|}}_{y_{b,d}} & z_{b,d} \end{bmatrix}$$

Geometric controller of a quadcopter

- Controller's job
  - $p_b \rightarrow p_{b,d}$
  - $\dot{p}_b \rightarrow \dot{p}_{b,d}$
  - $Proj[x_b] \rightarrow Proj[x_{b,d}]$
- Tracking errors for the translational motion

$$e_p = p_b - p_{b,d}, \dot{e}_p = \dot{p}_b - \dot{p}_{b,d}$$

- Tracking errors for the angular motion

$$e_R = \frac{1}{2}(R_{b,d}^T R_b - R_b^T R_{b,d})^{\vee}$$
$$e_\omega = \omega_b^b - R_b^T R_{b,d} \omega_{b,d}^{b,d}$$

  - The angular error is defined in SO(3)
  - The angular velocity error can be seen as the velocity of $R_{b,d}^T R_b$ in the body frame (see [3])

Geometric controller of a quadcopter

- Assumption

$$\left\|-mge_3 + m\ddot{p}_{b,d}\right\| < \text{positive constant}$$

  - This ensures that the planned desired linear acceleration does not exceed the gravity acceleration

- Outer loop control

$$u_T = -(-K_p e_p - K_v \dot{e}_p - mge_3 + m\ddot{p}_{b,d})^T R_b e_3$$

$$z_{b,d} = -\frac{-K_p e_p - K_v \dot{e}_p - mge_3 + m\ddot{p}_{b,d}}{\left\|-K_p e_p - K_v \dot{e}_p - mge_3 + m\ddot{p}_{b,d}\right\|}$$

- Inner loop control (after retrieved $R_{b,d}$)

$$\tau^b = -K_R e_R - K_\omega e_\omega + S(\omega_b^b) I_b\, \omega_b^b - I_b\left(S\left(\omega_b^b\right) R_b^T R_{b,d} \omega_{b,d}^{b,d} - R_b^T R_{b,d} \dot{\omega}_{b,d}^{b,d}\right)$$

- Closed−loop system

$$m\ddot{e}_p + K_v\dot{e}_p + K_p e_p = \underbrace{-\frac{u_T}{e_3^T R_{b,d}^T R_b e_3}\left[\left(e_3^T R_{b,d}^T R_b e_3\right) R_b e_3 - R_{b,d} e_3\right]}_{X}$$

Plays the role of the interconnection term as seen in the hierarchical control, and it is zero when the orientation error is zero

$$I_b \dot{e}_\omega + K_\omega e_\omega + K_R e_R = 0$$

- Notice that only Lyapunov theory can give us the exponential convergence of the error to zero
  - This is true if the initial attitude error is less then 90°

- Sketch of how to obtain the closed-loop system (angular part)
  - Recall the definition of the angular velocity error

$$e_\omega = \omega_b^b - R_b^T R_{b,d}\,\omega_{b,d}^{b,d}$$

  - Taking its time derivative yields

$$\dot{e}_\omega = \dot{\omega}_b^b - \frac{d}{dt}\left(R_b^T R_{b,d}\,\omega_{b,d}^{b,d}\right)$$

  - Multiply both sides by I_b, we have $I_b \dot{e}_\omega = I_b \dot{\omega}_b^b - I_b \frac{d}{dt}\left(R_b^T R_{b,d}\,\omega_{b,d}^{b,d}\right)$
  - Substituting the system angular dynamics in the purple term above yields

$$I_b \dot{e}_\omega = -S\left(\omega_b^b\right) I_b \omega_b^b + \tau^b - I_b \frac{d}{dt}\left(R_b^T R_{b,d}\,\omega_{b,d}^{b,d}\right)$$

  - Notice that

$$\frac{d}{dt}\left(R_b^T R_{b,d}\,\omega_{b,d}^{b,d}\right) = R_b^T R_{b,d}\,\dot{\omega}_{b,d}^{b,d} + \dot{R}_b^T R_{b,d}\,\omega_{b,d}^{b,d} + R_b^T \dot{R}_{b,d}\,\omega_{b,d}^{b,d}$$

  - Recalling that $\dot{R}_{b,d} = R_{b,d} S\left(\omega_{b,d}^{b,d}\right)$, then $\dot{R}_b^T = -S\left(\omega_b^b\right) R_b^T$, then

$$\frac{d}{dt}\left(R_b^T R_{b,d}\,\omega_{b,d}^{b,d}\right) = R_b^T R_{b,d}\,\dot{\omega}_{b,d}^{b,d} - S\left(\omega_b^b\right) R_b^T R_{b,d}\,\omega_{b,d}^{b,d} + R_b^T R_{b,d}\underbrace{S\left(\omega_{b,d}^{b,d}\right)\omega_{b,d}^{b,d}}_{0}$$

  - Finally

$$I_b \dot{e}_\omega = -S\left(\omega_b^b\right) I_b \omega_b^b + \tau^b - I_b\left(R_b^T R_{b,d}\,\dot{\omega}_{b,d}^{b,d} - S\left(\omega_b^b\right) R_b^T R_{b,d}\,\omega_{b,d}^{b,d}\right)$$

- Substituting $\tau^b = -K_R e_R - K_\omega e_\omega + S(\omega_b^b) I_b\, \omega_b^b - I_b (S(\omega_b^b) R_b^T R_{b,d} \omega_{b,d}^{b,d} - R_b^T R_{b,d} \dot{\omega}_{b,d}^{b,d})$ into the previous expression yields

$$I_b \dot{e}_\omega = \cancel{-S(\omega_b^b) I_b \omega_b^b} + -K_R e_R - K_\omega e_\omega + \cancel{S(\omega_b^b) I_b\, \omega_b^b} - I_b (\cancel{S(\omega_b^b) R_b^T R_{b,d} \omega_{b,d}^{b,d}} - \cancel{R_b^T R_{b,d} \dot{\omega}_{b,d}^{b,d}}) - I_b (\cancel{R_b^T R_{b,d} \dot{\omega}_{b,d}^{b,d} - S(\omega_b^b) R_b^T R_{b,d} \omega_{b,d}^{b,d}})$$

- We then obtain $I_b \dot{e}_\omega + K_\omega e_\omega + K_R e_R = 0$
- Since e_ω is not the time derivative of e_R, the proof of exponential stability for the above closed-loop system is obtained from Lyapuonov theroy

- Sketch of how to obtain the closed-loop system (linear part)
  - Recall the definition of the position velocity error $\dot{e}_p = \dot{p}_b - \dot{p}_{b,d}$, it is possible to write $\ddot{e}_p = \ddot{p}_b - \ddot{p}_{b,d}$
  - Multiply both sides by the mass of the quadcopter, $m\ddot{e}_p = m\ddot{p}_b - m\ddot{p}_{b,d}$
  - Substituting the system linear dynamics in the purple term above yields

$$m\ddot{e}_p = mge_3 - u_T R_b e_3 - m\ddot{p}_{b,d}$$

  - Add and subtract the term $\frac{u_T}{e_3^T R_{b,d}^T R_b e_3} R_{b,d} e_3$

$$m\ddot{e}_p = mge_3 - u_T R_b e_3 - m\ddot{p}_{b,d} + \frac{u_T}{e_3^T R_{b,d}^T R_b e_3} R_{b,d} e_3 - \frac{u_T}{e_3^T R_{b,d}^T R_b e_3} R_{b,d} e_3$$

    - $R_b e_3 = z_b$
    - $e_3^T R_{b,d}^T = z_{b,d}^T$
    - $e_3^T R_{b,d}^T R_b e_3$ is the cosine angle between z_b and $z_{b,d}$

- Recall the term $X = \frac{u_T}{e_3^T R_{b,d}^T R_b e_3}\left[\left(e_3^T R_{b,d}^T R_b e_3\right) R_b e_3 - R_{b,d} e_3\right]$, the previous expression can be written as

$$m\ddot{e}_p = mge_3 - m\ddot{p}_{b,d} - \frac{u_T}{e_3^T R_{b,d}^T R_b e_3} R_{b,d} e_3 - X$$

- Let $A = -K_p e_p - K_v \dot{e}_p - mge_3 + m\ddot{p}_{b,d}$ be the desired control force for the translational dynamics, then the total thrust should be taken as $u_T = -A^T R_b e_3$
- However, $A = \|A\| R_{b,d} e_3$, since it is directed along the z −axis of the desired body frame, and therefore $u_T = -\|A\| \left(R_{b,d} e_3\right)^T R_b e_3$
- Substituting the total thrust into the above expression yields

$$m\ddot{e}_p = mge_3 - m\ddot{p}_{b,d} + \frac{\cancel{\|A\| \left(R_{b,d} e_3\right)^T R_b e_3}}{\cancel{e_3^T R_{b,d}^T R_b e_3}} \frac{A}{\cancel{\|A\|}} - X$$

- We get

$$m\ddot{e}_p = mge_3 - m\ddot{p}_{b,d} + A - X$$

- That is

$$m\ddot{e}_p = \cancel{mge_3 - m\ddot{p}_{b,d}} - K_p e_p - K_v \dot{e}_p \cancel{- mge_3 + m\ddot{p}_{b,d}} - X$$

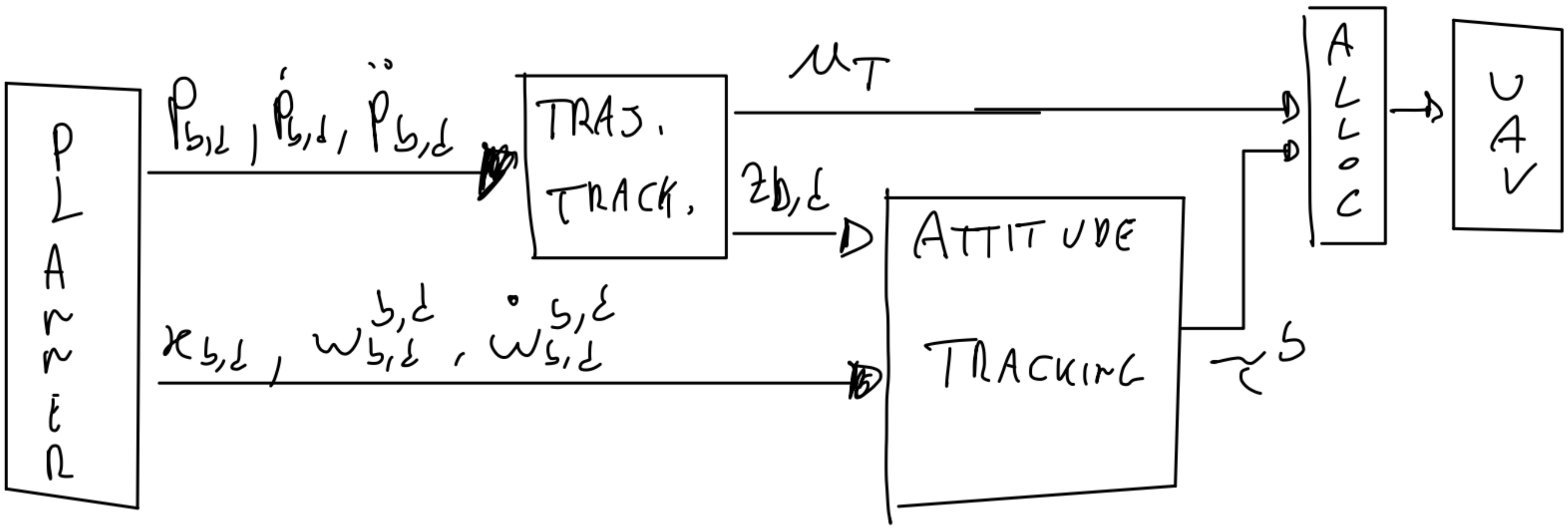
- We then obtain $m\ddot{e}_p + K_v \dot{e}_p + K_p e_p = -X$
- The interconnection term X plays the role of the interconnection term already seen within the hierarchical controller
- Notice that if $R_{b,d} = R_b$, then $X = 0$
- Lyapunov and perturbation theory give us conditions on the boundness of $\|e_p\|$ and $\|\dot{e}_p\|$

Geometric controller of a quadcopter

- Properties
  - One of the unique properties of the presented geometric controller is that it is entirely designed in the model-configuration space $\mathbb{R}^3 \times SO(3)$ using rotation matrices, avoiding any representation singularities
  - Exponential stability of the closed-loop system can be proven if the initial attitude error is less than 90^o
  - The total thrust can be written as

$$u_T = -\|A\|\left(R_{b,d}e_3\right)^T R_b e_3 = -\|A\| z_{b,d}{}^T z_b$$

    - It contains the scalar product between $z_{b,d}$ and z_b
    - If we have a big error in the attitude, this results in a big angle between these two axes; however, in this case, the total thrust decreases in magnitude since the dot product decreases at big angles
  - Therefore, this controller is designed such that the position tracking error converges to zero when there is no attitude error, and it is limited for non-zero attitude tracking errors to achieve asymptotic stability of the complete closed-loop dynamics


PLANNER
p b,d , ṗ b,d , p̈ b,d
TRAJ. TRACK.
u T
z b,d
ATTITUDE TRACKING
x b,d , ω b,d , ω̇ b,d
τ b
ALLOC
UAV

- Consider the RPY quadrotor dynamic model and, now, external wrench disturbance

$$\begin{cases} m\ddot{p}_b = mge_3 - u_T R_b e_3 + f_e \\ M(\eta_b)\ddot{\eta}_b = -C(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b + \tau_e \end{cases}$$

- Consider the generalised momentum vector $q \in \mathbb{R}^6$

$$q = \begin{bmatrix} mI_3 & O_3 \\ O_3 & M(\eta_b) \end{bmatrix} \begin{bmatrix} \dot{p}_b \\ \dot{\eta}_b \end{bmatrix}$$

  - I_X and O_X are identity and zero matrices of proper dimensions
- The time derivative of the momentum is

$$\dot{q} = \begin{bmatrix} mge_3 - u_T R_b e_3 + f_e \\ C^T(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b + \tau_e \end{bmatrix}$$

  - The notable property $\dot{M}(\eta_b) = C^T(\eta_b, \dot{\eta}_b) + C(\eta_b, \dot{\eta}_b)$ has been exploited

Estimator-based control

- Estimator's goal
  - Estimate the wrench $\begin{bmatrix} f_e \\ \tau_e \end{bmatrix}$ through $\begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix}$
- To simplify things, we want a linear relationship between the external wrench and its estimation in the Laplace domain

$$\mathcal{L}\left[\begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix}\right] = \mathrm{G(s)}\mathcal{L}\left[\begin{bmatrix} f_e \\ \tau_e \end{bmatrix}\right]$$

  - $G(s) \in \mathbb{C}^{6\times 6}$ is a diagonal matrix of transfer functions
- To further simplify, let's choose first-order transfer functions

$$G_i(s) = \frac{k_o}{s + c_o}, i = 1, \cdots, 6$$

  - $k_o, c_o > 0$
  - What is the best choice for the gains?
    - $c_o \to +\infty$ … it is impossible: a compromise must be found
    - $k_o = c_o$ not to have any gain in the estimation

- Apply the inverse of the Laplace operator to $\mathcal{L}\left[\begin{bmatrix}\hat{f}_e \\ \hat{\tau}_e\end{bmatrix}\right] = \mathrm{G}(s)\mathcal{L}\left[\begin{bmatrix}f_e \\ \tau_e\end{bmatrix}\right]$, with the hypotheses that $\begin{bmatrix}\hat{f}_e(0) \\ \hat{\tau}_e(0)\end{bmatrix} = 0$ and $q(0) = 0$
  - This means that the estimation should start before the UAV's take-off

$$\frac{d}{dt}\begin{bmatrix}\hat{f}_e(t) \\ \hat{\tau}_e(t)\end{bmatrix} = k_o\begin{bmatrix}f_e(t) \\ \tau_e(t)\end{bmatrix} - k_o\begin{bmatrix}\hat{f}_e(t) \\ \hat{\tau}_e(t)\end{bmatrix}$$

- Recall

$$\dot{q} = \begin{bmatrix} mge_3 - u_T R_b e_3 + f_e \\ C^T(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b + \tau_e \end{bmatrix} \Rightarrow \begin{bmatrix}f_e \\ \tau_e\end{bmatrix} = \dot{q} - \begin{bmatrix} mge_3 - u_T R_b e_3 \\ C^T(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b \end{bmatrix}$$

- Therefore, folding the last expression into the purple equation before and integrating yield

$$\begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix} = k_o \left(q - \int_0^t \begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix} + \begin{bmatrix} mge_3 - u_T R_b e_3 \\ C^T(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b \end{bmatrix} dt \right)$$

- The necessary quantities to estimate the external wrench are
  - Attitude → IMU
  - Angular velocity → IMU
  - Linear velocity → GPS/on-bord devices
  - Inputs → Employed controller
  - Model parameters → System modelling
- The integral of the estimator is approximated numerically
- The estimation can be further filtered (e.g., Butterworth filters)
  - Higher-order transfer functions can be used
- The same estimator can be obtained from the coordinate-free dynamic model

- How can we use the estimation inside the hierarchical controller?
- We can do the following modifications to the equations above
  - Partial feedback linearization

$$\tau^b = I_b Q(\eta_b)\tilde{\tau} + Q^{-T}(\eta_b)C(\eta_b, \dot{\eta}_b)\dot{\eta}_b - Q^{-T}(\eta_b)\hat{\tau}_e$$

  - Definition

$$\mu_d = [\mu_x \quad \mu_y \quad \mu_z]^T = -\frac{1}{m}u_T R_b(\eta_b)e_3 + g e_3 + \frac{1}{m}\hat{f}_e$$

  - Thrust and desired roll-pitch computation

$$\bar{\mu} = [\bar{\mu}_x \quad \bar{\mu}_y \quad \bar{\mu}_z]^T = \mu - \frac{1}{m}\hat{f}_e$$

$$u_T = m\sqrt{\bar{\mu}_x{}^2 + \bar{\mu}_y{}^2 + (\bar{\mu}_z - g)^2}$$

$$\varphi_d = \sin^{-1}\left(\frac{m}{u_T}(\bar{\mu}_y \cos\psi_d - \bar{\mu}_x \sin\psi_d)\right)$$

$$\theta_d = \tan^{-1}\left(\frac{\bar{\mu}_x \cos\psi_d + \bar{\mu}_y \sin\psi_d}{\bar{\mu}_z - g}\right)$$

- It is possible to develop the same estimator starting from the coordinate-free dynamic model of a UAV and, with suitable modifications, insert the estimation within the geometric controller

- Second-order estimator

$$G_i(s) = \frac{\omega_{n,i}^2}{s^2 + 2\zeta_i\omega_{n,i}s + \omega_{n,i}^2}, i = 1, \cdots, 6$$

$$\frac{d^2}{dt^2}\begin{bmatrix}\hat{f}_e(t)\\ \hat{\tau}_e(t)\end{bmatrix} + K_2\frac{d}{dt}\begin{bmatrix}\hat{f}_e(t)\\ \hat{\tau}_e(t)\end{bmatrix} + K_2K_1\begin{bmatrix}\hat{f}_e(t)\\ \hat{\tau}_e(t)\end{bmatrix} = K_2K_1\begin{bmatrix}f_e(t)\\ \tau_e(t)\end{bmatrix}$$

$$\begin{bmatrix}\hat{f}_e\\ \hat{\tau}_e\end{bmatrix} = K_2\left(\int_0^t -\begin{bmatrix}\hat{f}_e\\ \hat{\tau}_e\end{bmatrix} + K_1\left(q - \int_0^t \begin{bmatrix}\hat{f}_e\\ \hat{\tau}_e\end{bmatrix} + \begin{bmatrix}mge_3 - u_T R_b e_3\\ C^T(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b\end{bmatrix} dt\right) dt\right)$$

$$K_2 = 2\zeta_i\omega_{n,i}I_6$$

$$K_2K_1 = \omega_{n,i}^2 I_6$$

- r^{th}-order estimator

$$G_i(s) = \frac{c_0}{s^r + c_{r-1}s^{r-1} + \cdots + c_1 s + c_0}, i = 1, \cdots, 6$$

  - Hurwitz polynomial

$$\sum_{i=0}^{r} \prod_{j=-r}^{-(i+1)} K_{-j} \frac{d^i}{dt^i} \begin{bmatrix} \hat{f}_e(t) \\ \hat{\tau}_e(t) \end{bmatrix} = \prod_{i=1}^{r} K_i \begin{bmatrix} f_e(t) \\ \tau_e(t) \end{bmatrix}$$

$$\gamma_1(t) = K_1 \left(q - \int_0^t \begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix} + \begin{bmatrix} mge_3 - u_T R_b e_3 \\ C^T(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b \end{bmatrix} dt \right)$$

$$\gamma_i(t) = K_i \int_0^t - \begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix} + \gamma_{i-1}\, dt, \qquad i = 2, \cdots, r$$

$$\prod_{i=j+1}^{r} K_i = c_j, \qquad j = 0, \cdots, r-1$$

Passivity-based controller with compensation of external wrench

- The main objective of this controller is to avoid the feedback linearization of the angular part
  - At the same time, the estimation of the previously presented estimator is employed to compensate for the unknown terms (external disturbances + unmodeled dynamics)
- We consider the RPY quadrotor dynamic model

$$\begin{cases} m\ddot{p}_b = mge_3 - u_T R_b e_3 + f_e \\ M(\eta_b)\ddot{\eta}_b = -C(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b + \tau_e \end{cases}$$

  - As usual, the planner gives the desired trajectory for the position components and the yaw angle

Passivity-based controller with compensation of external wrench

- Define the following reference quantities

$$\dot{\eta}_r = \dot{\eta}_{b,d} - \sigma e_\eta$$
$$\ddot{\eta}_r = \ddot{\eta}_{b,d} - \nu \dot{e}_\eta$$
$$e_\eta = \eta_b - \eta_{b,d}$$
$$\dot{e}_\eta = \dot{\eta}_b - \dot{\eta}_{b,d}$$
$$\nu_\eta = \dot{e}_\eta + \sigma e_\eta$$

  - $\sigma > 0$ a positive coupling parameter
- The following passivity-based control can be defined for the inner (angular) loop

$$\tau^b = Q^{-T}\left(M(\eta_b)\ddot{\eta}_r + C(\eta_b, \dot{\eta}_b)\dot{\eta}_r - \hat{\tau}_e - D_o \nu_\eta - K_o e_\eta\right)$$

  - $D_o, K_o \in \mathbb{R}^{3\times 3}$ positive definite gain matrices

Passivity-based controller with compensation of external wrench

- For the outer loop, define the following errors

$$e_p = p_b - p_{b,d}$$
$$\dot{e}_p = \dot{p}_b - \dot{p}_{b,d}$$
$$\ddot{e}_p = \ddot{p}_b - \ddot{p}_{b,d}$$
$$\mu_d = [\mu_x \quad \mu_y \quad \mu_z]^T = -\frac{1}{m} u_T R_b(\eta_{b,d}) e_3 + g e_3 + \frac{1}{m} \hat{f}_e$$

- Replacing $\eta_b = e_\eta + \eta_{b,d}$ and defining $\tilde{f} = f_e - \hat{f}_e$ the linear estimation error yield

$$\ddot{p}_b = \mu_d + \frac{1}{m} u_T \delta + \frac{1}{m} \tilde{f}$$

  - With δ the interconnection term as for the hierarchical controller
  - As for the hierarchical controller, we forget for one moment the underactuation and we consider μ_d our input

$$\mu_d = -K_d \dot{e}_p - K_p e_p + \ddot{p}_{b,d}$$

Passivity-based controller with compensation of external wrench

- The closed-loop equations are

$$m\ddot{e}_p + K_d\dot{e}_p + K_p e_p = u_T\delta + \tilde{f}$$
$$M\dot{v}_\eta + (C + D_o)v_\eta + K_o e_\eta = \tilde{\tau}$$

  - The right side of the linear closed-loop equation acts like an external force
    - It depends on both the quadrotor's attitude error and the estimation error
    - Boundness of the errors can be proven through Lyapunov and perturbation theories
  - The right side of the angular closed-loop equation acts like an external disturbance of the mass-damping-spring system of the left side
    - This disturbance is given by the estimation error of the angular part
    - This equation can be proven to establish a passive relationship between $\tilde{\tau}$ and v_η
  - Both closed-loop equations can be seen as mass-damping-spring systems with programmable stiffness, partially-programmable damping and given mass and inertia
    - K_p and K_d have physical meanings
    - Recalling that the linear part is the slowest, the closed-loop bandwidth for the linear part is defined
    - Upon this, the estimator's bandwidth can be tuned: it must be faster than the closed-loop linear dynamics (larger bandwidth)
    - K_o and D_o can be finally tuned
    - The coupling factor, σ, should be properly tuned since it affects the tracking results dramatically
      - The robot may vibrate for small values of σ, while larger values improve tracking
      - A common choice is $K_o = \sigma D_o$ that is equivalent to write a quadratic optimization problem

Passivity-based controller with compensation of external wrench

- The total thrust can be computed from the desired acceleration vector as usual

$$\bar{\mu} = [\bar{\mu}_x \quad \bar{\mu}_y \quad \bar{\mu}_z]^T = \mu_d - \frac{1}{m}\hat{f}_e$$

$$u_T = m\sqrt{\bar{\mu}_x{}^2 + \bar{\mu}_y{}^2 + (\bar{\mu}_z - g)^2}$$

$$\varphi_d = \sin^{-1}\left(\frac{m}{u_T}(\bar{\mu}_y \cos\psi_d - \bar{\mu}_x \sin\psi_d)\right)$$

$$\theta_d = \tan^{-1}\left(\frac{\bar{\mu}_x \cos\psi_d + \bar{\mu}_y \sin\psi_d}{\bar{\mu}_z - g}\right)$$

  - Notice that the condition $\bar{\mu} = \mu_d - \frac{1}{m}\hat{f}_e = g e_3$ should be avoided as usual

- The proposed controller can be employed without the estimator of external forces following the same passages for the linear part as for the hierarchical controller

Introduction

- UAVs and UAMs are usually involved in applications that require flying close to different structures, objects, and obstacles
- These devices can also grasp objects or make contacts with the environment for inspection tasks
- In all cases, the airflow produced by the rotors is constrained by the environment
  - This results in a change of the total thrust and torque

Ground effect for the single propeller

- This is the effects that a UAV is subject to when it flies closes to the ground
- The most employed theoretical result for the ground effect is based on the so-called potential aerodynamic assumption
  - This assumption considers the fluid
    - Inviscid
    - Incompressible
    - Irrotational
    - Steady
- In order to model the effect of a surface like the ground, the method of images is employed
  - This method consists in placing a virtual rotor on the opposite side of the surface and at the same distance
    - If one rotor is placed at $[0 \quad 0 \quad h]^T$, the other rotor is placed at $[0 \quad 0 \quad -h]^T$

Ground effect for the single propeller

- Let v_{IGE} and T_{IGE} be the induced velocity and the thrust of the rotor inside the ground effect, respectively
- Let v_{OGE} and T_{OGE} be the induced velocity and the thrust of the rotor outside the ground effect, respectively
- From power preserving theory, the following relation holds

$$v_{IGE}T_{IGE} = v_{OGE}T_{OGE}$$

- From the previous one and from aerodynamic considerations

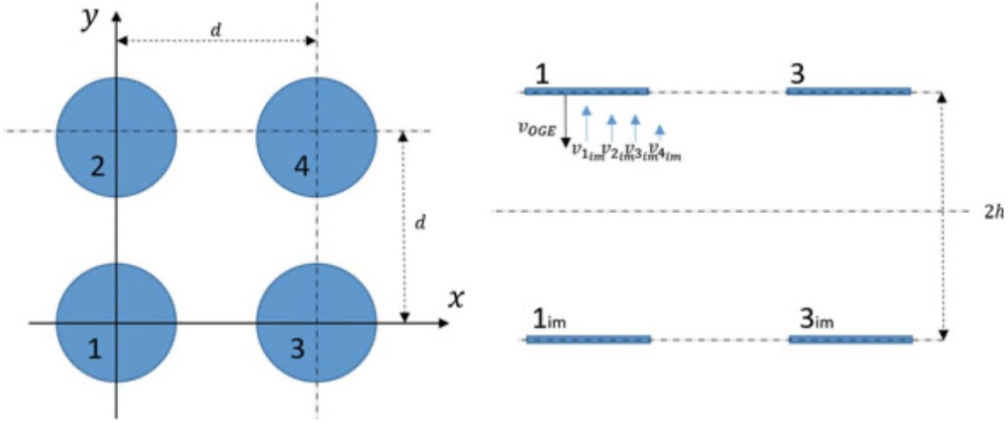
$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1 - \left(\frac{\rho}{4z}\right)^2}$$

  - With ρ the radius of the propeller, and z the height of the propeller from the ground
  - Notice that for $\frac{z}{\rho} = 2$, the ratio is 1.016
    - Therefore, this formula predicts that the ground effect is negligible when the rotor is more than one diameter off the ground

$$\frac{z}{\rho} > 2 \Longrightarrow z > 2\rho$$

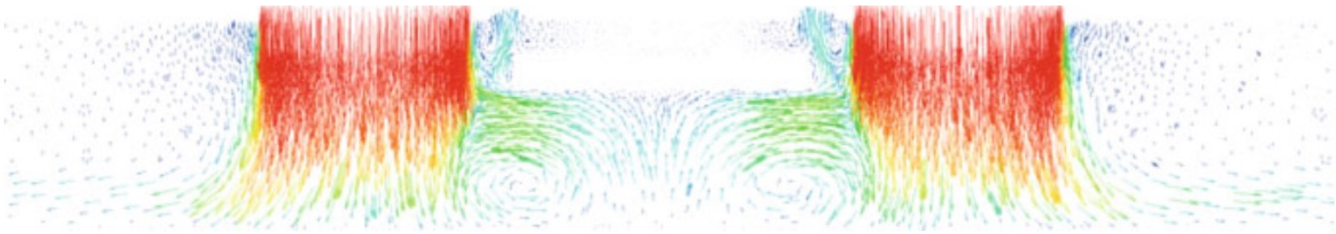
- In the quadrotor case, each propeller has an image source of equal strength placed below the correspondent rotor at a distance equal to twice the height of the rotor
- It is necessary to re-write the expression of the velocity induced inside the ground effect to take into account the effects of the other rotors
- Supposing that the power remains constant for the complete multirotor, the following expression holds

$$\frac{u_{T,IGE}}{u_{T,OGE}} = \frac{1}{1 - \left(\frac{\rho}{4z}\right)^2 - \frac{z\rho^2}{\sqrt{(d^2+4z^2)^3}} - \frac{z\rho^2}{2\sqrt{(2d^2+4z^2)^3}}}$$



Ground effect for the quadrotor

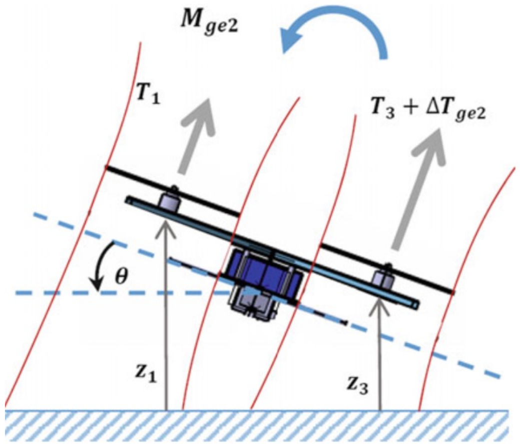
- This picture shows a computational fluid dynamics (CFD) simulation of two rotors close to the ground



- The simulation shows that the airflow in the middle of the space between the two rotors is reverted after hitting the ground
- The airflow hits the multirotor frame producing and extra lifting force

Ground effect for the quadrotor

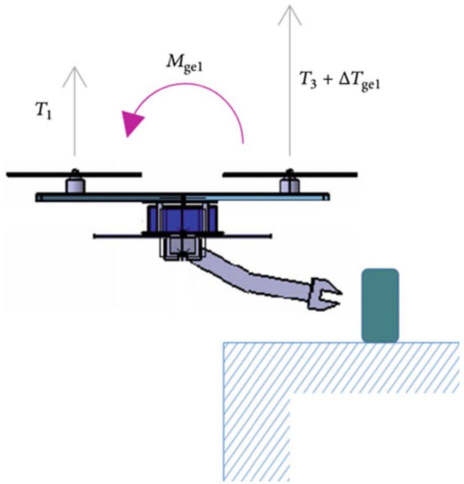
- This picture shows the hovering multirotor subject to an attitude perturbation



- Since the increment in the rotor thrust depends on the distance of each rotor to the ground, and since the distances above are different, a disturbance moment, M_{ge}, will appear
  - In the condition above, this disturbance indeed tries to stabilize the vehicle

Ground effect for the quadrotor

- This picture shows a UAM hovering or it is flying at low speed but only a part of the rotors are subject to the ground effect



- This asymmetrical problem produces a different aerodynamic effect in each rotor
- In the figure, the moment M_{ge} will arise and will try to rotate the multirotor, preventing the grasp of the object

Ceiling effect

- This effect arises when a multirotor needs to approach some objects from below (i.e., bridges, tunnels, …)
- In the single rotor case, it happens that the thrust increases significantly closes to the ceil, pushing the rotor even closer to it
  - Therefore, it is important to take into account such an effect to avoid crashes
  - The increased thrust is a consequence of the increase of the propeller rotational velocity, caused by the vacuum effect which decreases the propeller drag when it is very close to the ceil, so that the propeller rotates faster
- The effect can be approximated by the following formula

$$\frac{T_{ICE}}{T_{OCE}} = \frac{1}{1 - \frac{1}{k_1}\left(\frac{\rho}{z + k_2}\right)^2}$$

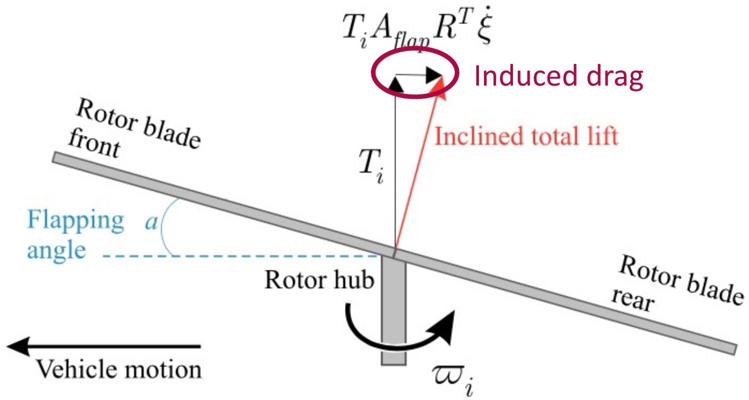
  - T_{ICE} and T_{OCE} are the single thrust inside and outside the ceiling effect, respectively
  - k_1, k_2 are parameters to tune
- Therefore, through this effect, it is possible to develop more thrust for the same spent power
  - This results in less energy consumption and more flight time (if crashes are avoided …)
  - This energy saving result is particularly interesting in those applications requiring to maintain a contact with the ceiling

Wall and pipe effects

- Wall effect
  - This effect appears when the UAV is flying close to a wall
  - In case of flat UAVs (fixed propellers), the wall effect is negligible since the propellers and the wall are orthogonal
  - This may be not true in case of tilted or tilting UAVs
- Pipe effect
  - In oil and gas refineries, the environments cluttered and full of pipelines
  - Flying over a pipe, or close to a pipe, or approaching a pipe from below have the same effects of the ground, wall, and ceil effects, respectively
  - However, the size and the shape of the pipes may mitigate a lot these effects

Flapping dynamics

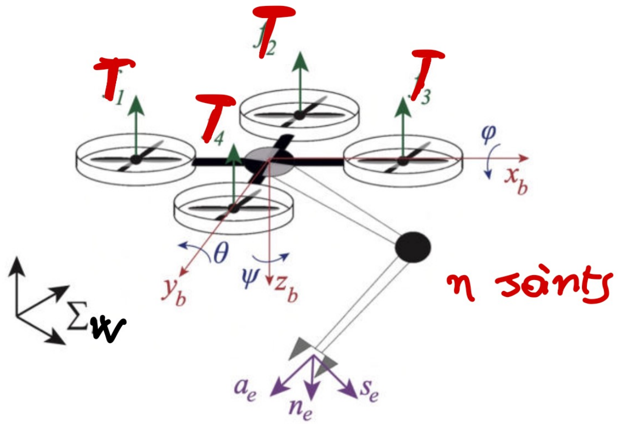
- During the flight, the high velocities of the propellers causes the flapping of the blades
  - Flapping means an up-and-down oscillation due to some elasticity components of the blade
- The flapping causes a different drag force that is often omitted or neglected
  - The reason is that these forces are dissipative and contribute to the stability of the vehicle during hovering
- Besides, since these drag forces are equally distributed in hovering, they can be responsible for the aerodynamic torque causing the yaw
- However, during a trajectory, these forces may become a disturbance
- Aerodynamic drag due to blade flapping affects the UAV during the forward flight
  - In detail, the advancing rotor blade has a higher velocity then the rear one and it will generate e lift force
  - In turn, this creates a bigger induced drag than the rear blade
  - This results in a net force that is opposite to the current forward motion



Modelling of an aerial manipulator

- Let be the following vector the generalised joint vector

$$\zeta = \begin{bmatrix} p_b \\ \eta_b \\ q_1 \\ \vdots \\ q_n \end{bmatrix} \in \mathbb{R}^{6+n}$$



- Let consider the kinematics of the robot with respect to the body frame of the UAV and then expressed in the world fixed frame

$$T_e = T_b T_e^b$$

  - $T_e \in SE(3) = \mathbb{R}^3 \times SO(3)$ the end-effctor frame in the world frame
  - $T_b = \begin{bmatrix} R_b & p_b \\ 0^T & 1 \end{bmatrix} \in SE(3)$ the UAV body frame in the world frame
  - $T_e^b \in SE(3)$ the end-effector frame in the UAV body frame

- The differential kinematics is

$$\begin{bmatrix} \dot{p}_b \\ \omega_b^b \\ \dot{p}_e^b \\ \omega_e^b \end{bmatrix} = \begin{bmatrix} I_3 & O_3 & O \\ O_3 & Q(\eta_b) & O \\ O & O & J(q) \end{bmatrix} \dot{\zeta}$$

  - $\dot{p}_e^b \in \mathbb{R}^3$ the linear velocity of the end-effector in the body frame
  - $\omega_e^b \in \mathbb{R}^3$ the angular velocity of the end−effector in the body frame
- Now, put a frame at each link's centre of mass
  - Let $p_{l_i} \in \mathbb{R}^3$ be the position of the center of mass of the i-th link in the world fixed frame

$$p_{l_i} = p_b + R_b p_{bl_i}^b, \qquad i = 1, \dots, n$$

    - $p_{bl_i}^b \in \mathbb{R}^3$ is the position of the center of mass of the link in the body frame

- The following expressions hold for $i = 1, \ldots, n$

$$\dot{p}_{bl_i} = J_p^{(l_i)} \dot{q}$$

$$\omega_{bl_i} = J_o^{(l_i)} \dot{q}$$

- Therefore, deriving p_{l_i} and ω_{l_i} with respect to time yields

$$\dot{p}_{l_i} = \dot{p}_b - S(R_b p_{bl_i})\omega_b + R_b J_p^{(l_i)} \dot{q}$$

$$\omega_{l_i} = \omega_b - R_b J_o^{(l_i)} \dot{q}$$

Modelling of an aerial manipulator

- The dynamic model can be derived through the Euler-Lagrangian equations, considering the generic element of the generalised joint vector, ζ, and the associated generalised force, u_i
- The Lagrangian is $\mathcal{L} = K - U$
- The kinetic energy is $K = K_b + \sum_{i=1}^{n} K_{l_i}$
    - $K_b = \frac{1}{2} m \dot{p}_b^T p_b + \frac{1}{2} \dot{\eta}_b^T Q^T I_b Q \dot{\eta}_b$ is the UAV kinetic energy
    - $K_{l_i} = \frac{1}{2} m_{l_i} \dot{p}_{l_i}^T p_{l_i} + \frac{1}{2} \omega_{l_i}^T R_b R_{l_i}^b H_{l_i} R_b^{l_i} R_b^T \omega_{l_i}$ is the kinetic energy of the i-th link
        - m_{l_i} mass of the i-th link
        - $R_{l_i}^b \in SO(3)$ is the rotation matrix between the centre of mass of the i-th link and the body frame
    - The total kinetic energy is given by the mass matrix whose components are

$$\begin{aligned}
\boldsymbol{B}_{11} &= \left(m_b + \sum_{i=1}^{n} m_{l_i} \right) \boldsymbol{I}_3 \\
\boldsymbol{B}_{22} &= \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{H}_b \boldsymbol{Q} + \sum_{i=1}^{n} \Big(m_{l_i} \boldsymbol{T}_b^{\mathrm{T}} \boldsymbol{S}(\boldsymbol{R}_b \boldsymbol{p}_{bl_i}^b)^{\mathrm{T}} \boldsymbol{S}(\boldsymbol{R}_b \boldsymbol{p}_{bl_i}^b) \boldsymbol{T}_b \\
&\quad + \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{R}_{l_i}^b \boldsymbol{H}_{l_i} \boldsymbol{R}_b^{l_i} \boldsymbol{Q} \Big) \\
\boldsymbol{B}_{33} &= \sum_{i=1}^{n} \left(m_{l_i} \boldsymbol{J}_P^{(l_i)^{\mathrm{T}}} \boldsymbol{J}_P^{(l_i)} + \boldsymbol{J}_O^{(l_i)^{\mathrm{T}}} \boldsymbol{R}_{l_i}^b \boldsymbol{H}_{l_i} \boldsymbol{R}_b^{l_i} \boldsymbol{J}_O^{(l_i)} \right) \\
\boldsymbol{B}_{12} &= \boldsymbol{B}_{21}^{\mathrm{T}} = -\sum_{i=1}^{n} \left(m_{l_i} \boldsymbol{S}(\boldsymbol{R}_b \boldsymbol{p}_{bl_i}^b) \boldsymbol{T}_b \right) \\
\boldsymbol{B}_{13} &= \boldsymbol{B}_{31}^{\mathrm{T}} = \sum_{i=1}^{n} \left(m_{l_i} \boldsymbol{R}_b \boldsymbol{J}_P^{(l_i)} \right) \\
\boldsymbol{B}_{23} &= \boldsymbol{B}_{32}^{\mathrm{T}} = \sum_{i=1}^{n} \Big(\boldsymbol{Q}^{\mathrm{T}} \boldsymbol{R}_{l_i}^b \boldsymbol{H}_{l_i} \boldsymbol{R}_b^{l_i} \boldsymbol{J}_O^{(l_i)} \\
&\quad - m_{l_i} \boldsymbol{T}_b^{\mathrm{T}} \boldsymbol{S}(\boldsymbol{R}_b \boldsymbol{p}_{bl_i}^b)^{\mathrm{T}} \boldsymbol{R}_b \boldsymbol{J}_P^{(l_i)} \Big),
\end{aligned}$$

Modelling of an aerial manipulator

- The total potential energy is $U = U_b + \sum_{i=1}^{n} U_{l_i}$
  - $U_b = mge_3^T p_b$ is the UAV potential energy
  - $U_{l_i} = m_{l_i} g e_3^T (p_b + R_b p_{bl_i}^b)$ is the link potential energy
- Computing the Christoffel symbols as usual, the dynamic model of a UAM in compact form is

$$B(\zeta)\ddot{\zeta} + C(\zeta, \dot{\zeta})\dot{\zeta} + \bar{g}(\zeta) = u$$

  - $\bar{g}(\zeta) = \frac{\partial U(\zeta)}{\partial \zeta}$ is the gravity vector
  - $u \in \mathbb{R}^{n+6}$ is not the real input vector which is computed below
- Let define the real input vector as

$$f = \begin{bmatrix} u_T \\ \tau^b \\ \tau_q \end{bmatrix} \in \mathbb{R}^{4+n}$$

  - This in case of a flat UAV configuration and a fully-actuated manipulator
  - $\tau_q \in \mathbb{R}^n$ are the manipulator torques

- The relationship between u and f is given by

$$u = \Omega f$$

  - $\Omega = diag\{R_b e_3, Q^T, I_n\} \in \mathbb{R}^{(n+6)\times(n+4)}$
  - $\mathrm{B}(\zeta)\ddot{\zeta} + C(\zeta,\dot{\zeta})\dot{\zeta} + \bar{g}(\zeta) = \Omega f$

Centralized control approach

- The UAM is seen as a unique entity
- It is a classical mechanical system with dynamic model $\mathrm{B}(\zeta)\ddot{\zeta} + C(\zeta, \dot{\zeta})\dot{\zeta} + \bar{g}(\zeta) = \Omega f$
- If the UAV is tilted or tilting, the mapping given by Ω may be bijective (fully actuation)
  - Otherwise, as in the case of quadrotor, we have than $rank(\Omega) < \dim[\zeta]$ and the system is underactuated
    - In this case, one solution is to design the controller for the entire control vector, u
    - Then, the pseudo-inverse of Ω can be calculated since $\Omega^T\Omega$ is always invertible
    - Otherwise, a hierarchical controller for the UAM may be devised

Decentralized control approach

- The UAV and the manipulator are seen as two separated entities
  - The flight is decoupled from the manipulation
- The UAV is controlled independently from the manipulator
  - Each device has its own controller implementing suitable estimators to compensate for reciprocal interactions and disturbances