Robotics

Nicolas Marchand

Nicolas.Marchand@gipsa-lab.fr

Control Systems Department, **gipsa**-lab
Grenoble, France

ENSE3-ASI

Introduction

- Historical perspective
  - First use of the word Robot (means forced labor or serf in Czech) in the play R.U.R. (Rossum's Universal Robots) by Karel Capek (1890-1938) in January 1921.

  In R.U.R., Capek poses a paradise, where the machines initially bring so many benefits but in the end bring an equal amount of blight in the form of unemployment and social unrest

Metropolis, Fritz Lang, 1927

- Science fiction
  - Often a bad image: men against robots, dystopic society, etc. More and more a good image.

Formal definition (Robot Institute of America)

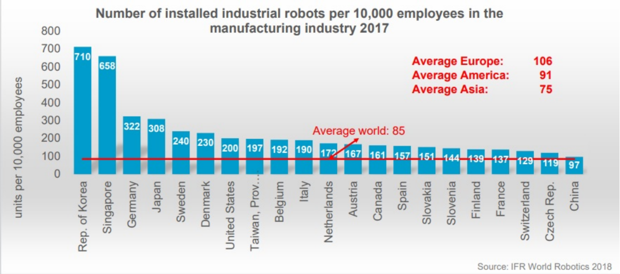
A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks

Robots and their image

- Robots have a bad image (1930-1960)
  - Robots take human works
  - Robots are dangerous since potentially independent and more intelligent than we are
- Robots have a better image (1960-today)
  - Robots can make things that human can not do (space, etc.)
  - Human can do things that robots can not do (we still are clever)
  - Robots can be games
  - Robots can be good or bad

Robotics industry: where ? (1/many)

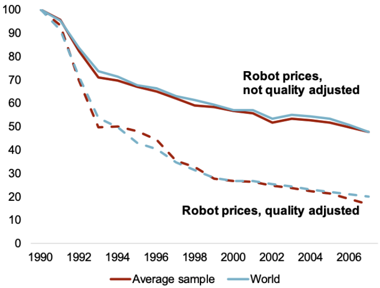
- Number of robots for every 10 000 workers:



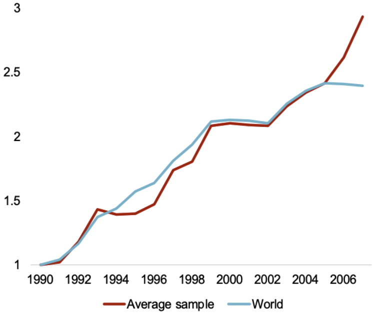
- 70% of robots in companies with more than 1000 employees
- 17% of robots in companies with less than 300 employees
- In 2002, 95% of robots > 30k€ and 32% of robots > 60k€
- 79% of decrease of the mean price between 1990 and 2002
- Average price in 2018: 45k€ (63k€ in 2009)
- Big robots manufacturers: ABB (S), KUKA (G), Fanuc (JP), etc.

Robotics industry: how much ? (2/many)

- Robot price, evolution:



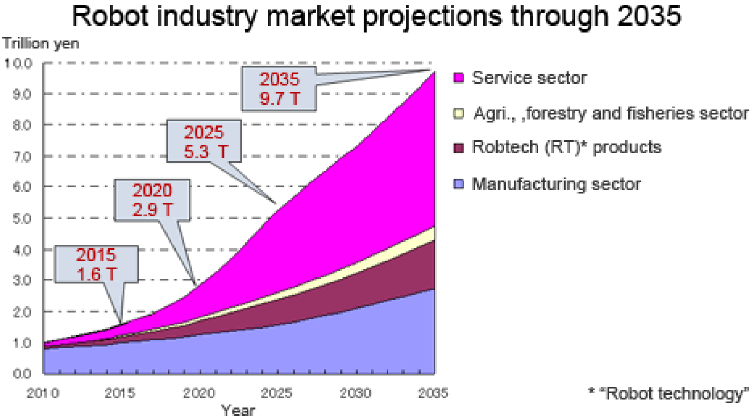
- Robot quality, evolution:



- Decrease of the price, increase of the quality
- "The Impact of Industrial Robots on EU Employment and Wages: A Local Labour Market Approach", F. Chiacchio, G. Petropoulos and D. Pichler, 2018

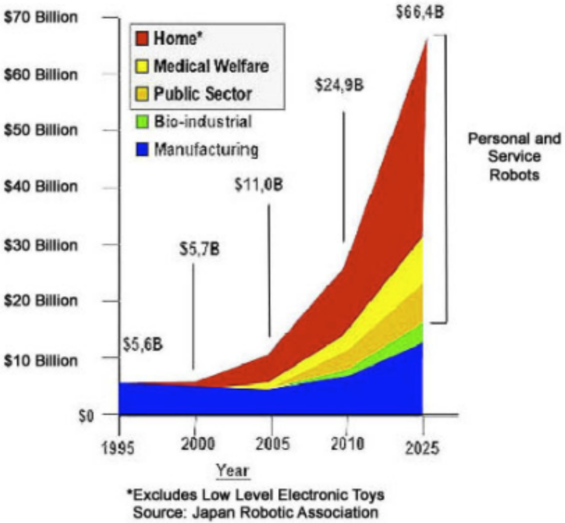
Robotics industry: for what ? (3/many)

- In which industry sectors:



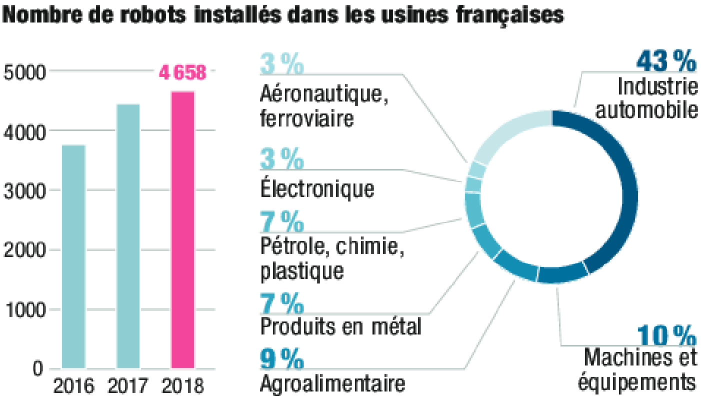
Robotics industry: for what ? (3/many)

- In which industry sectors:



Robotics industry: for what ? (3/many)

- In which industry sectors:

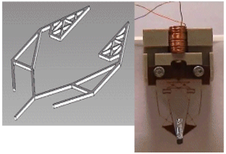


Robotics industry: examples (4/many)

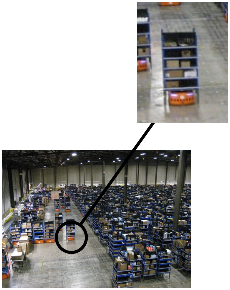
Vacuum cleaner (Kärcher)

Forest robot

Kuka robot for automotive industry



Micromanipulator

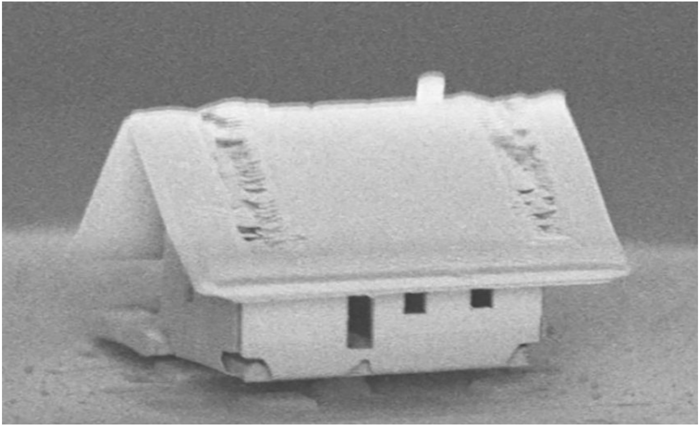


Hollywood robots

Surgical robot

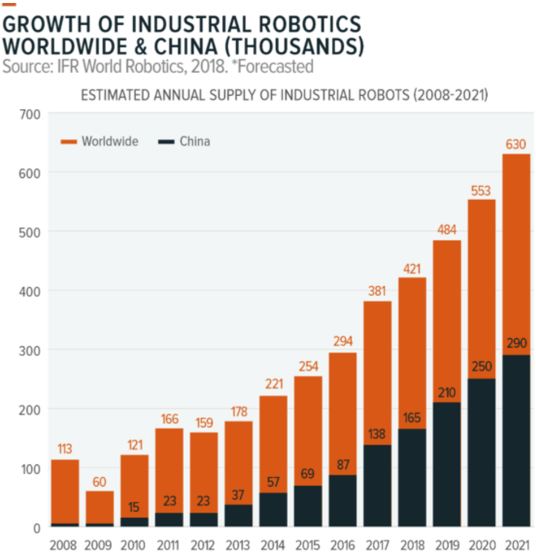
Robotics industry: nano robotics (5/many)

- Example of nano manipulation

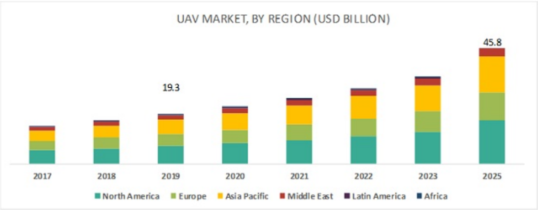


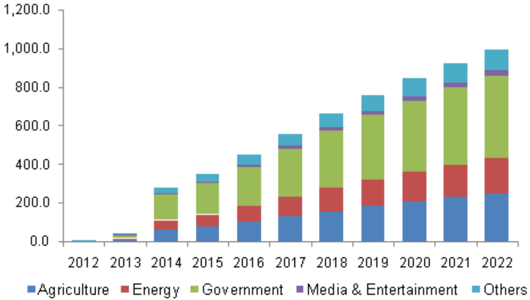
Nano house from FEMTO-ST (France)

Robotics industry: evolution (6/many)



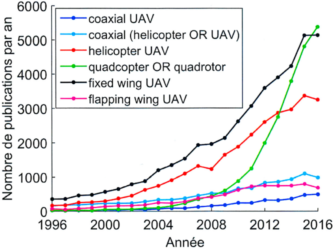
Robotics industry: UAVs (7/many)

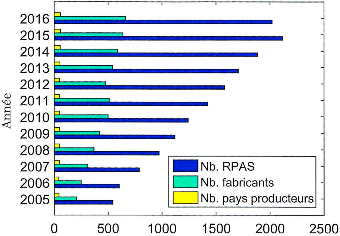




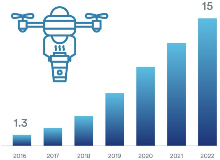
Robotics industry: UAVs (8/many)

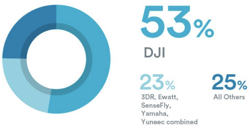
- Development of the drone's industry:





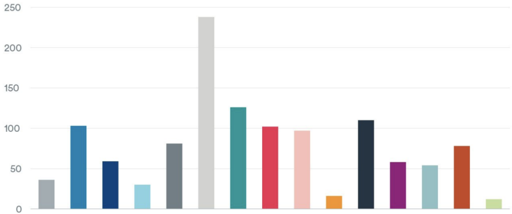
- Foundation of DJI: 2006





- Very competitive market with a high technological level of integration
- Commercial margin of 10% to 15% (more than 50% on iPhone)

ROBOTICS INDUSTRY: UAVS (10/MANY)



36% Agriculture

103% Building Inspection

59% Construction

30% Mining

81% Transportation

238% Delivery & Logistics

126% Oil & Gas

102% Power Lines

97% Renewable Energy

16% Media & Entertainment

110% Police & Fire

58% Traffic Monitoring

54% Environmental Monitoring

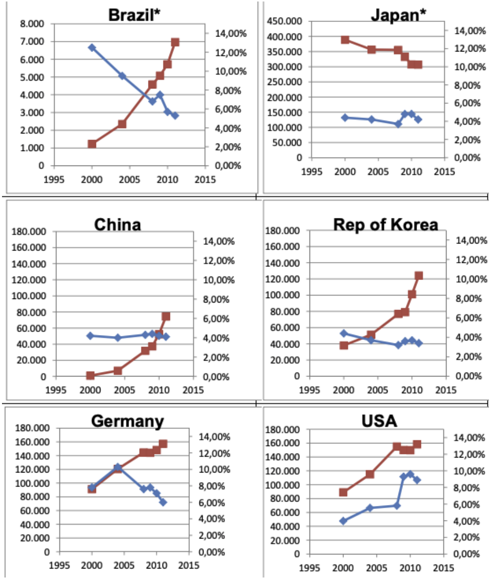
78% Disaster Response

12% Public Land Management

Robolution (1/5)

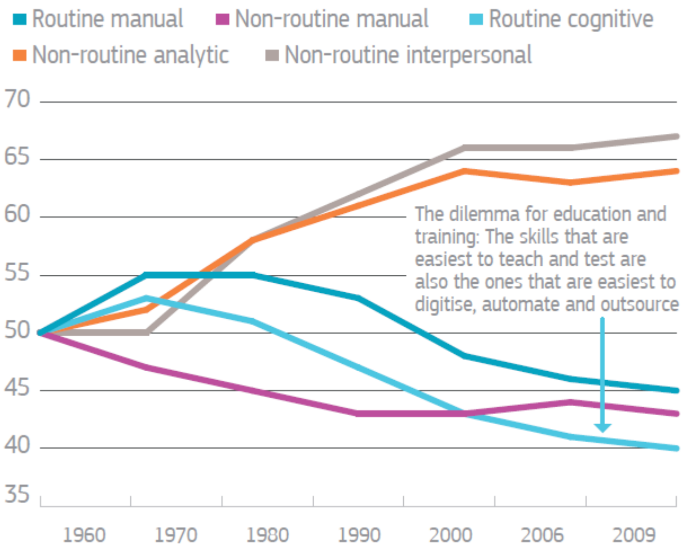
- Robotics enables 90% of cost reduction (60% for delocation)
- Each new robot destroys 6.2 jobs [MIT/Boston 1990-2007, 2017]
- 47% of jobs in the US, 50% of jobs in Europe have a high risk of being replaced by robots in the next 20 years [Oxford, 2013] ... but only 9% according [OCDE, 2016]
- Poor countries are more vulnerable, especially world factories (85% of the jobs in Ethiopia, 77% in Chine [World Bank])
- Sectors with high impact: Administration et Production
- Winner sectors: Finance, Maths/Sciences, Education
- No link between unemployment and robots
- Helps to relocate jobs in countries where the consumers are
- Very few studies on created jobs (compared to destroyed jobs)
- 800 000 direct jobs in robotics in 2020 and more than 2 millions in connected domains (electronic, energy, agriculture, etc.)

Robolution (2/5)

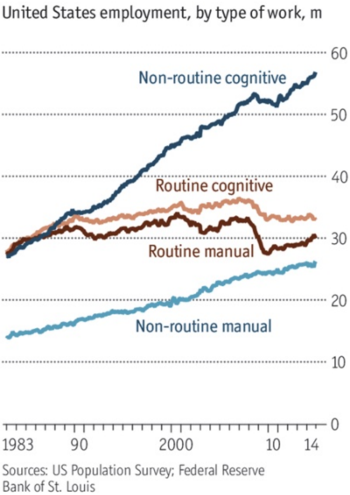


- Number of robots
- Unemployement (%)

Robolution (3/5)



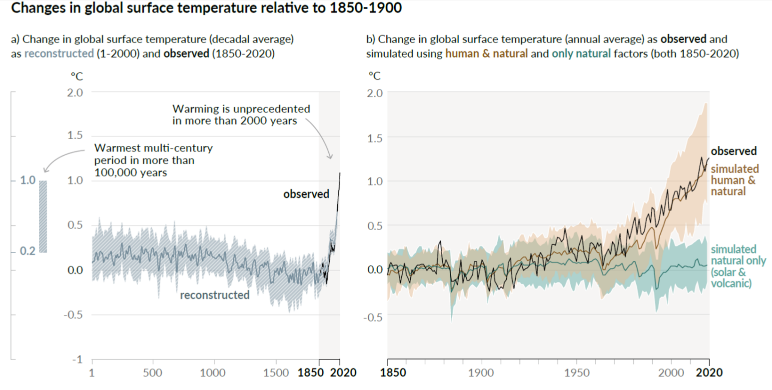
Robolution (4/5)



Robolution (5/5)

- What about the previous industrial revolution ?
  - Machines have created more jobs than they have replaced in the last 140 years
  - Working is getting less and less exhausting
  - Increase of new jobs (+580% éducation)
  - But we had fears, as in any big change periods:
    - 1675 : Destruction of machines by weavers (England), 1788 : 2000 workers break weaving machines (France), 1811-1812 : Luddism (Angleterre)
    - 1858 : Karl Marx is prophesies the replacement of the humans by machines
    - 1930 : John Maynard Keynes invents the term "technological unemployment"

Global warming challenge (1/6)



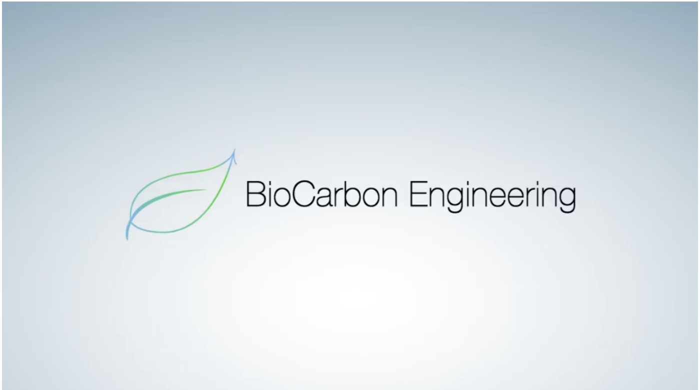
- Global warming challenge
- Exhaustion of raw materials

Global warming challenge (2/6)

- Human activity impact on climate, biodiversity and health...
- Robotic systems require ressources
- Solutions:
  - Economic decrease
  - Including the warming challenge in a strategy of conception
    - Recycling
    - New usages (replacing existing ones)

Global warming challenge (3/6)

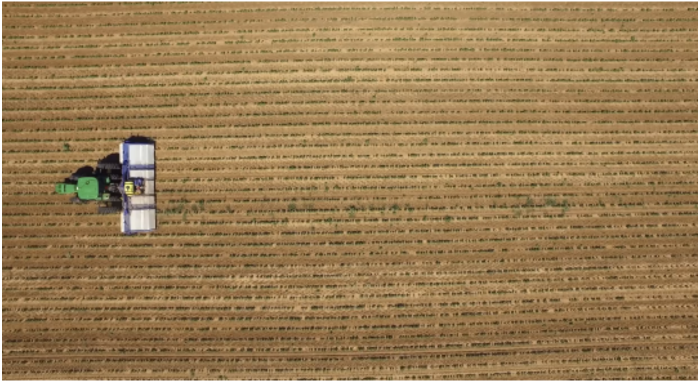
Global warming challenge (4/6)



- 10 UAVs could plant up to 400 000 trees per day
- Much less carbon consuming than other means

Global warming challenge (5/6)

Global warming challenge (6/6)



Outline

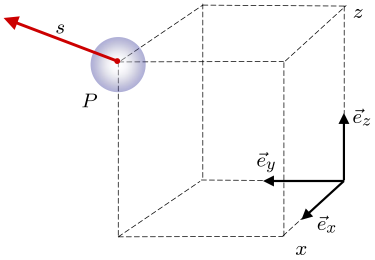
- Basic mechanics for robotics
  - Space representation
    frames, coordinate transformation, etc.
  - Force and torques
- Modelisation
- Control for robots
  - All potential problems:
    Oscillations, dry friction, saturations, etc.
  - Linear approaches
  - Nonlinear approaches

Outline

Outline

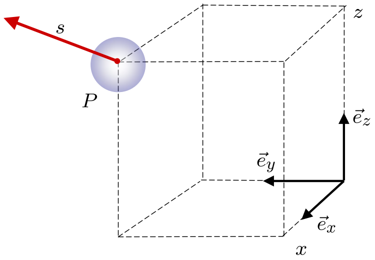
Position and speed

- The **position** of some point P in the **fixed** frame $\mathcal{F}(o, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ is the vector $\vec{p} = (x, y, z)^T$



Position and speed

- The **position** of some point P in the **fixed** frame $\mathcal{F}(o, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ is the vector $\vec{p} = (x, y, z)^T$
- The **speed** of P in $\mathcal{F}$ is the vector $\vec{s} = \dot{\vec{p}} = (\dot{x}, \dot{y}, \dot{z})^T$



Rotations

- A rotation is represented by a 3×3 matrix R such that $R^T = R^{-1}$ and $\det R = 1$

Rotations

- A rotation is represented by a 3×3 matrix R such that $R^T = R^{-1}$ and $\det R = 1$
- A **rotation** of angle θ around:

Rotations

- A rotation is represented by a 3×3 matrix R such that $R^T = R^{-1}$ and $\det R = 1$
- A **rotation** of angle θ around:
  - axis $\vec{e}_x$ is given by:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

Rotations

- A rotation is represented by a 3×3 matrix R such that $R^T = R^{-1}$ and $\det R = 1$
- A **rotation** of angle θ around:
  - axis $\vec{e}_x$ is given by:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_y$ is given by:

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

Rotations

- A rotation is represented by a 3×3 matrix R such that $R^T = R^{-1}$ and $\det R = 1$
- A **rotation** of angle θ around:
  - axis $\vec{e}_x$ is given by:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_y$ is given by:

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_z$ is given by:

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotations

- A rotation is represented by a 3×3 matrix R such that $R^T = R^{-1}$ and $\det R = 1$
- A **rotation** of angle θ around:
  - axis $\vec{e}_x$ is given by:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_y$ is given by:

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_z$ is given by:

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  - a unit vector $\vec{u} = (u_x, u_y, u_z)^T$:

$$\begin{pmatrix} u_x^2 + (1-u_x^2)c_\theta & u_x u_y(1-c_\theta) - u_z s_\theta & u_x u_z(1-c_\theta) + u_y s_\theta \\ u_x u_y(1-c_\theta) + u_z s_\theta & u_y^2 + (1-u_y^2)c_\theta & u_y u_z(1-c_\theta) - u_x s_\theta \\ u_x u_z(1-c_\theta) - u_y s_\theta & u_y u_z(1-c_\theta) + u_x s_\theta & u_z^2 + (1-u_z^2)c_\theta \end{pmatrix}$$

with $c_\cdot = \cos(\cdot)$ and $s_\cdot = \sin(\cdot)$ (and later on $t_\cdot = \tan(\cdot)$)

Rotations

- A rotation is represented by a 3×3 matrix R such that $R^T = R^{-1}$ and $\det R = 1$
- A **rotation** of angle θ around:
  - axis $\vec{e}_x$ is given by:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_y$ is given by:

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_z$ is given by:

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  - a unit vector $\vec{u} = (u_x, u_y, u_z)^T$:

$$\begin{pmatrix} u_x^2 + (1-u_x^2)c_\theta & u_x u_y(1-c_\theta) - u_z s_\theta & u_x u_z(1-c_\theta) + u_y s_\theta \\ u_x u_y(1-c_\theta) + u_z s_\theta & u_y^2 + (1-u_y^2)c_\theta & u_y u_z(1-c_\theta) - u_x s_\theta \\ u_x u_z(1-c_\theta) - u_y s_\theta & u_y u_z(1-c_\theta) + u_x s_\theta & u_z^2 + (1-u_z^2)c_\theta \end{pmatrix}$$

  with $c_. = \cos(\cdot)$ and $s_. = \sin(\cdot)$ (and later on $t_. = \tan(\cdot)$)
- The coordinates q of point Q obtained by rotating P with rotation R is $q = Rp$

Rotations

- A rotation is represented by a 3×3 matrix R such that $R^T = R^{-1}$ and $\det R = 1$
- A **rotation** of angle θ around:
  - axis $\vec{e}_x$ is given by:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_y$ is given by:

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

  - axis $\vec{e}_z$ is given by:

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  - a unit vector $\vec{u} = (u_x, u_y, u_z)^T$:

$$\begin{pmatrix} u_x^2 + (1-u_x^2)c_\theta & u_x u_y(1-c_\theta) - u_z s_\theta & u_x u_z(1-c_\theta) + u_y s_\theta \\ u_x u_y(1-c_\theta) + u_z s_\theta & u_y^2 + (1-u_y^2)c_\theta & u_y u_z(1-c_\theta) - u_x s_\theta \\ u_x u_z(1-c_\theta) - u_y s_\theta & u_y u_z(1-c_\theta) + u_x s_\theta & u_z^2 + (1-u_z^2)c_\theta \end{pmatrix}$$

  with $c_\cdot = \cos(\cdot)$ and $s_\cdot = \sin(\cdot)$ (and later on $t_\cdot = \tan(\cdot)$)
- The coordinates q of point Q obtained by rotating P with rotation R is $q = Rp$
- The rotation resulting from 2 successive rotations R_1 and then R_2 is $R_2 R_1$

Products and associated tools

- The **scalar product** $< v_1, v_2 >$ is defined by: $< v_1, v_2 > := v_1^T v_2 \in \mathbb{R}$

Products and associated tools

- The **scalar product** $< v_1, v_2 >$ is defined by: $< v_1, v_2 > := v_1^T v_2 \in \mathbb{R}$
- The **cross product** $v_1 \times v_2$ is defined by:

$$v_1 \times v_2 := \begin{pmatrix} v_{1y} v_{2z} - v_{1z} v_{2y} \\ v_{1z} v_{2x} - v_{1x} v_{2z} \\ v_{1x} v_{2y} - v_{1y} v_{2x} \end{pmatrix} \in \mathbb{R}^3$$

Products and associated tools

- The **scalar product** $< v_1, v_2 >$ is defined by: $< v_1, v_2 > := v_1^T v_2 \in \mathbb{R}$
- The **cross product** $v_1 \times v_2$ is defined by:

$$v_1 \times v_2 := \begin{pmatrix} v_{1y} v_{2z} - v_{1z} v_{2y} \\ v_{1z} v_{2x} - v_{1x} v_{2z} \\ v_{1x} v_{2y} - v_{1y} v_{2x} \end{pmatrix} \in \mathbb{R}^3$$

- The **skew-symmetric matrix** associated to a vector $p = (x, y, z)^T$ is:

$$p^\times := \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

Products and associated tools

- The **scalar product** $< v_1, v_2 >$ is defined by: $< v_1, v_2 > := v_1^T v_2 \in \mathbb{R}$
- The **cross product** $v_1 \times v_2$ is defined by:

$$v_1 \times v_2 := \begin{pmatrix} v_{1y} v_{2z} - v_{1z} v_{2y} \\ v_{1z} v_{2x} - v_{1x} v_{2z} \\ v_{1x} v_{2y} - v_{1y} v_{2x} \end{pmatrix} \in \mathbb{R}^3$$

- The **skew-symmetric matrix** associated to a vector $p = (x, y, z)^T$ is:

$$p^\times := \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

- The set of skew-symmetric matrix with the brackett $[M_1, M_2] = M_1 M_2 - M_2 M_1$ is called $SO(3)$ and forms an algebra

Products and associated tools

- The **scalar product** $< v_1, v_2 >$ is defined by: $< v_1, v_2 >:= v_1^T v_2 \in \mathbb{R}$
- The **cross product** $v_1 \times v_2$ is defined by:

$$v_1 \times v_2 := \begin{pmatrix} v_{1y}v_{2z} - v_{1z}v_{2y} \\ v_{1z}v_{2x} - v_{1x}v_{2z} \\ v_{1x}v_{2y} - v_{1y}v_{2x} \end{pmatrix} \in \mathbb{R}^3$$

- The **skew-symmetric matrix** associated to a vector $p = (x, y, z)^T$ is:

$$p^\times := \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

- The set of skew-symmetric matrix with the brackett $[M_1, M_2] = M_1M_2 - M_2M_1$ is called $SO(3)$ and forms an algebra
- Skew-symmetric matrices and cross product:

Products and associated tools

- The **scalar product** $< v_1, v_2 >$ is defined by: $< v_1, v_2 > := v_1^T v_2 \in \mathbb{R}$
- The **cross product** $v_1 \times v_2$ is defined by:

$$v_1 \times v_2 := \begin{pmatrix} v_{1y}v_{2z} - v_{1z}v_{2y} \\ v_{1z}v_{2x} - v_{1x}v_{2z} \\ v_{1x}v_{2y} - v_{1y}v_{2x} \end{pmatrix} \in \mathbb{R}^3$$

- The **skew-symmetric matrix** associated to a vector $p = (x, y, z)^T$ is:

$$p^\times := \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

- The set of skew-symmetric matrix with the brackett $[M_1, M_2] = M_1M_2 - M_2M_1$ is called $SO(3)$ and forms an algebra
- Skew-symmetric matrices and cross product:

$$v^\times u = v \times u$$

Products and associated tools

- The **scalar product** $< v_1, v_2 >$ is defined by: $< v_1, v_2 >:= v_1^T v_2 \in \mathbb{R}$
- The **cross product** $v_1 \times v_2$ is defined by:

$$v_1 \times v_2 := \begin{pmatrix} v_{1y}v_{2z} - v_{1z}v_{2y} \\ v_{1z}v_{2x} - v_{1x}v_{2z} \\ v_{1x}v_{2y} - v_{1y}v_{2x} \end{pmatrix} \in \mathbb{R}^3$$

- The **skew-symmetric matrix** associated to a vector $p = (x, y, z)^T$ is:

$$p^\times := \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

- The set of skew-symmetric matrix with the brackett $[M_1, M_2] = M_1M_2 - M_2M_1$ is called $SO(3)$ and forms an algebra
- Skew-symmetric matrices and cross product:

$$v^\times u = v \times u$$

- Skew-symmetric matrices and rotations

Products and associated tools

- The **scalar product** $< v_1, v_2 >$ is defined by: $< v_1, v_2 >:= v_1^T v_2 \in \mathbb{R}$
- The **cross product** $v_1 \times v_2$ is defined by:

$$v_1 \times v_2 := \begin{pmatrix} v_{1y}v_{2z} - v_{1z}v_{2y} \\ v_{1z}v_{2x} - v_{1x}v_{2z} \\ v_{1x}v_{2y} - v_{1y}v_{2x} \end{pmatrix} \in \mathbb{R}^3$$

- The **skew-symmetric matrix** associated to a vector $p = (x, y, z)^T$ is:

$$p^\times := \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

- The set of skew-symmetric matrix with the brackett $[M_1, M_2] = M_1M_2 - M_2M_1$ is called $SO(3)$ and forms an algebra
- Skew-symmetric matrices and cross product:

$$v^\times u = v \times u$$

- Skew-symmetric matrices and rotations

$$u^\times \sin\theta + (I - uu^T)\cos\theta + uu^T = \exp((u\theta)^\times)$$

is the rotation of angle θ leaving axis u fixed

Attitude representation: angles

- Attitude:

Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?

Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**

Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations

Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles

Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles
  - Quaternions

Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles
  - Quaternions
  - Rotation matrix

Attitude representation: angles

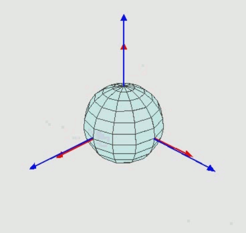
- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles
  - Quaternions
  - Rotation matrix
  - Tait-Bryan angles, Fick angles, Helmholtz angles, dip-slip-rake, azimuth-elevation-skew, . . .

Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles
  - Quaternions
  - Rotation matrix
  - Tait-Bryan angles, Fick angles, Helmholtz angles, dip-slip-rake, azimuth-elevation-skew, . . .

L. Euler (1707-1783)

- **Euler angles**: 3 angles, 27 possible rotations

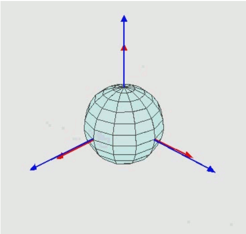


Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles
  - Quaternions
  - Rotation matrix
  - Tait-Bryan angles, Fick angles, Helmholtz angles, dip-slip-rake, azimuth-elevation-skew, ...

L. Euler (1707-1783)

- **Euler angles**: 3 angles, 27 possible rotations
- Engineering and robotics communities typically use 3-1-3 Euler angles

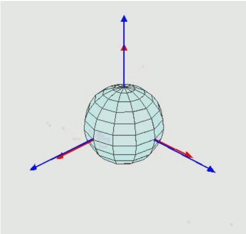


Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles
  - Quaternions
  - Rotation matrix
  - Tait-Bryan angles, Fick angles, Helmholtz angles, dip-slip-rake, azimuth-elevation-skew, . . .

L. Euler (1707-1783)

- **Euler angles**: 3 angles, 27 possible rotations
- Engineering and robotics communities typically use 3-1-3 Euler angles
- Representations with singularities



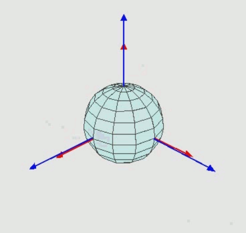
Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles
  - Quaternions
  - Rotation matrix
  - Tait-Bryan angles, Fick angles, Helmholtz angles, dip-slip-rake, azimuth-elevation-skew, . . .

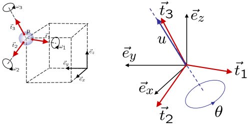
L. Euler (1707-1783)

W.R. Hamilton (1805-1865)

- **Euler angles**: 3 angles, 27 possible rotations
- Engineering and robotics communities typically use 3-1-3 Euler angles
- Representations with singularities



- **Quaternions**



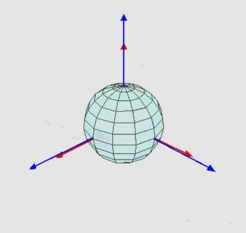
Attitude representation: angles

- Attitude:
  - equivalent of position for angles: what is the orientation of an object w.r.t. the ground ?
  - gives the **rotation that transforms the reference frame into the body frame**
- Many attitude representations
  - Euler angles
  - Quaternions
  - Rotation matrix
  - Tait-Bryan angles, Fick angles, Helmholtz angles, dip-slip-rake, azimuth-elevation-skew, . . .

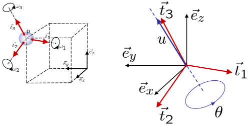
L. Euler (1707-1783)

W.R. Hamilton (1805-1865)

- **Euler angles**: 3 angles, 27 possible rotations
- Engineering and robotics communities typically use 3-1-3 Euler angles
- Representations with singularities



- **Quaternions**

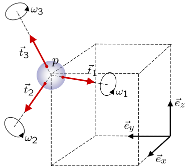


- u fixed by rotation of angle θ
- the quaternion is:

$$q = \begin{pmatrix} u_x \sin\theta/2 \\ u_y \sin\theta/2 \\ u_z \sin\theta/2 \\ \cos\theta/2 \end{pmatrix} = \begin{pmatrix} \vec{q} \\ q_0 \end{pmatrix}$$

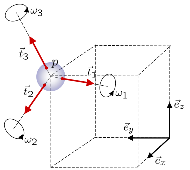
Attitude representation : angular velocities

- The angular velocity $\omega = (\omega_1, \omega_2, \omega_3)^T$ represents the rotation speed w.r.t. each axis of the body frame



Attitude representation : angular velocities

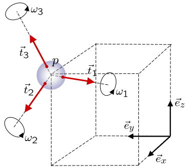
- The angular velocity $\omega = (\omega_1, \omega_2, \omega_3)^T$ represents the rotation speed w.r.t. each axis of the body frame



- **Caution:** *Angular velocities are not the time derivatives of Euler angles*

Attitude representation : angular velocities

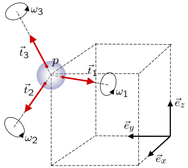
- The angular velocity $\omega = (\omega_1, \omega_2, \omega_3)^T$ represents the rotation speed w.r.t. each axis of the body frame



- **Caution:** *Angular velocities <u>are not</u> the time derivatives of Euler angles*
- Angular velocities are given by:

Attitude representation : angular velocities

- The angular velocity $\omega = (\omega_1, \omega_2, \omega_3)^T$ represents the rotation speed w.r.t. each axis of the body frame

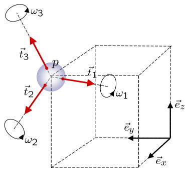


- **Caution:** *Angular velocities <u>are not</u> the time derivatives of Euler angles*
- Angular velocities are given by:
  - Rotation matrix:

$$\dot{R} = R\omega^{\times}$$

Attitude representation : angular velocities

- The angular velocity $\omega = (\omega_1, \omega_2, \omega_3)^T$ represents the rotation speed w.r.t. each axis of the body frame



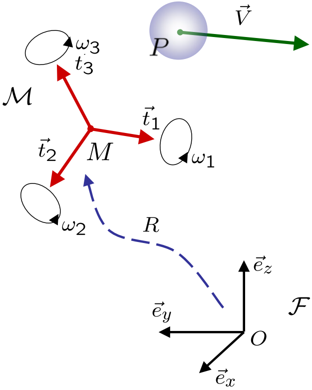
- **Caution:** *Angular velocities are not the time derivatives of Euler angles*
- Angular velocities are given by:
  - Rotation matrix:

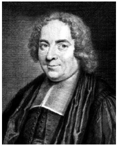
$$\dot{R} = R\omega^{\times}$$

  - Quaternions :

$$\begin{aligned}\dot{\vec{q}} &= \frac{1}{2}\Omega(\vec{\omega})q \\ &= \frac{1}{2}\Xi(q)\vec{\omega}\end{aligned} \quad \text{with} \begin{cases} \Omega(\vec{\omega}) = \begin{pmatrix} 0 & -\vec{\omega}^T \\ \vec{\omega} & -\vec{\omega}^{\times} \end{pmatrix} \\ \Xi(q) = \begin{pmatrix} -\vec{q}^T \\ I_{3\times 3}q_0 + \vec{q}^{\times} \end{pmatrix} \end{cases}$$

Moving frames





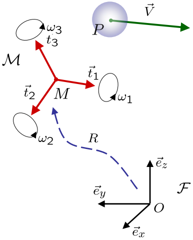
P. Varignon (1654-1722)

Varignon's formula

$$\frac{d\vec{U}}{dt}^{\mathcal{M}} = \frac{d\vec{U}}{dt}^{\mathcal{F}} + \Omega^{\mathcal{F}/\mathcal{M}} \times \vec{U}^{\mathcal{F}}$$

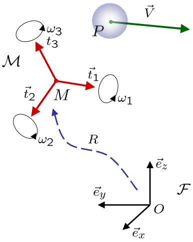
Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed inertial frame



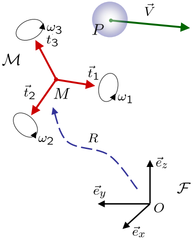
Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed inertial frame
- $\mathcal{M} := (M, \vec{t}_1, \vec{t}_2, \vec{t}_3)$: mobile frame



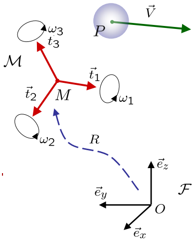
Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed inertial frame
- $\mathcal{M} := (M, \vec{t}_1, \vec{t}_2, \vec{t}_3)$: mobile frame
- R: rotation matrix s.t. $\mathcal{M} = R\mathcal{F}$



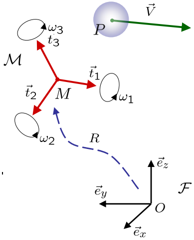
Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed inertial frame
- $\mathcal{M} := (M, \vec{t}_1, \vec{t}_2, \vec{t}_3)$: mobile frame
- R: rotation matrix s.t. $\mathcal{M} = R\mathcal{F}$
- $\Omega^{\mathcal{M}/\mathcal{F}}$: angular velocity matrix of $\mathcal{M}$ w.r.t.



Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed inertial frame
- $\mathcal{M} := (M, \vec{t}_1, \vec{t}_2, \vec{t}_3)$: mobile frame
- R: rotation matrix s.t. $\mathcal{M} = R\mathcal{F}$
- $\Omega^{\mathcal{M}/\mathcal{F}}$: angular velocity matrix of $\mathcal{M}$ w.r.t.



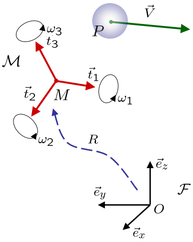
- **Velocities**:
  - Absolute velocity

$$\left.\frac{d\vec{OP}}{dt}\right|^{\mathcal{F}} = \left.\frac{d\vec{OM}}{dt}\right|^{\mathcal{F}} + \left.\frac{d\vec{MP}}{dt}\right|^{\mathcal{M}} + \Omega^{\mathcal{M}/\mathcal{F}} \times \vec{MP}$$

  - Speed of $\mathcal{M}$ w.r.t $\mathcal{F}$
  - Relative velocity
  - Due to the rotation of $\mathcal{M}$ w.r.t. $\mathcal{F}$

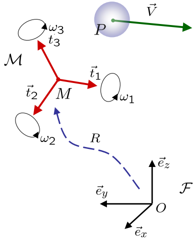
Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed frame



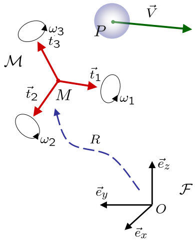
Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed frame
- $\mathcal{M} := (M, \vec{t}_1, \vec{t}_2, \vec{t}_3)$: mobile frame



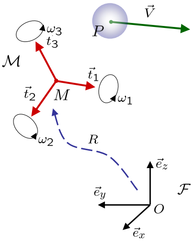
Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed frame
- $\mathcal{M} := (M, \vec{t}_1, \vec{t}_2, \vec{t}_3)$: mobile frame
- R: rotation matrix s.t. $\mathcal{M} = R\mathcal{F}$



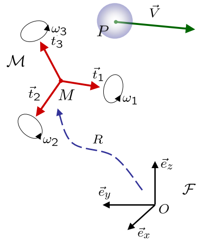
Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed frame
- $\mathcal{M} := (M, \vec{t}_1, \vec{t}_2, \vec{t}_3)$: mobile frame
- R: rotation matrix s.t. $\mathcal{M} = R\mathcal{F}$
- $\Omega^{\mathcal{M}/\mathcal{F}}$: angular velocity matrix of $\mathcal{M}$ w.r.t. $\mathcal{F}$



Moving frames

- $\mathcal{F} := (O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ fixed frame
- $\mathcal{M} := (M, \vec{t}_1, \vec{t}_2, \vec{t}_3)$: mobile frame
- R: rotation matrix s.t. $\mathcal{M} = R\mathcal{F}$
- $\Omega^{\mathcal{M}/\mathcal{F}}$: angular velocity matrix of $\mathcal{M}$ w.r.t. $\mathcal{F}$
- **Acceleration**:



$$\ddot{P}^{\mathcal{F}} := \left(\frac{d\dot{P}^{\mathcal{F}}}{dt}\right)^{\mathcal{F}} = \frac{d\dot{P}^{\mathcal{M}}}{dt}^{\mathcal{F}} + \frac{d\Omega^{\mathcal{M}/\mathcal{F}} \times P^{\mathcal{F}}}{dt}$$

- $\frac{d\dot{P}^{\mathcal{M}}}{dt}^{\mathcal{F}} = \ddot{P}^{\mathcal{M}} + \Omega^{\mathcal{M}/\mathcal{F}} \times \dot{P}^{\mathcal{M}}$ (Varignon's formula)
- $\frac{d\Omega^{\mathcal{M}/\mathcal{F}} \times P^{\mathcal{F}}}{dt} = \dot{\Omega}^{\mathcal{M}/\mathcal{F}} \times P^{\mathcal{F}} + \Omega^{\mathcal{M}/\mathcal{F}} \times \dot{P}^{\mathcal{F}}$

$$= \dot{\Omega}^{\mathcal{M}/\mathcal{F}} \times P^{\mathcal{F}} + \Omega^{\mathcal{M}/\mathcal{F}} \times \dot{P}^{\mathcal{M}} + \Omega^{\mathcal{M}/\mathcal{F}} \times (\Omega^{\mathcal{M}/\mathcal{F}} \times P^{\mathcal{F}})$$

all together:

$$\ddot{P}^{\mathcal{M}} = \ddot{P}^{\mathcal{F}} - 2\Omega \times \dot{P}^{\mathcal{M}} - \dot{\Omega} \times P^{\mathcal{F}} - \Omega \times (\Omega \times P^{\mathcal{F}})$$

- Coriolis effect ($2\Omega \times \dot{P}^{\mathcal{M}}$)
- Euler effect (tangent acceleration) ($\dot{\Omega} \times P^{\mathcal{F}}$)
- Centrifugal effect ($\Omega \times (\Omega \times P^{\mathcal{F}})$)

Newton's laws

Consider:

- an inertial frame $\mathcal{F}$
- a body of mass $m := \sum_i m_i$ composed of elements located in $\vec{p}_i$ with speed $\vec{v}_i$ in $\mathcal{F}$
- *or* a body of mass $m := \int_{\text{body}} dm$ composed of elementary part located in $\vec{p}_{dm}$ with speed $\vec{v}_{dm}$ in $\mathcal{F}$
- $\vec{p} := \frac{\sum_i m_i \vec{p}_i}{m}$ defines the position of its center of mass G in $\mathcal{F}$
- *or* $\vec{p} := \frac{\int_{\text{body}} dm \vec{p}_{dm}}{m}$ defines the position of its center of mass G in $\mathcal{F}$
- $\vec{v} := \dot{\vec{p}}$ defines speed of the center of mass
- $\vec{r}_i := (\vec{p}_i - \vec{p})$ (resp. $\vec{r}_{dm} := (\vec{p}_{dm} - \vec{p})$)

Linear Momentum

$$\vec{P} := \sum_i m_i \vec{v}_i = m\vec{v} \in \mathbb{R}^3$$

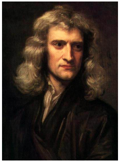
$$\vec{P} := \int_{\text{body}} \vec{v}_{dm} dm \in \mathbb{R}^3$$

Angular Momentum

$$\vec{L} := \sum_i m_i (\vec{p}_i - \vec{p}) \times \vec{v}_i$$

$$\vec{L} := \int_{\text{body}} (\vec{p}_{dm} - \vec{p}) \times \vec{v}_{dm} dm = \underbrace{\int_{\text{body}} ||\vec{r}_{dm}||^2 dm}_{J:\ \text{moment of inertia}} \vec{\omega}$$

Newton's laws



I. Newton (1643-1727)

J. L. Lagrange (1736-1813)

Consider:

- a rigid body
- an inertial frame $\mathcal{F}$
- a moving frame $\mathcal{M}$ centered in the center of mass and aligned with the main axis of the rigid body
- Let $\vec{F}_i$'s be forces applying on the body with moment arm $\vec{a}_i$

Newton's second law

$$\sum \vec{F} = \frac{d\vec{P}^{\mathcal{F}}}{dt}$$

Conservation of the angular momentum

$$\sum \vec{\tau} = \frac{d\vec{L}^{\mathcal{F}}}{dt}$$

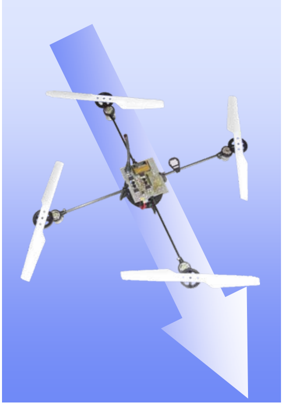
- In a moving frame (Varignon's formula):

$$\frac{d\vec{L}^{\mathcal{F}}}{dt} = \frac{d\vec{L}^{\mathcal{M}}}{dt} + \Omega \times \vec{L}$$

Outline

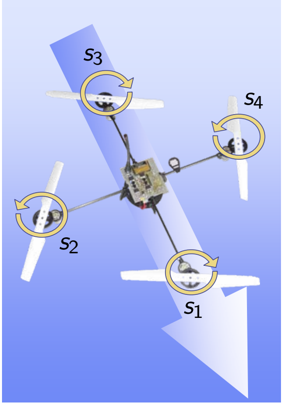
1. Introduction
2. Outline
3. Mechanics basis of rigid bodies
4. **UAV's model**
5. Recreational break
6. Legislation aspects
7. Introduction to control
8. Attitude control

How it works



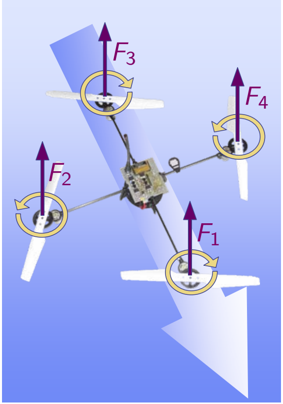
How it works

- 4 fixed rotors with controlled rotation speed s_i



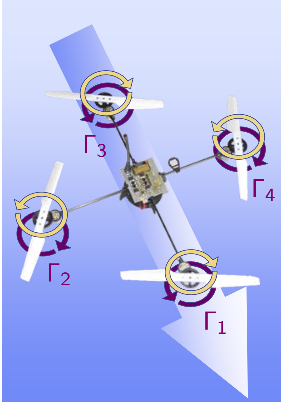
How it works

- 4 fixed rotors with controlled rotation speed s_i
- 4 generated forces F_i



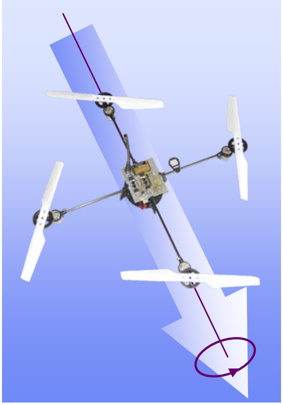
How it works

- 4 fixed rotors with controlled rotation speed s_i
- 4 generated forces F_i
- 4 counter-rotating torques Γ_i



How it works

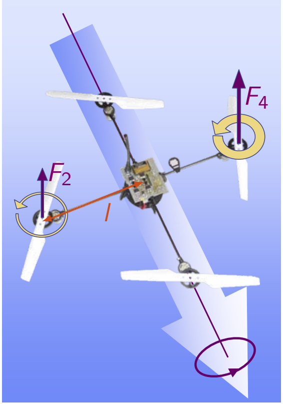
- 4 fixed rotors with controlled rotation speed s_i
- 4 generated forces F_i
- 4 counter-rotating torques Γ_i
- **Roll movement**



How it works

- 4 fixed rotors with controlled rotation speed s_i
- 4 generated forces F_i
- 4 counter-rotating torques Γ_i
- **Roll movement** generated with a dissymmetry between left and right forces:

$$\Gamma_r = l(F_4 - F_2)$$

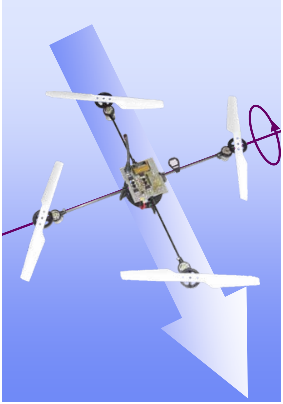


How it works

- 4 fixed rotors with controlled rotation speed s_i
- 4 generated forces F_i
- 4 counter-rotating torques Γ_i
- **Roll movement** generated with a dissymmetry between left and right forces:

$$\Gamma_r = l(F_4 - F_2)$$

- **Pitch movement**



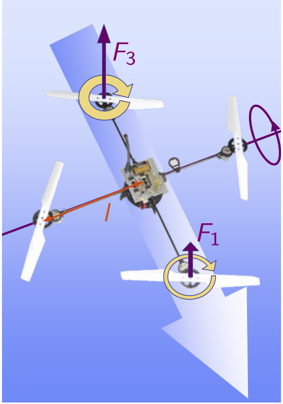
How it works

- 4 fixed rotors with controlled rotation speed s_i
- 4 generated forces F_i
- 4 counter-rotating torques Γ_i
- **Roll movement** generated with a dissymmetry between left and right forces:

$$\Gamma_r = l(F_4 - F_2)$$

- **Pitch movement** generated with a dissymmetry between front and rear forces:

$$\Gamma_p = l(F_1 - F_3)$$



How it works

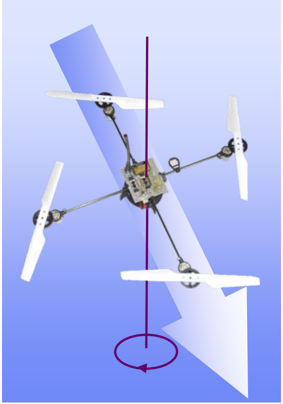
- 4 fixed rotors with controlled rotation speed s_i
- 4 generated forces F_i
- 4 counter-rotating torques Γ_i
- **Roll movement** generated with a dissymmetry between left and right forces:

$$\Gamma_r = l(F_4 - F_2)$$

- **Pitch movement** generated with a dissymmetry between front and rear forces:

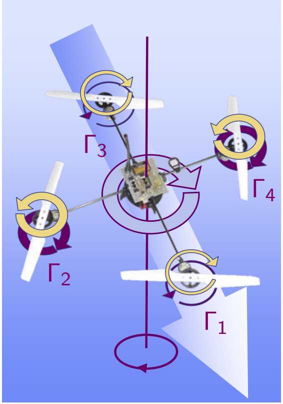
$$\Gamma_p = l(F_1 - F_3)$$

- **Yaw movement**



How it works

- 4 fixed rotors with controlled rotation speed s_i
- 4 generated forces F_i
- 4 counter-rotating torques Γ_i
- **Roll movement** generated with a dissymmetry between left and right forces:
$$\Gamma_r = l(F_4 - F_2)$$
- **Pitch movement** generated with a dissymmetry between front and rear forces:
$$\Gamma_p = l(F_1 - F_3)$$
- **Yaw movement** generated with a dissymmetry between front/rear and left/right torques:
$$\Gamma_y = \Gamma_1 + \Gamma_3 - \Gamma_2 - \Gamma_4$$



MODELING THE QUADROTOR

Actuation and aerodynamics

- **Electrical motor**: A 2^{nd} order system with friction and saturation

Actuation and aerodynamics

- **Electrical motor**: A 2[nd] order system with friction and saturation usually *approximated* by a 1[rst] order system:

$$\dot{s}_i = -\frac{k_m^2}{J_r R} s_i - \frac{1}{J_r}\tau_{\text{load}} + \frac{k_m}{J_r R}\,\mathrm{sat}_{\bar{U}_i}(U_i) \quad i \in \{1,2,3,4\}$$

s_i: rotation speed
U_i: voltage applied to the motor; real control variable
τ_{load}: motor load: $\tau_{\text{load}} = k_{gearbox} c_D |s_i| s_i$ with c_D drag coefficient

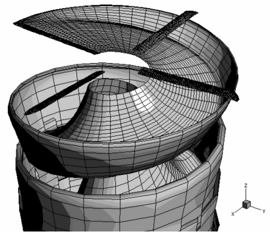
Actuation and aerodynamics

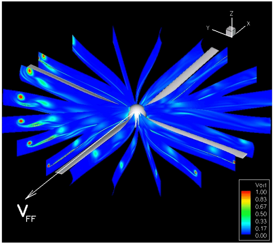
- **Electrical motor**: A 2nd order system with friction and saturation usually *approximated* by a 1rst order system:

$$\dot{s}_i = -\frac{k_m^2}{J_r R} s_i - \frac{1}{J_r}\tau_{\text{load}} + \frac{k_m}{J_r R}\,\mathrm{sat}_{\bar{U}_i}(U_i) \quad i \in \{1,2,3,4\}$$

s_i: rotation speed
U_i: voltage applied to the motor; real control variable
τ_{load}: motor load: $\tau_{\text{load}} = k_{gearbox} c_D |s_i| s_i$ with c_D drag coefficient

- **Aerodynamical forces and torques**: Very complex models exist





Actuation and aerodynamics

- **Electrical motor**: A 2^{nd} order system with friction and saturation usually *approximated* by a 1^{rst} order system:

$$\dot{s}_i = -\frac{k_m^2}{J_r R} s_i - \frac{1}{J_r}\tau_{\text{load}} + \frac{k_m}{J_r R}\,\text{sat}_{\bar{U}_i}(U_i) \quad i \in \{1,2,3,4\}$$

s_i: rotation speed

U_i: voltage applied to the motor; real control variable

τ_{load}: motor load: $\tau_{\text{load}} = k_{gearbox} c_D |s_i| s_i$ with c_D drag coefficient

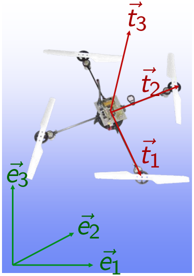
- **Aerodynamical forces and torques**: Very complex models exist but overcomplicated for control, better use the *simplified* model:

$$\begin{aligned} F_i &= c_T s_i^2 \\ \Gamma_r &= l c_T (s_4^2 - s_2^2) \\ \Gamma_p &= l c_T (s_1^2 - s_3^2) \\ \Gamma_y &= l c_D (s_1^2 + s_3^2 - s_2^2 - s_4^2) \end{aligned} \qquad i \in \{1,2,3,4\}$$

c_T: thrust coefficient, c_D: drag coefficient

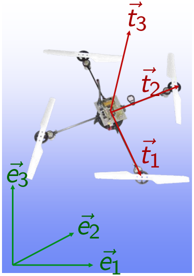
Frames and variables

- Two frames
  - a fixed frame $\mathcal{E}(\vec{e}_1, \vec{e}_2, \vec{e}_3)$
  - a frame attached to the X4 $\mathcal{T}(\vec{t}_1, \vec{t}_2, \vec{t}_3)$



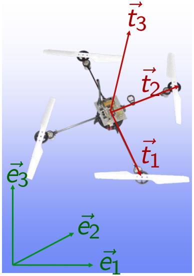
Frames and variables

- Two frames
  - a fixed frame $\mathcal{E}(\vec{e}_1, \vec{e}_2, \vec{e}_3)$
  - a frame attached to the X4 $\mathcal{T}(\vec{t}_1, \vec{t}_2, \vec{t}_3)$
- Frame change
  - a rotation matrix R from $\mathcal{T}$ to $\mathcal{E}$



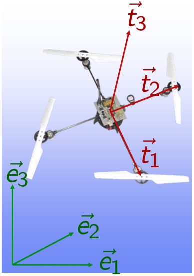
Frames and variables

- Two frames
  - a fixed frame $\mathcal{E}(\vec{e}_1, \vec{e}_2, \vec{e}_3)$
  - a frame attached to the X4 $\mathcal{T}(\vec{t}_1, \vec{t}_2, \vec{t}_3)$
- Frame change
  - a rotation matrix R from $\mathcal{T}$ to $\mathcal{E}$
- State variables:



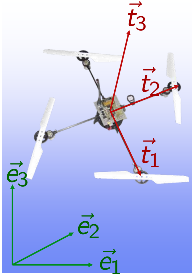
Frames and variables

- Two frames
  - a fixed frame $\mathcal{E}(\vec{e}_1, \vec{e}_2, \vec{e}_3)$
  - a frame attached to the X4 $\mathcal{T}(\vec{t}_1, \vec{t}_2, \vec{t}_3)$
- Frame change
  - a rotation matrix R from $\mathcal{T}$ to $\mathcal{E}$
- State variables:
  - Cartesian coordinates (in $\mathcal{E}$)
    - position $\vec{p}$
    - velocity $\vec{v}$



Frames and variables

- Two frames
  - a fixed frame $\mathcal{E}(\vec{e}_1, \vec{e}_2, \vec{e}_3)$
  - a frame attached to the X4 $\mathcal{T}(\vec{t}_1, \vec{t}_2, \vec{t}_3)$
- Frame change
  - a rotation matrix R from $\mathcal{T}$ to $\mathcal{E}$



- State variables:
  - Cartesian coordinates (in $\mathcal{E}$)
    - position $\vec{p}$
    - velocity $\vec{v}$
  - Attitude coordinates:
    - angular velocity $\vec{\omega}$ in the moving frame $\mathcal{T}$
    - *either:* Euler angles three successive rotations about $\vec{t}_3$, $\vec{t}_1$ and $\vec{t}_3$ of angles angles ϕ, θ and ψ giving R
    - *or:* Quaternion representation $(q_0, \vec{q}) = (\cos\beta/2, \vec{u}\sin\beta/2)$ represent a rotation of angle β about $\vec{u}$

Newton's laws

- **Cartesian coordinates**:

$$\begin{cases} \dot{\vec{p}} = \vec{v} \\ m\dot{\vec{v}} = -mg\vec{e}_3 + R \cdot \underbrace{\sum_i F_i(s_i)\vec{t}_3}_{\vec{T}\text{ : control thrust}} + \vec{F}_{ext} \end{cases}$$

Newton's laws

- **Cartesian coordinates**:

$$\begin{cases} \dot{\vec{p}} = \vec{v} \\ m\dot{\vec{v}} = -mg\vec{e}_3 + R \cdot \underbrace{\sum_i F_i(s_i)\vec{t}_3}_{\vec{T}\ :\ \text{control thrust}} + \vec{F}_{ext} \end{cases}$$

- **Attitude**:

Newton's laws

- **Cartesian coordinates**:

$$\left\{\begin{array}{rcl} \dot{\vec{p}} & = & \vec{v} \\ m\dot{\vec{v}} & = & -mg\vec{e}_3 + R \cdot \underbrace{\sum_i F_i(s_i)\vec{t}_3}_{\vec{T}\,:\ \text{control thrust}} + \vec{F}_{ext} \end{array}\right.$$

- **Attitude**:
  - Rotation matrix formalism:

$$\left\{\begin{array}{rcl} \dot{R} & = & R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} & = & -\vec{\omega}^{\times}J\vec{\omega} + \vec{\Gamma}_c + \vec{\Gamma}_{ext} \end{array}\right. \quad \text{with } \vec{\omega}^{\times} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

$\vec{\omega}^{\times}$ is the skew symmetric tensor associated to $\vec{\omega}$

Newton's laws

Robotics

N. Marchand

Introduction

Outline

Mechanics

Cartesian coordinates

Orientation

Frames

Newton

UAV's model

Recreational break

Legislation

EU legislation

Sub-categories of operations

Categories of UAVs

Introduction to control

Main components

Control loops

Attitude control

Position control

- **Cartesian coordinates**:

$$\begin{cases} \dot{\vec{p}} = \vec{v} \\ m\dot{\vec{v}} = -mg\vec{e}_3 + R \cdot \underbrace{\sum_i F_i(s_i)\vec{t}_3}_{\vec{T}\ :\ \text{control thrust}} + \vec{F}_{ext} \end{cases}$$

- **Attitude**:
  - Rotation matrix formalism:

$$\begin{cases} \dot{R} = R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times}J\vec{\omega} + \vec{\Gamma}_c + \vec{\Gamma}_{ext} \end{cases} \quad \text{with } \vec{\omega}^{\times} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

  $\vec{\omega}^{\times}$ is the skew symmetric tensor associated to $\vec{\omega}$

  - Quaternion formalism:

$$\begin{cases} \dot{q} = \frac{1}{2}\Omega(\vec{\omega})q \\ \quad = \frac{1}{2}\Xi(q)\vec{\omega} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times}J\vec{\omega} + \vec{\Gamma}_c + \vec{\Gamma}_{ext} \end{cases} \quad \text{with } \begin{cases} \Omega(\vec{\omega}) = \begin{pmatrix} 0 & -\vec{\omega}^T \\ \vec{\omega} & -\vec{\omega}^{\times} \end{pmatrix} \\ \Xi(q) = \begin{pmatrix} -\vec{q}^T \\ I_{3\times3}q_0 + \vec{q}^{\times} \end{pmatrix} \end{cases}$$

Newton's laws

- **Cartesian coordinates**:

$$\begin{cases} \dot{\vec{p}} = \vec{v} \\ m\dot{\vec{v}} = -mg\vec{e}_3 + R \cdot \underbrace{\sum_i F_i(s_i)\vec{t}_3}_{\vec{T}\ :\ \text{control thrust}} + \vec{F}_{ext} \end{cases}$$

- **Attitude**:
  - Rotation matrix formalism:

$$\begin{cases} \dot{R} = R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c + \vec{\Gamma}_{ext} \end{cases} \quad \text{with } \vec{\omega}^{\times} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

  $\vec{\omega}^{\times}$ is the skew symmetric tensor associated to $\vec{\omega}$

  - Quaternion formalism:

$$\begin{cases} \dot{q} = \frac{1}{2}\Omega(\vec{\omega})q \\ \quad = \frac{1}{2}\Xi(q)\vec{\omega} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c + \vec{\Gamma}_{ext} \end{cases} \quad \text{with} \begin{cases} \Omega(\vec{\omega}) = \begin{pmatrix} 0 & -\vec{\omega}^T \\ \vec{\omega} & -\vec{\omega}^{\times} \end{pmatrix} \\ \Xi(q) = \begin{pmatrix} -\vec{q}^T \\ I_{3\times3}q_0 + \vec{q}^{\times} \end{pmatrix} \end{cases}$$

where $\vec{\Gamma}_c = \begin{pmatrix} \Gamma_r(s_2, s_4) \\ \Gamma_p(s_1, s_3) \\ \Gamma_y(s_1, s_2, s_3, s_4) \end{pmatrix}$ are the control torques

The wronskian matrix

- Consider the 1-2-3 Euler angles (ϕ, θ, ψ)
- The rotation matrix is given by:

$$R = R_z R_y R_x = \begin{pmatrix} c_\theta c_\phi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\phi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

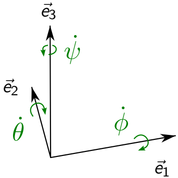
The wronskian matrix

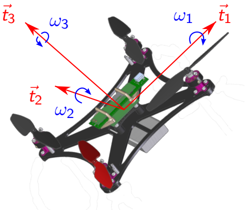
- Consider the 1-2-3 Euler angles (ϕ, θ, ψ)
- The rotation matrix is given by:

$$R = R_z R_y R_x = \begin{pmatrix} c_\theta c_\phi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\phi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

- The relation between the time derivative of the Euler angles and the angular velocity is:

$$\vec{\omega} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_z \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_z R_y \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} = W^{-1} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$





The wronskian matrix

- Consider the 1-2-3 Euler angles (ϕ, θ, ψ)
- The rotation matrix is given by:

$$R = R_z R_y R_x = \begin{pmatrix} c_\theta c_\phi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\phi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

- The relation between the time derivative of the Euler angles and the angular velocity is:

$$\vec{\omega} = \begin{pmatrix} \dot\phi \\ 0 \\ 0 \end{pmatrix} + R_z \begin{pmatrix} 0 \\ \dot\theta \\ 0 \end{pmatrix} + R_z R_y \begin{pmatrix} 0 \\ 0 \\ \dot\psi \end{pmatrix} = W^{-1} \begin{pmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{pmatrix}$$

- W is called the **wronskian matrix** given by (for 1-2-3 Euler angles):

$$W = \begin{pmatrix} 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \\ 0 & c_\phi & -s_\phi \\ 1 & s_\phi t_\theta & c_\phi t_\theta \end{pmatrix}$$

- This matrix is singular for $\theta = \pi/2 + k\pi$

The wronskian matrix

- Consider the 3-1-3 Euler angles (ϕ, θ, ψ)
- The rotation matrix is given by:

$$R = R_z R_x R_z = \begin{pmatrix} c_\psi c_\phi - s_\psi c_\theta s_\phi & -c_\psi s_\phi - s_\psi c_\theta c_\phi & s_\psi s_\theta \\ s_\psi c_\phi + c_\psi c_\theta s_\phi & -s_\psi s_\phi + c_\psi c_\theta c_\phi & -c_\psi s_\theta \\ s_\theta s_\phi & s_\theta c_\phi & c_\theta \end{pmatrix}$$

- The relation between the time derivative of the Euler angles and the angular velocity is:

$$\vec{\omega} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_x \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_z R_x \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} = W^{-1} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

- W is called the **wronskian matrix** given by (for 3-1-3 Euler angles):

$$W^{-1} = \begin{pmatrix} s_\psi s_\theta & c_\psi & 0 \\ c_\psi s_\theta & -s_\psi & 0 \\ c_\theta & 0 & 1 \end{pmatrix} \qquad W = \begin{pmatrix} \frac{s_\psi}{s_\theta} & \frac{c_\psi}{s_\theta} & 0 \\ c_\psi & -s_\psi & 0 \\ -\frac{s_\psi c_\theta}{s_\theta} & -\frac{c_\psi c_\theta}{s_\theta} & 1 \end{pmatrix}$$

- This matrix is singular for $\theta = 0 + k\pi$

A first model: review of nonlinearities

$$\left\{\begin{array}{rcl}
\dot{s}_i & = & -\dfrac{k_m^2}{J_r R} s_i - \dfrac{k_{gearbox} c_D}{J_r} |s_i|\, s_i + \dfrac{k_m}{J_r R} \operatorname{sat}_{\bar{U}_i}(U_i) \\
\dot{\vec{p}} & = & \vec{v} \\
m\dot{\vec{v}} & = & -mg\,\vec{e}_3 + R \begin{pmatrix} 0 \\ 0 \\ \sum_i F_i(s_i) \end{pmatrix} \\
\dot{R} & = & R\vec{\omega}^\times \\
J\dot{\vec{\omega}} & = & -\vec{\omega}^\times J\vec{\omega} + \begin{pmatrix} \Gamma_r(s_2, s_4) \\ \Gamma_p(s_1, s_3) \\ \Gamma_y(s_1, s_2, s_3, s_4) \end{pmatrix}
\end{array}\right.$$

A first model: review of nonlinearities

$$\begin{cases} \dot{s}_i = -\frac{k_m^2}{J_r R} s_i - \frac{k_{gearbox} c_D}{J_r} |s_i| \, s_i + \frac{k_m}{J_r R} \operatorname{sat}_{\bar{U}_i}(U_i) \\ \dot{\vec{p}} = \vec{v} \\ m\dot{\vec{v}} = -mg\vec{e}_3 + R \begin{pmatrix} 0 \\ 0 \\ \sum_i F_i(s_i) \end{pmatrix} \\ \dot{R} = R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times} J\vec{\omega} + \begin{pmatrix} \Gamma_r(s_2, s_4) \\ \Gamma_p(s_1, s_3) \\ \Gamma_y(s_1, s_2, s_3, s_4) \end{pmatrix} \end{cases}$$

In red: nonlinearities

In blue: where the control variables act

Parameter identification

- **Electrical motor**:
  - For small input steps, the system behaves very close to a **linear** first order system
  - Hence, use linear identification tools
  - $\bar{U}_i$ is found on the data-sheet of the motor (damage avoidance)

Parameter identification

- **Electrical motor**:
  - For small input steps, the system behaves very close to a **linear** first order system
  - Hence, use linear identification tools
  - $\bar{U}_i$ is found on the data-sheet of the motor (damage avoidance)
- **Aerodynamical parameters**: c_T and c_D

c_T and c_D measured with specific test beds, depends upon temperature, distance from ground, etc.

Parameter identification

- **Electrical motor**:
  - For small input steps, the system behaves very close to a **linear** first order system
  - Hence, use linear identification tools
  - $\bar{U}_i$ is found on the data-sheet of the motor (damage avoidance)
- **Aerodynamical parameters**: c_T and c_D

c_T **and** c_D measured with specific test beds, depends upon temperature, distance from ground, etc.

Parameter identification

- **Electrical motor**:
  - For small input steps, the system behaves very close to a **linear** first order system
  - Hence, use linear identification tools
  - $\bar{U}_i$ is found on the data-sheet of the motor (damage avoidance)
- **Aerodynamical parameters**: c_T and c_D

c_T and c_D measured with specific test beds, depends upon temperature, distance from ground, etc.

- **Mechanical parameters**:

Parameter identification

- **Electrical motor**:
  - For small input steps, the system behaves very close to a **linear** first order system
  - Hence, use linear identification tools
  - $\bar{U}_i$ is found on the data-sheet of the motor (damage avoidance)
- **Aerodynamical parameters**: c_T and c_D

c_T and c_D measured with specific test beds, depends upon temperature, distance from ground, etc.

- **Mechanical parameters**:

  l length of an arm of the helicopter, easy to measure

Parameter identification

- **Electrical motor**:
  - For small input steps, the system behaves very close to a **linear** first order system
  - Hence, use linear identification tools
  - $\bar{U}_i$ is found on the data-sheet of the motor (damage avoidance)
- **Aerodynamical parameters**: c_T and c_D

  c_T and c_D measured with specific test beds, depends upon temperature, distance from ground, etc.
- **Mechanical parameters**:

  l length of an arm of the helicopter, easy to measure

  m total mass of the helicopter, easy to measure

Parameter identification

- **Electrical motor**:
  - For small input steps, the system behaves very close to a **linear** first order system
  - Hence, use linear identification tools
  - $\bar{U}_i$ is found on the data-sheet of the motor (damage avoidance)
- **Aerodynamical parameters**: c_T and c_D

c_T and c_D measured with specific test beds, depends upon temperature, distance from ground, etc.

- **Mechanical parameters**:

  l length of an arm of the helicopter, easy to measure

  m total mass of the helicopter, easy to measure

  J body inertia, hard to have precisely

Parameter identification

- **Electrical motor**:
  - For small input steps, the system behaves very close to a **linear** first order system
  - Hence, use linear identification tools
  - $\bar{U}_i$ is found on the data-sheet of the motor (damage avoidance)
- **Aerodynamical parameters**: c_T and c_D

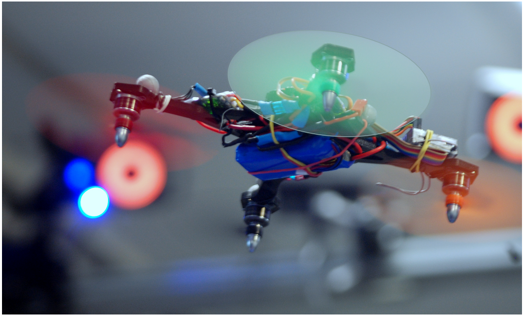
c_T and c_D measured with specific test beds, depends upon temperature, distance from ground, etc.

- **Mechanical parameters**:

l length of an arm of the helicopter, easy to measure
m total mass of the helicopter, easy to measure
J body inertia, hard to have precisely
I_r rotor inertia, hard to have precisely

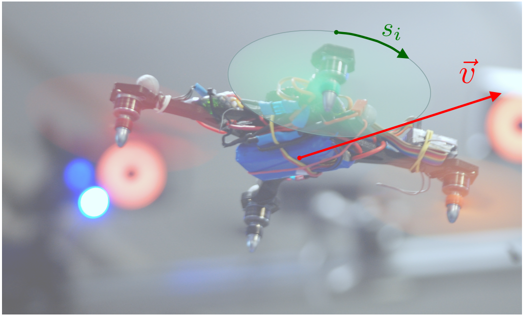
The flapping effect

- The thrust was assumed to be $\sum_i F_i(s_i)\vec{t}_3$, that is colinear to $\vec{t}_3$
- It has been proved to be false because it neglects the effect of the apparent wind speed, this is the **flapping effect**



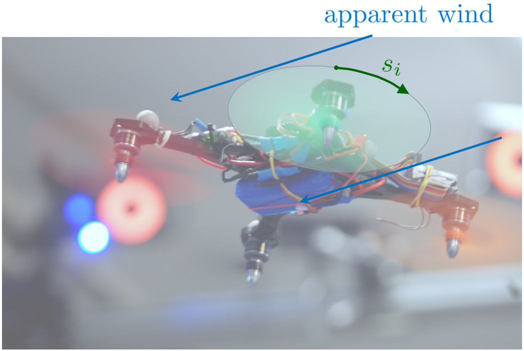
THE FLAPPING EFFECT

- The thrust was assumed to be $\sum_i F_i(s_i)\vec{t}_3$, that is colinear to $\vec{t}_3$
- It has been proved to be false because it neglects the effect of the apparent wind speed, this is the **flapping effect**



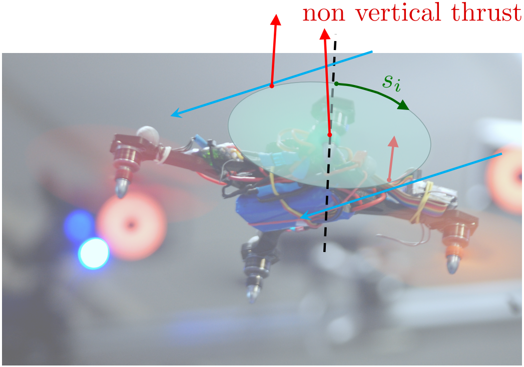
The flapping effect

- The thrust was assumed to be $\sum_i F_i(s_i)\vec{t}_3$, that is colinear to $\vec{t}_3$
- It has been proved to be false because it neglects the effect of the apparent wind speed, this is the **flapping effect**



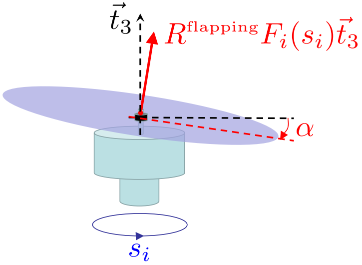
The flapping effect

- The thrust was assumed to be $\sum_i F_i(s_i)\vec{t}_3$, that is colinear to $\vec{t}_3$
- It has been proved to be false because it neglects the effect of the apparent wind speed, this is the **flapping effect**
- Higher thrust on one side of the blades
- The thrust becomes $\sum_i R_i^{\text{flapping}} F_i(s_i)\vec{t}_3$, torques are also modified



The flapping effect

- The thrust was assumed to be $\sum_i F_i(s_i)\vec{t}_3$, that is colinear to $\vec{t}_3$
- It has been proved to be false because it neglects the effect of the apparent wind speed, this is the **flapping effect**
- Higher thrust on one side of the blades
- The thrust becomes $\sum_i R_i^{\text{flapping}} F_i(s_i)\vec{t}_3$, torques are also modified



Modeling more into details: the flapping effect

- The flapping matrix takes can be decomposed :

$$\begin{aligned} R^{\text{flapping}} &= R_x^{\text{flapping}} \cdot R_y^{\text{flapping}} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c(\beta) & -s(\beta) \\ 0 & s(\beta) & c(\beta) \end{pmatrix} \cdot \begin{pmatrix} c(\alpha) & 0 & s(\alpha) \\ 0 & 1 & 0 \\ -s(\alpha) & 0 & c(\alpha) \end{pmatrix} \end{aligned}$$

Modeling more into details: the flapping effect

- The flapping matrix takes can be decomposed :

$$
\begin{aligned}
R^{\text{flapping}} &= R_x^{\text{flapping}} \cdot R_y^{\text{flapping}} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c(\beta) & -s(\beta) \\ 0 & s(\beta) & c(\beta) \end{pmatrix} \cdot \begin{pmatrix} c(\alpha) & 0 & s(\alpha) \\ 0 & 1 & 0 \\ -s(\alpha) & 0 & c(\alpha) \end{pmatrix}
\end{aligned}
$$

- α and β can be composed as follows :

$$
\begin{aligned}
\alpha &= \alpha_v + \alpha_\omega \\
\beta &= \beta_v + \beta_\omega
\end{aligned}
$$

Modeling more into details: the flapping effect

- The flapping matrix takes can be decomposed :

$$
\begin{aligned}
R^{\text{flapping}} &= R_x^{\text{flapping}} \cdot R_y^{\text{flapping}} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c(\beta) & -s(\beta) \\ 0 & s(\beta) & c(\beta) \end{pmatrix} \cdot \begin{pmatrix} c(\alpha) & 0 & s(\alpha) \\ 0 & 1 & 0 \\ -s(\alpha) & 0 & c(\alpha) \end{pmatrix}
\end{aligned}
$$

- α and β can be composed as follows :

$$
\begin{aligned}
\alpha &= \alpha_v + \alpha_\omega \\
\beta &= \beta_v + \beta_\omega
\end{aligned}
$$

- α_v and β_v represent the contribution of the linear speed of the body to the flapping effect

Modeling more into details: the flapping effect

- The flapping matrix takes can be decomposed :

$$\begin{aligned} R^{\text{flapping}} &= R_x^{\text{flapping}} \cdot R_y^{\text{flapping}} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c(\beta) & -s(\beta) \\ 0 & s(\beta) & c(\beta) \end{pmatrix} \cdot \begin{pmatrix} c(\alpha) & 0 & s(\alpha) \\ 0 & 1 & 0 \\ -s(\alpha) & 0 & c(\alpha) \end{pmatrix} \end{aligned}$$

- α and β can be composed as follows :

$$\begin{aligned} \alpha &= \alpha_v + \alpha_\omega \\ \beta &= \beta_v + \beta_\omega \end{aligned}$$

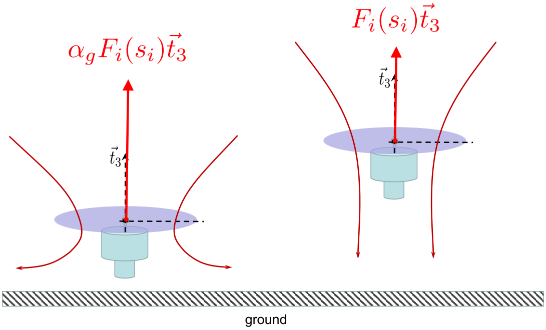
- α_v and β_v represent the contribution of the linear speed of the body to the flapping effect
- a_ω and b_ω represent the contribution of the rotational speed of the body to the flapping effect

The ground effect

- The thrust was assumed to be $\sum_i F_i(s_i)\vec{t}_3$, with $F_i(s_i) = c_T s_i^2$

The ground effect

- The thrust was assumed to be $\sum_i F_i(s_i)\vec{t}_3$, with $F_i(s_i) = c_T s_i^2$
- Unfortunately, c_T is not constant but depends upon
  - the density of the air, therefore of the temperature
  - the ground distance : it is the ground effect, $\alpha_g(z) \geq 1$



Rotors effects

- Each rotor may be thought of as a rigid disc rotating around the vertical axis the body frame, with angular velocity s_i. The rotor's axis of rotation is itself moving with the angular velocity of the frame. This leads to the following **gyroscopic torque** :

$$\vec{\Gamma}_{\text{gyro}} = I_r \vec{\omega} \times \vec{t}_3 \sum_i (-1) i \, |s_i|$$

  - I_r is the inertia matrix of a rotor

Rotors effects

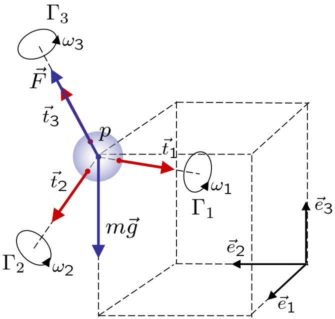
- Each rotor may be thought of as a rigid disc rotating around the vertical axis the body frame, with angular velocity s_i. The rotor's axis of rotation is itself moving with the angular velocity of the frame. This leads to the following **gyroscopic torque** :

$$\vec{\Gamma}_{\text{gyro}} = I_r \vec{\omega} \times \vec{t}_3 \sum_i (-1) i \, |s_i|$$

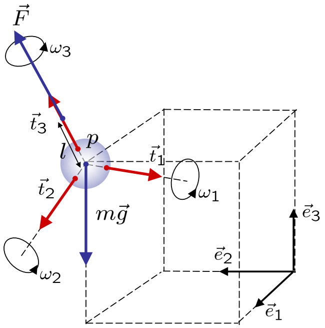
  - I_r is the inertia matrix of a rotor
- Each rotor produces a counter rotating torque that can be expressed as:

$$\begin{aligned} s_{res} &:= \sum_i (-1)^i \, |s_i| \\ \vec{\Gamma}_I &= I_r \dot{s}_{res} \vec{t}_3 \end{aligned}$$

Uncentered forces effects



Uncentered forces effects



- Superposition of thrust center and mass center
- Modified torque and forces:

$$J\dot{\vec{\omega}} = -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c + \vec{\Gamma}_{ext} + \vec{F} \times \vec{PA} \quad (1)$$

where P is the center of mass and A the point where the thrust force applies

Other effects

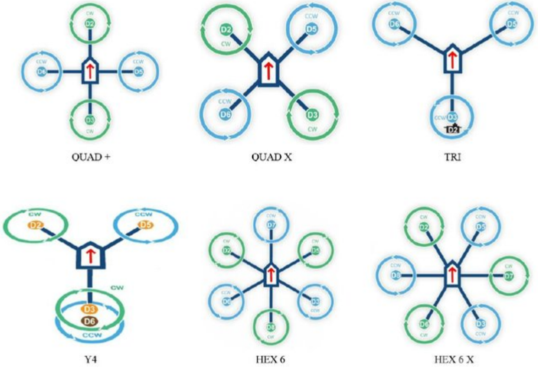
- External forces
- Air friction: $-K_v \, ||\vec{v}|| \, \vec{v}$
- Many neglected non linear effects

The mixing matrix

- The **mixing matrix** M_x links the torques and thrust force to the rotational speed of the rotors

The mixing matrix

- The **mixing matrix** M_x links the torques and thrust force to the rotational speed of the rotors
- Depends on the considered configuration (not the same for + or x configuration)



The mixing matrix

- The **mixing matrix** M_x links the torques and thrust force to the rotational speed of the rotors
- Depends on the considered configuration (not the same for + or x configuration)
- For the + configuration presented before, we have:

$$\begin{pmatrix} T \\ \Gamma_r \\ \Gamma_p \\ \Gamma_y \end{pmatrix} = \underbrace{\begin{pmatrix} c_T & c_T & c_T & c_T \\ 0 & -lc_T & 0 & lc_T \\ lc_T & 0 & -lc_T & 0 \\ lc_D & -lc_D & lc_D & -lc_D \end{pmatrix}}_{M_x} \begin{pmatrix} s_1^2 \\ s_2^2 \\ s_3^2 \\ s_4^2 \end{pmatrix}$$

The mixing matrix

- The **mixing matrix** M_x links the torques and thrust force to the rotational speed of the rotors
- Depends on the considered configuration (not the same for + or x configuration)
- For the + configuration presented before, we have:

$$\begin{pmatrix} T \\ \Gamma_r \\ \Gamma_p \\ \Gamma_y \end{pmatrix} = \underbrace{\begin{pmatrix} c_T & c_T & c_T & c_T \\ 0 & -lc_T & 0 & lc_T \\ lc_T & 0 & -lc_T & 0 \\ lc_D & -lc_D & lc_D & -lc_D \end{pmatrix}}_{M_x} \begin{pmatrix} s_1^2 \\ s_2^2 \\ s_3^2 \\ s_4^2 \end{pmatrix}$$

- Flapping and other effect renders the relation between the rotor's speeds and control thrust and torques complex

The mixing matrix

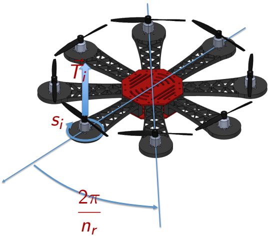
- The **mixing matrix** M_x links the torques and thrust force to the rotational speed of the rotors
- Depends on the considered configuration (not the same for $+$ or x configuration)
- For the $+$ configuration presented before, we have:

$$\begin{pmatrix} T \\ \Gamma_r \\ \Gamma_p \\ \Gamma_y \end{pmatrix} = \underbrace{\begin{pmatrix} c_T & c_T & c_T & c_T \\ 0 & -lc_T & 0 & lc_T \\ lc_T & 0 & -lc_T & 0 \\ lc_D & -lc_D & lc_D & -lc_D \end{pmatrix}}_{M_x} \begin{pmatrix} s_1^2 \\ s_2^2 \\ s_3^2 \\ s_4^2 \end{pmatrix}$$

- Flapping and other effect renders the relation between the rotor's speeds and control thrust and torques complex
- With flapping appears coupling phenomenon: the thrust affects the yaw movement and the drag affects thrust/roll/pitch movements

The mixing matrix: general case

- Consider an UAV with $n_r > 3$ rotors

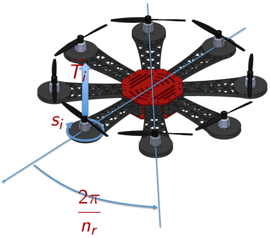


- where $\sigma_i = 1$ if the direction of rotation of the i^{th} rotor is clockwise and $\sigma_i = -1$ if it is counterclockwise
- If the number of rotors is even, $\sigma_{i+1} = -\sigma_i$.
- When the number of rotors is odd, $\sigma_{i+1} = -\sigma_i$ except for $i = \dfrac{n_r - 1}{2}$ where $\sigma_{\frac{n_r-1}{2}+1} = \sigma_{\frac{n_r-1}{2}}$.

The mixing matrix: general case

- Consider an UAV with $n_r > 3$ rotors
- Each rotor rotation speed is s_i and generates a single thrust and torque:

$$T_i = c_T s_i^2$$
$$Q_i = lc_D s_i^2$$



- where $\sigma_i = 1$ if the direction of rotation of the i^{th} rotor is clockwise and $\sigma_i = -1$ if it is counterclockwise
- If the number of rotors is even, $\sigma_{i+1} = -\sigma_i$.
- When the number of rotors is odd, $\sigma_{i+1} = -\sigma_i$ except for $i = \frac{n_r - 1}{2}$ where $\sigma_{\frac{n_r-1}{2}+1} = \sigma_{\frac{n_r-1}{2}}$.

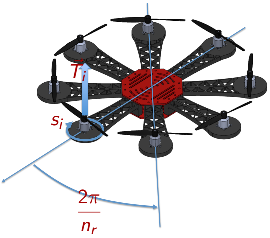
The mixing matrix: general case

- Consider an UAV with $n_r > 3$ rotors
- Each rotor rotation speed is s_i and generates a single thrust and torque:

$$T_i = c_T s_i^2$$
$$Q_i = l c_D s_i^2$$

- The thrust and torques are then:

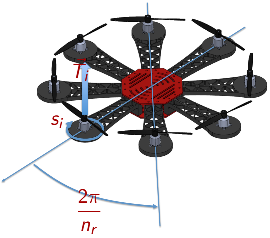
$$\Gamma_r = l \sum_{i=1}^{n_r} \sin\left[\frac{2\pi(i-1)}{n_r}\right] T_i$$
$$\Gamma_p = l \sum_{i=1}^{n_r} \cos\left[\frac{2\pi(i-1)}{n_r}\right] T_i$$
$$\Gamma_y = \sum_{i=1}^{n_r} \sigma_i Q_i$$
$$T = \sum_{i=1}^{n_r} T_i$$



- where $\sigma_i = 1$ if the direction of rotation of the i^{th} rotor is clockwise and $\sigma_i = -1$ if it is counterclockwise
- If the number of rotors is even, $\sigma_{i+1} = -\sigma_i$.
- When the number of rotors is odd, $\sigma_{i+1} = -\sigma_i$ except for $i = \frac{n_r - 1}{2}$ where $\sigma_{\frac{n_r-1}{2}+1} = \sigma_{\frac{n_r-1}{2}}$.

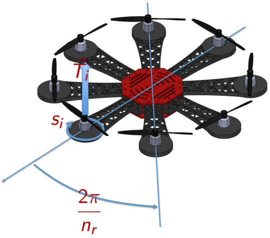
The mixing matrix: general case

- Consider an UAV with $n_r > 3$ rotors



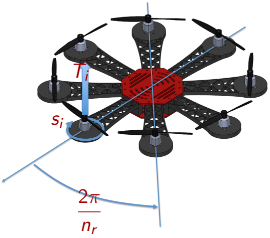
The mixing matrix: general case

- Consider an UAV with $n_r > 3$ rotors
- Each rotor rotation speed is s_i and generates a single thrust and torque



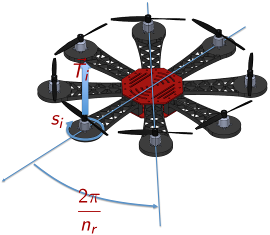
The mixing matrix: general case

- Consider an UAV with $n_r > 3$ rotors
- Each rotor rotation speed is s_i and generates a single thrust and torque
- The thrust and torques can be computed as functions of the s_i



The mixing matrix: general case

- Consider an UAV with $n_r > 3$ rotors
- Each rotor rotation speed is s_i and generates a single thrust and torque
- The thrust and torques can be computed as functions of the s_i



- The mixing matrix is:

$$\begin{pmatrix} \Gamma_r \\ \Gamma_p \\ \Gamma_y \\ T \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \dots & c_T \sin\left[\frac{2\pi(i-1)}{n_r}\right] & \dots & c_T \sin\left[\frac{2\pi(n_r-1)}{n_r}\right] \\ c_T & \dots & c_T \cos\left[\frac{2\pi(i-1)}{n_r}\right] & \dots & c_T \cos\left[\frac{2\pi(n_r-1)}{n_r}\right] \\ c_D\sigma_1 & \dots & c_D\sigma_i & \dots & c_D\sigma_{n_r} \\ c_T & \dots & c_T & \dots & c_T \end{pmatrix}}_{:=\Xi} \begin{pmatrix} s_1^2 \\ \vdots \\ s_{n_r}^2 \end{pmatrix}$$

$$= \sum_{i=1}^{n_r} \Xi_i s_i^2$$

Complete model

- **Actuation**: depends upon the type of electrical drive you use

Complete model

- **Actuation**: depends upon the type of electrical drive you use
- **Body**:

$$\left\{\begin{array}{rcl} \dot{\vec{p}} & = & \vec{v} \\ m\dot{\vec{v}} & = & -mg\,\vec{e}_3 - K_v\,\|\vec{v}\|\,\vec{v} + R\,\vec{T} + \vec{F}_{ext} \\ \dot{R} & = & R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} & = & -\vec{\omega}^{\times}J\vec{\omega} + I_r\dot{s}_{res}\vec{t}_3 + I_r\vec{\omega}\times\vec{t}_3\sum_i(-1)i\,|s_i| + \vec{\Gamma}_c + \vec{\Gamma}_{ext} \end{array}\right.$$

Complete model

- **Actuation**: depends upon the type of electrical drive you use
- **Body**:

$$\left\{\begin{array}{rcl} \dot{\vec{p}} & = & \vec{v} \\ m\dot{\vec{v}} & = & -mg\,\vec{e}_3 - K_v\,||\vec{v}||\,\vec{v} + R\,\vec{T} + \vec{F}_{ext} \\ \dot{R} & = & R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} & = & -\vec{\omega}^{\times} J\vec{\omega} + I_r\dot{s}_{res}\vec{t}_3 + I_r\vec{\omega}\times\vec{t}_3\sum_i(-1)i\,|s_i| + \vec{\Gamma}_c + \vec{\Gamma}_{ext} \end{array}\right.$$

- **Thrust**:

$$\vec{T} = \sum_i R_i^{\text{flapping}}\alpha_g c_T s_i^2 \vec{t}_3$$

Complete model

- **Actuation**: depends upon the type of electrical drive you use
- **Body**:

$$\begin{cases} \dot{\vec{p}} &= \vec{v} \\ m\dot{\vec{v}} &= -mg\,\vec{e}_3 - K_v\,||\vec{v}||\,\vec{v} + R\,\vec{T} + \vec{F}_{ext} \\ \dot{R} &= R\vec{\omega}^\times \\ J\dot{\vec{\omega}} &= -\vec{\omega}^\times J\vec{\omega} + I_r\dot{s}_{res}\vec{t}_3 + I_r\vec{\omega}\times\vec{t}_3\sum_i(-1)i\,|s_i| + \vec{\Gamma}_c + \vec{\Gamma}_{ext} \end{cases}$$

- **Thrust**:

$$\vec{T} = \sum_i R_i^{\text{flapping}}\alpha_g c_T s_i^2\vec{t}_3$$

- **Torques**:

$$\vec{\Gamma}_c = \sum_i R_i^{\text{flapping}}\alpha_g c_T s_i^2\vec{t}_3 \times p_{rotor_i}^{\mathcal{T}} + \sum_i(-1)^{i+1}c_D s_i^2\vec{t}_3$$

Towards control (1/2)

- Controlling a complex system often resume in finding *intermediate control variables*

Towards control (1/2)

- Controlling a complex system often resume in finding *intermediate control variables*
  - Actuator control (usually a voltage) u

Towards control (1/2)

- Controlling a complex system often resume in finding *intermediate control variables*
  - Actuator control (usually a voltage) u
  - Variables to control (usually a speed or a position) p

Towards control (1/2)

- Controlling a complex system often resume in finding *intermediate control variables*
  - Actuator control (usually a voltage) u
  - Variables to control (usually a speed or a position) p
  - Intermediate control variables C

$$u \rightleftarrows C \rightleftarrows p$$

  - Actuator dynamics

Towards control (1/2)

- Controlling a complex system often resume in finding *intermediate control variables*
  - Actuator control (usually a voltage) u
  - Variables to control (usually a speed or a position) p
  - Intermediate control variables C

$$u \rightleftarrows C \rightleftarrows p$$

  - Actuator dynamics
  - Robot dynamics

Towards control (1/2)

- Controlling a complex system often resume in finding *intermediate control variables*
  - Actuator control (usually a voltage) u
  - Variables to control (usually a speed or a position) p
  - Intermediate control variables C

$$u \rightleftarrows C \rightleftarrows p$$

  - Actuator dynamics
  - Robot dynamics
- Key strategy: build inner control loops to simplify the control problem
- Assume the dynamics of inner loops is neglectible w.r.t. outer ones

Towards control (2/2)

- Classic approach:
  - First control loop to control the rotation speed s_i of the blades
  - s_i and T and $\Gamma_{r,p,y}$ are linked through the mixing matrix: imposing $\vec{T}$ and $\vec{\Gamma}_c$, and s_i is the same
  - Attitude control: control the orientation of the UAV using T and $\vec{\Gamma}_c = (\Gamma_r, \Gamma_p, \Gamma_y)^T$
  - Position control/Trajectory tracking: control the position of the UAV using the orientation

Towards control (2/2)

- Classic approach:
  - First control loop to control the rotation speed s_i of the blades
  - s_i and T and $\Gamma_{r,p,y}$ are linked through the mixing matrix: imposing $\vec{T}$ and $\vec{\Gamma}_c$, and s_i is the same
  - Attitude control: control the orientation of the UAV using T and $\vec{\Gamma}_c = (\Gamma_r, \Gamma_p, \Gamma_y)^T$
  - Position control/Trajectory tracking: control the position of the UAV using the orientation
- Many alternatives, for instance:
  - First control loop to control the rotation speed s_i of the blades
  - s_i and T and $\Gamma_{r,p,y}$ are linked through the mixing matrix: imposing T and $\Gamma_{r,p,y}$, and s_i is the same
  - Control the angular velocity ω of the UAV using T and $\Gamma_{r,p,y}$
  - Speed control/Trajectory tracking: control the speed of the UAV using the angular velocity ω

Simplified model

- **Actuation**: Local inner loop, dynamics can be neglected

Simplified model

- **Actuation**: Local inner loop, dynamics can be neglected
- **Body**:

$$\begin{cases} \dot{\vec{p}} &= \vec{v} \\ m\dot{\vec{v}} &= -mg\vec{e}_3 + R\vec{T} \\ \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

Simplified model

- **Actuation**: Local inner loop, dynamics can be neglected
- **Body**:

$$\begin{cases} \dot{\vec{p}} &= \vec{v} \\ m\dot{\vec{v}} &= -mg\,\vec{e}_3 + R\,\vec{T} \\ \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

- **Control**: $\vec{T}$ and $\vec{\Gamma}_c$

Some dynamical system modeling basis

- Linear system:

$$\dot{x} = Ax + Bu$$

- General nonlinear system:

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

- Affine in the control system:

$$\dot{x} = f(x) + g(x)u$$

- x denotes the state of the system: how the system is
- u is the control variable of the system: how to *move* the system
- Every nonlinear system can be locally approximated by its linear approximation

Linear approximation (1/3)

- Consider the 1-2-3 Euler representation

Linear approximation (1/3)

- Consider the 1-2-3 Euler representation
- The rotation matrix is:

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

Linear approximation (1/3)

- Consider the 1-2-3 Euler representation
- The rotation matrix is:

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

- We assume that the s_i are controlled or at least join a given reference speed s_i^r sufficiently rapidly to neglect its dynamics

Linear approximation (1/3)

- Consider the 1-2-3 Euler representation
- The rotation matrix is:

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

- We assume that the s_i are controlled or at least join a given reference speed s_i^r sufficiently rapidly to neglect its dynamics
- We take $x := (\phi, \theta, \psi, \dot\phi, \dot\theta, \dot\psi, p^T, v^T)^T$ as state vector

Linear approximation (1/3)

- Consider the 1-2-3 Euler representation
- The rotation matrix is:

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

- We assume that the s_i are controlled or at least join a given reference speed s_i^r sufficiently rapidly to neglect its dynamics
- We take $x := (\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}, p^T, v^T)^T$ as state vector
- We take $u := (s_1, \ldots, s_{n_r})$ as control variable

Linear approximation (1/3)

- Consider the 1-2-3 Euler representation
- The rotation matrix is:

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

- We assume that the s_i are controlled or at least join a given reference speed s_i^r sufficiently rapidly to neglect its dynamics
- We take $x := (\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}, p^T, v^T)^T$ as state vector
- We take $u := (s_1, \ldots, s_{n_r})$ as control variable
- We chose a constant reference position x^r of the form $(0, 0, \psi^r, 0, 0, 0, {p^r}^T, 0)^T$: one position and one direction

Linear approximation (1/3)

- Consider the 1-2-3 Euler representation
- The rotation matrix is:

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

- We assume that the s_i are controlled or at least join a given reference speed s_i^r sufficiently rapidly to neglect its dynamics
- We take $x := (\phi, \theta, \psi, \dot\phi, \dot\theta, \dot\psi, p^T, v^T)^T$ as state vector
- We take $u := (s_1, \ldots, s_{n_r})$ as control variable
- We chose a constant reference position x^r of the form $(0, 0, \psi^r, 0, 0, 0, {p^r}^T, 0)^T$: one position and one direction
- The nominal input u^r must compensate the weight of the robot

$$\Xi u^r = \begin{pmatrix} \Gamma_r \\ \Gamma_p \\ \Gamma_y \\ T \end{pmatrix}^r = \begin{pmatrix} 0 \\ 0 \\ 0 \\ mg \end{pmatrix}$$

Linear approximation (2/3)

- Let $\tilde{x} := x - x^r$ and $\tilde{u} := u - u^r$ denote respectively the variation of the state and the control vectors

Linear approximation (2/3)

Robotics
N. Marchand
Introduction
Outline
Mechanics
Cartesian coordinates
Orientation
Frames
Newton
UAV's model
Recreational break
Legislation
EU legislation
Sub-categories of operations
Categories of UAVs
Introduction to control
Main components
Control loops
Attitude control
Position control

- Let $\tilde{x} := x - x^r$ and $\tilde{u} := u - u^r$ denote respectively the variation of the state and the control vectors
- Linearizing the system in a neighborhood of x^r, one obtains the following linear system of dimension 12:

$$\dot{\tilde{x}} = A\tilde{x} + B\Xi\tilde{u}$$

with

$$A := \begin{pmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ \begin{matrix} 0 & g & 0 \\ -g & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{pmatrix}, \quad B := \begin{pmatrix} 0_{3\times3} & 0_{3\times1} \\ J^{-1} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & \begin{matrix} 0 \\ 0 \\ \frac{1}{m} \end{matrix} \end{pmatrix},$$

$$B\Xi_i = \begin{pmatrix} 0_{3\times1} \\ J^{-1}\begin{pmatrix} lc_T \sin\left[\frac{2\pi(i-1)}{n_r}\right] \\ lc_T \cos\left[\frac{2\pi(i-1)}{n_r}\right] \\ lc_D\sigma_i \end{pmatrix} \\ 0_{3\times1} \\ 0 \\ 0 \\ \frac{lc_T}{m} \end{pmatrix}$$

Linear approximation First practical (2/3)

- First practical work: Program on MATLAB/Simulink the nonlinear model of the UAV and build a first control based on its linearization

Linear approximation First practical (2/3)

- First practical work: Program on MATLAB/Simulink the nonlinear model of the UAV and build a first control based on its linearization
- the numerical values are:

Linear approximation First practical (2/3)

- First practical work: Program on MATLAB/Simulink the nonlinear model of the UAV and build a first control based on its linearization
- the numerical values are:
  - **Motor parameters**:

parameter	description	value	unit
k_m	motor constant	4.3×10^{-3}	N.m/A
J_r	rotor inertia	3.4×10^{-5}	J.g.m^2
R	motor resistance	0.67	Ω
$k_{gearbox}$	gearbox ratio	2.7×10^{-3}	-
$\bar{U}_i$	maximal voltage	12	V

Linear approximation First practical (2/3)

- First practical work: Program on MATLAB/Simulink the nonlinear model of the UAV and build a first control based on its linearization
- the numerical values are:
  - **Motor parameters**:

parameter	description	value	unit
k_m	motor constant	4.3×10^{-3}	N.m/A
J_r	rotor inertia	3.4×10^{-5}	J.g.m^2
R	motor resistance	0.67	Ω
$k_{gearbox}$	gearbox ratio	2.7×10^{-3}	-
$\bar{U}_i$	maximal voltage	12	V

  - **Aerodynamical parameters**:

parameter	description	value
c_T	thrust coefficient	3.8×10^{-6}
c_D	drag coefficient	2.9×10^{-5}

Linear approximation First practical (2/3)

- First practical work: Program on MATLAB/Simulink the nonlinear model of the UAV and build a first control based on its linearization
- the numerical values are:
  - **Motor parameters**:

parameter	description	value	unit
k_m	motor constant	4.3×10^{-3}	N.m/A
J_r	rotor inertia	3.4×10^{-5}	J.g.m^2
R	motor resistance	0.67	Ω
$k_{gearbox}$	gearbox ratio	2.7×10^{-3}	-
$\bar{U}_i$	maximal voltage	12	V

  - **Aerodynamical parameters**:

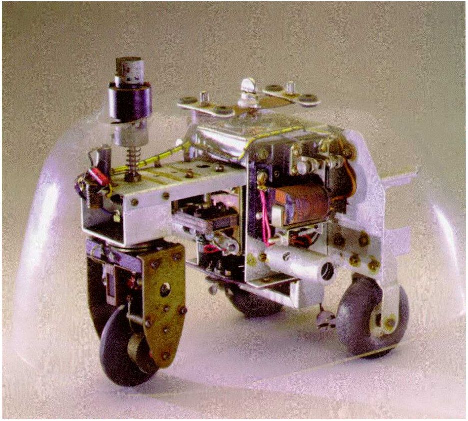
parameter	description	value
c_T	thrust coefficient	3.8×10^{-6}
c_D	drag coefficient	2.9×10^{-5}

  - **Body parameters**:

parameter	description	value	unit
J	inertia matrix	$\begin{pmatrix} 14.6 \times 10^{-3} & 0 & 0 \\ 0 & 7.8 \times 10^{-3} & 0 \\ 0 & 0 & 7.8 \times 10^{-3} \end{pmatrix}$	kg.m^2
m	mass of the UAV	0.458	kg
l	radius of the UAV	22.5	cm
g	gravity	9.81	m/s^2

Mobile robotics

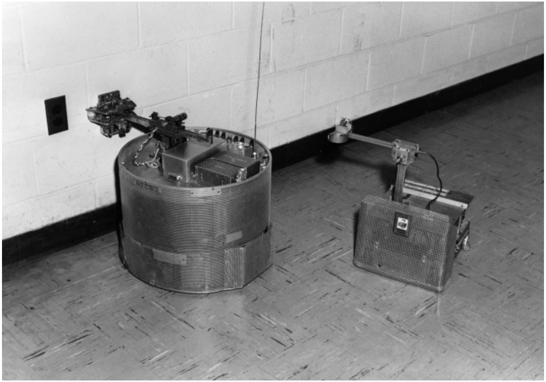
- Born in the 50s, aiming to *autonomously moving* robots



Grey Walter's "Turtle" (machina speculatrix): attracted by light

Mobile robotics

- Born in the 50s, aiming to autonomous mobile robots



John Hopkins Univ. "Beast" robot: first use of transistor based sensing (ultrasound and photodiodes)

Mobile robotics

- Born in the 50s, aiming to autonomous mobile robots

Shakey robot from Stanford Univ.
Platform used to show first results on AI (1969)

Mobile robotics

- Bio inspired locomotion: first biped robot

Honda E0 first biped robot (1986)

Mobile robotics

- Bio inspired locomotion: first biped walk

Rabbit robot CNRS-Grenoble (2004)

Mobile robotics

- Bio inspired locomotion: more about mobility

Boston Dynamics (SoftBank)

Mobile robotics

- SLAM: Simultaneous localization and mapping

https://github.com/erik-nelson/blam

Mobile robotics

- SLAM: Simultaneous localization and mapping

Mobile robotics

- Aerial robotics

Mobile robotics

- Bionics

Outline

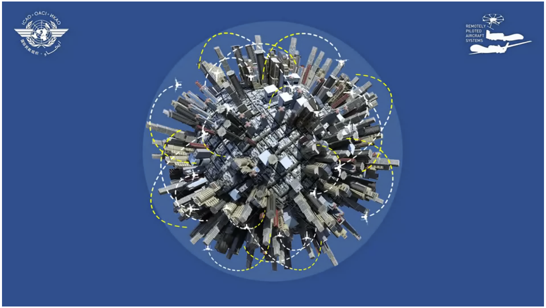
Legislation on UAVs

- Vocabulary:
  - UAS (Unmanned Aircraft System): Unmanned Aerial Vehicle + ground station + communication system
  - RPAS (Remotely Piloted Aircraft System): UAS with a remote ground pilot
- ICAO (International Civil Aviation Organization):
  - Adopts for international aviation: standards and recommended practices concerning air navigation, its infrastructure, flight inspection, prevention of unlawful interference, and facilitation of border-crossing procedures
  - In charge of RPAS since 2008
  - Creation in 2014 of the "RPAS Panel": integration of RPAS in the "IFR traffic"[a]
  - In 2016: the states member of the ICAO officially ask to give rules on how to handle RPAS
  - End 2016: ICAO produce the "UAS Toolkit" and RPAS are included in the GASP (Global Aviation Safety Plan)
  - 2015: ICAO has also to give rules for UAS... still working on it
  - To learn more: `https://bit.ly/38MpA6F`

[a] IFR = Instrument flight rules, means flights can be monitored using radars or aircraft position report when missing

Legislation on UAVs

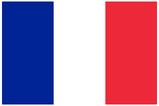
- Each state translate the ICAO recommandation into the law
- The ICAO UAS Toolkit can be consulted here: `https://bit.ly/3lwBx3U`
- Helps states to built their laws and operator to develop safe UAS
- Each state have transposed the recommandations: `https://bit.ly/3lBAvU9`



EU legislation on UAVs

- European community
  - July 4, 2018: Regulation (EU) 2018/1139: creation of the European Union Aviation Safety Agency
  - Regulations (EU) 2019/945 and (EU) 2020/105 transposed in each country legislation : 5 class of UAS
  - To learn more: `https://bit.ly/2IFeaqj` (fr) of `https://bit.ly/2IJUzpp` (eng)

- France
  - In France, legislation is detailed and explained by DGAC (Direction Générale de l'Aviation Civile)
  - The new European legislation applies as of January 1, 2020
  - The old French legislation is still applicable until July 31, 2020
  - 5 class of UAS, no more distinction between professional and non professional usage of UAS
  - A new "open class" of UAS gather professional and non professional usage of UAS below 25kg
  - To learn more: `https://cutt.ly/qg7TSy9`



EU legislation on UAVs

Main dispositions

- Three type of operations:
  - Open category: flight with direct view on the UAV, non dense geographic areas, low aerial traffic → low risk for people and goods
  - Specific category: flights not in the previous category (higher density, aerial traffic, etc.), without direct view on the UAV → moderate risk for people and goods
  - Certified category: risky operations (taxis, dangerous goods, etc.)
- UAV over 25kg
  - Only in the Specific or Certified categories
  - Immatriculation of the UAV and license plate (at least 10cm×5cm)
  - Geofensing (prevent unauthorized areas)
  - Each flight must have an authorization of the DSAC (direction de la sécurité de l'aviation civile)
  - The pilot must have an appropriate "permit"
  - Very similar to plane's legislation (flight plan, etc.)
- UAV under 25kg
  - Usually in the Open category (except for special cases)
  - 3 sub-categories of operations: A1, A2 and A3
  - 5 categories of UAVs: C0, C1, C2, C3 and C4

EU legislation for UAVs below 25kg

Sub-categories of operations

Sub-category	Distance to people	UAS	Identif. geofensing	Pilot formation
A1	Isolated people on the ground	C0 ($m < 250g$) DIY UAV ($m < 250g$)	no	• Read the manual given by the manufacturer • Formation and exam on Fox AlphaTango recommended
A1	Close to people	C1 ($m < 900g$)	yes	• Read the manual given by the manufacturer • Formation and exam on Fox AlphaTango mandatory
A2	Minimal distance to people: $30m$ $5m$ if low speed mode exists	C2 ($m < 4kg$)	yes	• Read the manual given by the manufacturer • Formation and exam on Fox AlphaTango mandatory • Mandatory autoformation (inline, declarative) • Mandatory theoretical exam, gives the "brevet d'aptitude de pilote à distance"

EU legislation for UAVs below 25kg

Sub-categories of operations

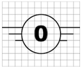
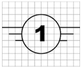
Sub-category	Distance to people	UAS, Identif. and geofensing	Pilot formation
A3	Far from people $d > 150$m from inhabitants/workers/etc	• DIY UAV ($250g < m < 25kg$): electronic identification if $m > 800g$ • C1 ($m < 900g$): electronic identification • C2 ($m < 4kg$): electronic identification • C3 ($m < 25kg$): electronic identification • C4 ($m < 25kg$): electronic identification if $m > 800g$	• Read the manual given by the manufacturer • Formation and exam on Fox AlphaTango mandatory

Main rules:

- No flight over people
- I always see my UAV (no night flight)
- I respect the maximum flight height, inhabited aircraft always have the priority
- I can have an automatic piloting system if i can take back the control at anytime
- I must be static to pilot an UAV (not in a moving vehicle)
- Nothing must fall from my UAV, i can not transport dangerous goods
- https://fox-alphatango.aviation-civile.gouv.fr/

EU legislation for UAVs below 25kg

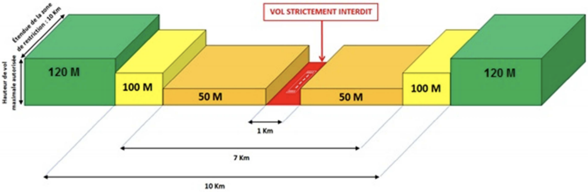
Categories of UAVs

Pictogramme d'identification	Nom de la classe	Exigences principales
	Classe C0	• Masse maximum au décollage de 250 g • Vitesse maximum verticale (vol en palier) de 19 m/s • En cas de suivi du sujet (Follow me) la distance maximum par rapport au pilote devra être de 50 m • Avoir une tension nominale ne dépassant pas 24 volts en continu
	Classe C1	• Masse maximum au décollage de 900 g • Vitesse maximum verticale (vol en palier) de 19 m/s • Comporter un numéro de série physique conforme à la norme ANSI/CTA-206 • Le drone doit permettre l'identification à distance en temps réel pendant toute la durée du vol • Etre équipé d'un système géovigilance permettant la limitation de l'espace aérien (position, altitude ⇒ Règlement (UE) 2019/947). • En cas de suivi du sujet (Follow me) la distance maximum par rapport au pilote devra être de 50 m • Etre équipé de feux (manoevrabilité, peceptibilité) • Avoir une tensesion nominale ne dépassant pas 24 volts en continu • Donner au pilote à distance un signal d'alerte clair lorsque la batterie du drone ou sa station de contrôle atteint un niveau bas

EU legislation for UAVs below 25kg

A focus on geofensing, class C1 and above

- Geofensing prevents the UAV from flying into certain areas
- Geofensing forces the UAV to stay in authorized areas
- Example: airports

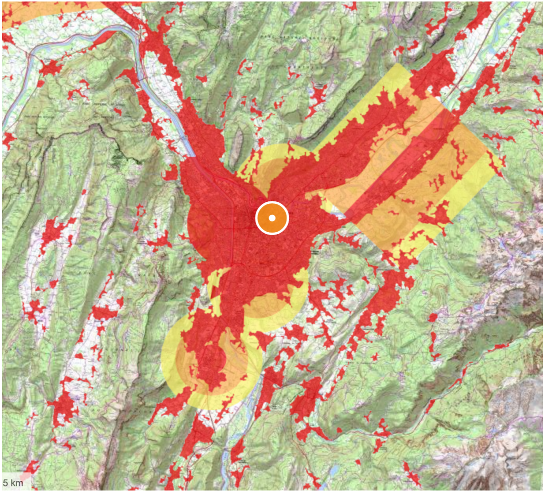


- The flying zone are divided in zones
  - forbidden zone
  - or with restricted altitude
- Can be obtained : `https://www.geoportail.gouv.fr/donnees/restrictions-pour-drones-de-loisir`

EU legislation for UAVs below 25kg

Typical flying zones

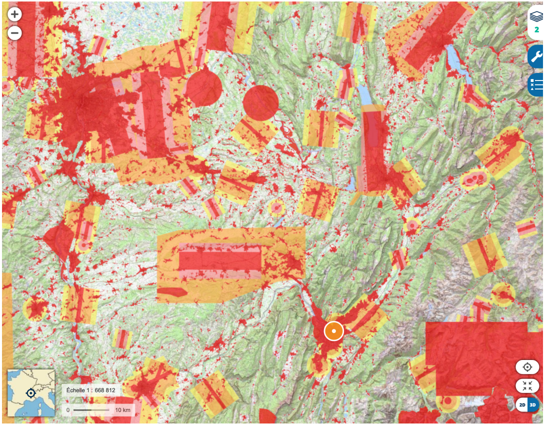
- Cities



EU legislation for UAVs below 25kg

Typical flying zones

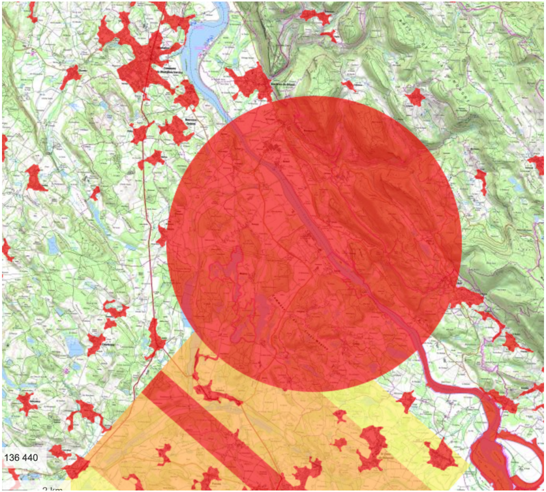
- Airports



EU legislation for UAVs below 25kg

Typical flying zones

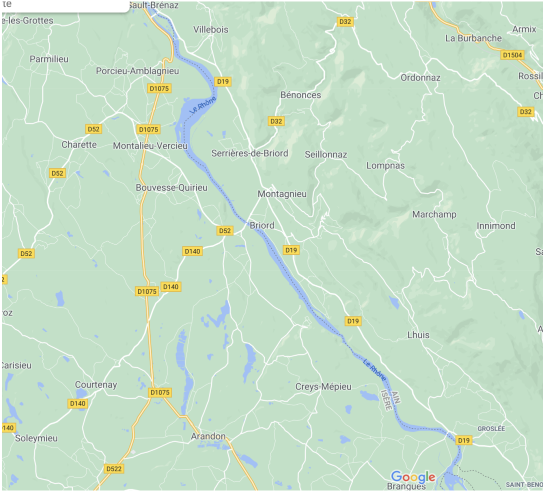
- Sensible infrastructures



EU legislation for UAVs below 25kg

Typical flying zones

- Sensible infrastructures



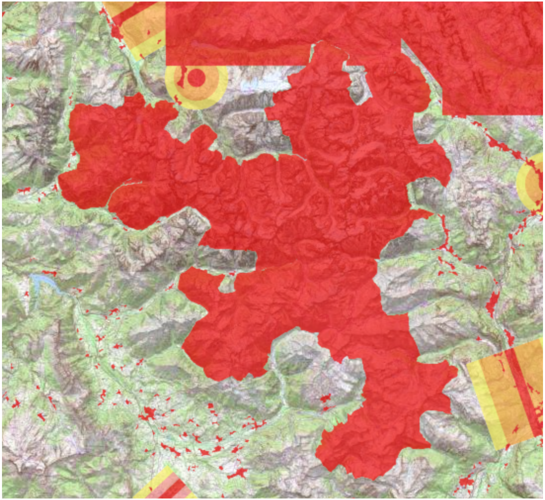
EU legislation for UAVs below 25kg

Typical flying zones

- Sensible infrastructures

EU legislation for UAVs below 25kg

Typical flying zones



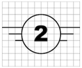
EU legislation for UAVs below 25kg

Typical flying zones

- Natural areas

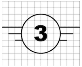
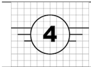
EU LEGISLATION FOR UAVS BELOW 25KG

Categories of UAVs

Pictogramme d'identification	Nom de la classe	Exigences principales
	Classe C2	• Masse maximum au décollage de 4 Kg • Etre équipé d'un mode à base vitesse sélectionnable par le pilote à distance et limitant la vitesse horizontale à 3 m/s maximum • Avoir une tension nominale ne dépassant pas 48 volts en continu • Comporter un numéro de série physique conforme à la norme ANSI/CTA-206 • Le drone doit permettre l'identification à distance en temps réel pendant toute la durée du vol • Donner au pilote à distance un signal d'alerte clair lorsque la batterie du drone ou sa station de contrôle atteint un niveau bas • Avoir un niveau de puissance acoustique LWA pondéré (*) apposée sur le drone et/ou sur son emballage • Etre équipé d'un système géovigilance permettant la limitation de l'espace aérien (position, altitude ⇒ Règlement (UE) 2019/947). • En cas de suivi du sujet (Follow me) la distance maximum par rapport au pilote devra être de 50 m • Etre équipé de feux (manoevrabilité, peceptibilité)

EU legislation for UAVs below 25kg

Categories of UAVs

Pictogramme d'identification	Nom de la classe	Exigences principales
	Classe C3	• Masse maximum au décollage de 25 Kg • Avoir un niveau de puissance acoustique LWA pondéré apposée sur le drone et/ou sur son emballage • Avoir une tension nominale ne dépassant pas 48 volts en continu • Etre équipé d'un système géovigilance permettant la limitation de l'espace aérien (position, altitude ⇒ Réglement d'éxécution (UE) 2019/947). • Le drone doit permettre l'identification à distance en temps réel pendant toute la durée du vol • Donner au pilote à distance un signal d'alerte clair lorsque la batterie du drone ou sa station de contrôle atteint un niveau bas • Comporter un numéro de série physique conforme à la norme ANSI/CTA-206
	Classe C4	• Masse maximum au décollage de 25 Kg • Ne pas être doté de modes de contrôle automatique

Outline

UAVs today

UAVs today

UAVs today

UAVs today

Main components of UAS

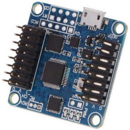
- Ground: UAV base station, remote control + ground PC
- UAV:

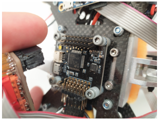
Frame

Motor (brushless most of the time)

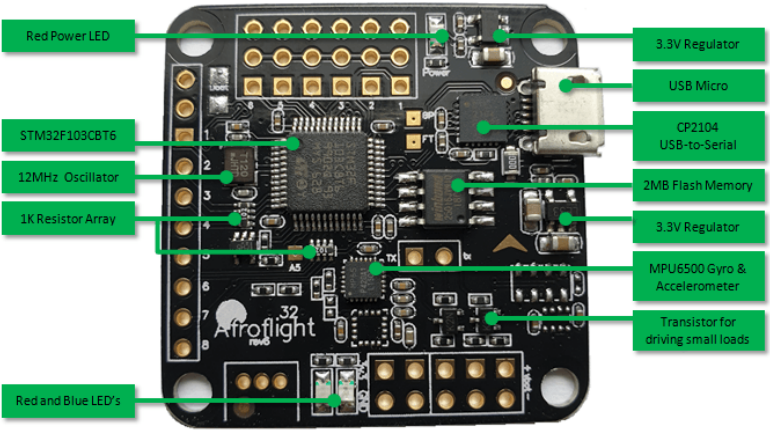
ESC (Electronic Speed Controller)

Flight controller card





Main components of UAS



Main components of UAS

- UAV:

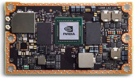
Blades

First Person View camera: fixed or pitch compensated

Camera

Battery

Image/IA card



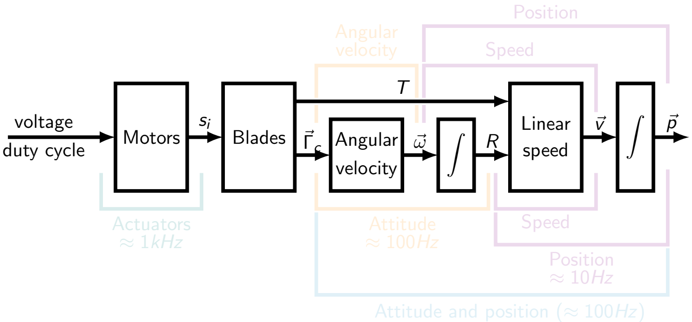
Main control loops in a UAV

- **Actuation**: Local inner loop, dynamics can be neglected. Handled by the ESC
- **Body**:

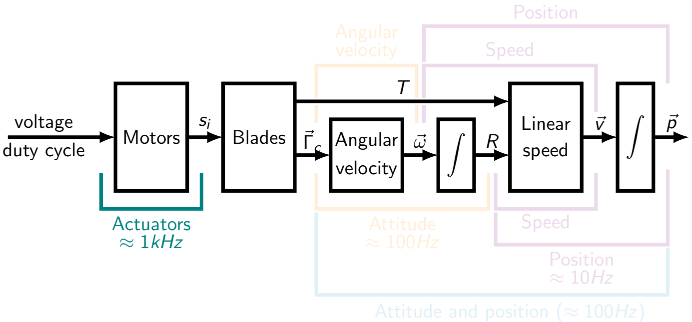
$$\begin{cases} \dot{\vec{p}} &= \vec{v} \\ m\dot{\vec{v}} &= -mg\vec{e}_3 + R\vec{T} \\ \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

- **Control**: $\vec{T}$ and $\vec{\Gamma}_c$

Main possible control loops in a UAV

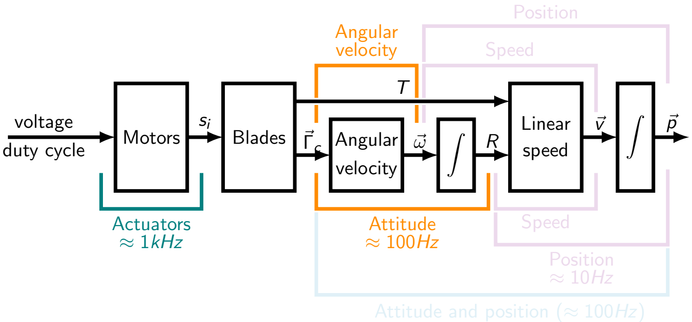


Main possible control loops in a UAV



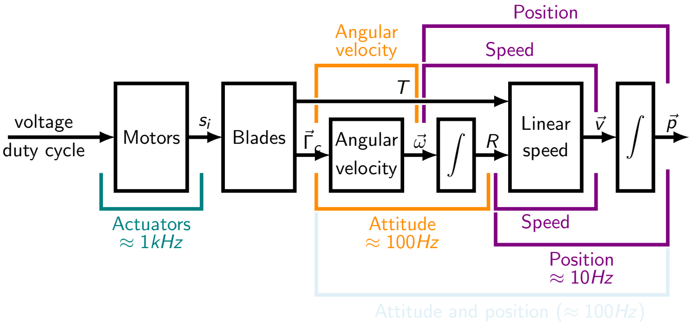
- Actuator
  - at $\approx 100Hz$
  - done by the ESC
  - open-loop / using the current as indirect measure of the speed / using a rotation speed sensor

Main possible control loops in a UAV



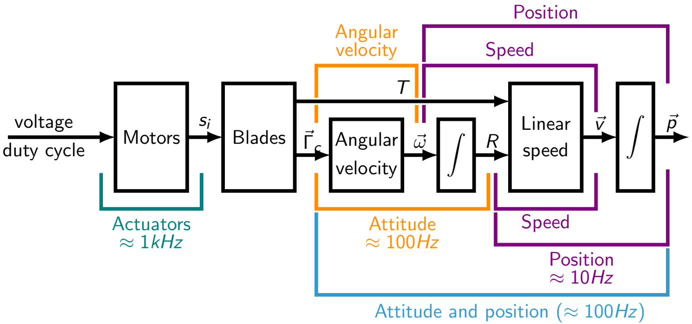
- Actuator
- Attitude
    - at $\approx 10Hz$
    - usually a PID
    - inertial sensors / control of angular velocity $\vec{\omega}$ and R using $\vec{\Gamma}_c$

Main possible control loops in a UAV



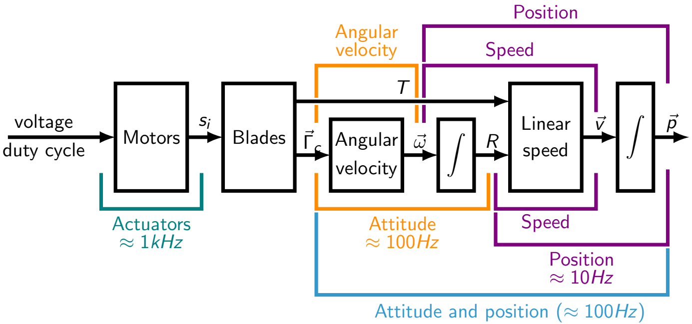
- Actuator
- Attitude
- Position
  - at $\approx 10Hz$
  - more complex control
  - control of the position, a trajectory, etc using $\vec{\omega}$ or R and $\vec{T}$

Main possible control loops in a UAV



- Actuator
- Attitude
- Position
- Attitude and position
  - at $\approx 100Hz$
  - Nonlinear approaches
  - control of the position, a trajectory, etc using directly $\vec{\Gamma}_c$ and $\vec{T}$

Main possible control loops in a UAV



- The dynamics of faster loops are neglected by slower ones
- Autonomous UAVs: control of $\vec{p}$
- Remote pilots:
  - control of R for beginners
  - control of $\vec{\omega}$ for advanced pilots

Some dynamical system modeling basis

- Linear system:

$$\dot{x} = Ax + Bu$$

- General nonlinear system:

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

- Affine in the control system:

$$\dot{x} = f(x) + g(x)u$$

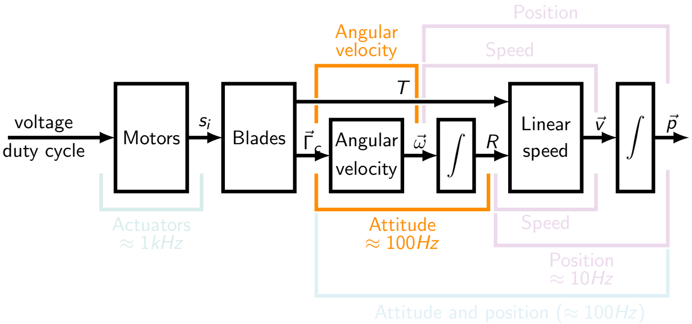
- x denotes the state of the system: how the system is
- u is the control variable of the system: how to *move* the system
- Every nonlinear system can be locally approximated by its linear approximation

Some basis of control: Lyapunov functions

- Generalized notion of energy: Lyapunov function
- A Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}^+$ is such that:
  - $V(x) > 0$ for all $x \neq 0$
  - $V(0) = 0$
- Stability and Lyapunov functions:
  - A system will converge to an equilibrium if it can only loose energy
  - or equivalently if $\dot{V}(x) < 0$ for all $x \neq 0$
- Therefore the aim will be more or less to find $u(x)$ such that

$$\dot{V} = \frac{\partial V(x)}{\partial x}\dot{x} < 0$$

Attitude control



Attitude control

Angular velocity control

$$\begin{cases} \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation

$$\begin{cases} \dot{\vec{\omega}} &= J^{-1}\vec{\Gamma}_c \quad \text{close to steady state } \vec{\omega}_{SS} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \\ \dot{\overline{\vec{\omega} - \vec{\omega}_{SS}}} &= J^{-1}\left[-\vec{\omega}_{SS}^{\times} J(\vec{\omega} - \vec{\omega}_{SS}) + \vec{\Gamma}_c\right] \quad \text{close to } \vec{\omega}_{SS} \end{cases}$$

  - Take $u := \vec{\Gamma}_c$ and $x := \vec{\omega} - \vec{\omega}_{SS}$
  - Then one has $\dot{x} = A(\omega_{SS})x + Bu$ ($B = J^{-1}$)
  - Linear system
  - Easy PID, optimal control
- Nonlinear control

Attitude control

Angular velocity control

$$\begin{cases} \dot{R} &= R\vec{\omega}^\times \\ J\dot{\vec{\omega}} &= -\vec{\omega}^\times J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
- Nonlinear control
    - Take $V = \vec{\omega}^T J\vec{\omega}$ as Lyapunov function
    - its time derivative gives:

$$\begin{aligned} \dot{V} &= \vec{\omega}^T J\dot{\vec{\omega}} + \dot{\vec{\omega}}^T J\vec{\omega} \\ &= \vec{\omega}^T[-\vec{\omega}^\times J\vec{\omega} + \vec{\Gamma}_c] + [-\vec{\omega}^\times J\vec{\omega} + \vec{\Gamma}_c]^T\vec{\omega} \\ &= 2\vec{\omega}^T\vec{\Gamma}_c \end{aligned}$$

    - Any control of the form $\vec{\Gamma}_c = -k_P\vec{\omega}$ with $k_P > 0$:
        - is such that $\dot{V} < 0$ for $\vec{\omega} \neq 0$
        - stabilizes $\vec{\omega}$ to zero
    - A P-controller stabilizes the angular velocity

Attitude control

Angular velocity control

$$\left\{\begin{array}{rcl}\dot{R} & = & R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} & = & -\vec{\omega}^{\times}J\vec{\omega}+\vec{\Gamma}_c\end{array}\right.$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
- Nonlinear control
  - Take $V=\vec{e}^T J\vec{e}$ with $\vec{e}=(\vec{\omega}-\vec{\omega}_{SS})$ as Lyapunov function
  - its time derivative gives:

$$\begin{aligned}\dot{V} &= 2\vec{e}^T J\dot{\vec{\omega}} \\ &= 2\vec{e}^T\vec{\Gamma}_c\end{aligned}$$

  - Any control of the form $\vec{\Gamma}_c=-k_P\vec{e}$ with $k_P>0$:
    - is such that $\dot{V}<0$ for $\vec{e}\neq 0$
    - stabilizes $\vec{\omega}$ to $\vec{\omega}_{SS}$
  - A P-controller stabilizes the angular velocity

Attitude control

Angular velocity control

$$\begin{cases} \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
- Nonlinear control
  - Take $V = \vec{\omega}^T J\vec{\omega} + \vec{\Omega}^T Q\vec{\Omega}$ as Lyapunov function with
    - $\vec{\Omega} = \int \vec{\omega}$ and $Q > 0$
  - its time derivative gives:

$$\begin{aligned} \dot{V} &= 2\vec{\omega}^T J\dot{\vec{\omega}} + 2\vec{\omega}^T Q\vec{\Omega} \\ &= 2\vec{\omega}^T[\vec{\Gamma}_c + Q\vec{\Omega}] \end{aligned}$$

  - Any control of the form $\vec{\Gamma}_c = -k_P\vec{\omega} - Q\vec{\Omega}$ with $k_P > 0$
    - is such that $\dot{V} < 0$ for $\vec{\omega} \neq 0$
    - stabilizes $\vec{\omega}$ and $\vec{\Omega}$ to zero
  - A PI-like controller stabilizes the angular velocity

Attitude control

Angular velocity control

$$\begin{cases} \dot{R} & = & R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} & = & -\vec{\omega}^{\times}J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
- Nonlinear control
  - Take $V = \vec{\omega}^T J\vec{\omega} + \vec{\Omega}^T Q\vec{\Omega}$ as l
    - $\vec{\Omega} = \int \vec{\omega}$ and $Q > 0$
  - its time derivative gives:

$$\begin{aligned} \dot{V} &= 2\vec{\omega}^T J\dot{\vec{\omega}} + 2\vec{\omega}^T Q\vec{\Omega} \\ &= 2\vec{\omega}^T[\vec{\Gamma}_c + Q\vec{\Omega}] \end{aligned}$$

  - Any control of the form $\vec{\Gamma}_c = -k_P\vec{\omega} - Q\vec{\Omega}$ with $k_P > 0$
    - is such that $\dot{V} < 0$ for $\vec{\omega} \neq 0$: $\mathcal{S} = \left\{(\vec{\omega},\vec{\Omega}) \text{ s.t. } \vec{\omega} = 0\right\}$
    - stabilizes $\vec{\omega}$ and $\vec{\Omega}$ to zero: $\mathcal{I} = \left\{(\vec{\omega},\vec{\Omega}) \text{ s.t. } \vec{\omega} = \vec{\Omega} = 0\right\}$
  - A PI-like controller stabilizes the angular velocity

LaSalle's Invariance principle

- V: a Lyapunov function
- $\dot{V} \leq 0$ everywhere
- $\mathcal{S} := \{x \text{ s.t. } V(x) = 0\}$
- If $\mathcal{I}$ is the largest invariant set in $\mathcal{S}$
- Then every trajectory converge to $\mathcal{I}$

Attitude control

Angular velocity control

$$\begin{cases} \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
- Nonlinear control
  - Take $V = \vec{e}^T J\vec{e} + \vec{E}^T Q\vec{E}$ as Lyapunov function with
    - $\vec{e} = \vec{\omega} - \vec{\omega}_{SS}$
    - $\vec{E} = \int \vec{e}$ and $Q > 0$
  - its time derivative gives:

$$\begin{aligned} \dot{V} &= 2\vec{e}^T J\dot{\vec{\omega}} + 2\vec{e}^T Q\vec{E} \\ &= 2\vec{e}^T[\vec{\Gamma}_c + Q\vec{E}] \end{aligned}$$

  - Any control of the form $\vec{\Gamma}_c = -k_P\vec{e} - Q\vec{E}$ with $k_P > 0$
    - is such that $\dot{V} < 0$ for $\vec{e} \neq 0$
    - stabilizes $\vec{\omega}$ and $\vec{\Omega}$ to zero
  - A PI-like controller stabilizes the angular velocity

Attitude control

Angular velocity control

$$\left\{\begin{array}{rcl}\dot{R} & = & R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} & = & -\vec{\omega}^{\times}J\vec{\omega}+\vec{\Gamma}_c\end{array}\right.$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
- Nonlinear control
  - Take $V = \vec{e}^T J\vec{e} + \vec{E}^T Q\vec{E}$ as L
    - $\vec{e} = \vec{\omega} - \vec{\omega}_{SS}$
    - $\vec{E} = \int \vec{e}$ and $Q > 0$
  - its time derivative gives:

$$\begin{array}{rcl}\dot{V} & = & 2\vec{e}^T J\dot{\vec{\omega}} + 2\vec{e}^T Q\vec{E} \\ & = & 2\vec{e}^T[\vec{\Gamma}_c + Q\vec{E}]\end{array}$$

  - Any control of the form $\vec{\Gamma}_c = -k_P\vec{e} - Q\vec{E}$ with $k_P > 0$
    - is such that $\dot{V} < 0$ for $\vec{e} \neq 0$: $\mathcal{S} = \left\{(\vec{e},\vec{E}) \text{ s.t. } \vec{e} = 0\right\}$
    - stabilizes $\vec{\omega}$ and $\vec{\Omega}$ to zero: $\mathcal{I} = \left\{(\vec{e},\vec{E}) \text{ s.t. } \vec{e} = \vec{E} = 0\right\}$
  - A PI-like controller stabilizes the angular velocity

LaSalle's Invariance principle

- V: a Lyapunov function
- $\dot{V} \leq 0$ everywhere
- $\mathcal{S} := \{x \text{ s.t. } V(x) = 0\}$
- If $\mathcal{I}$ is the largest invariant set in $\mathcal{S}$
- Then every trajectory converge to $\mathcal{I}$

Attitude control

Full attitude control

$$\begin{cases} \dot{R} &= R\vec{\omega}^\times \\ J\dot{\vec{\omega}} &= -\vec{\omega}^\times J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

$$\begin{cases} \dot{q} &= \frac{1}{2}\begin{pmatrix} -\vec{q}^T \\ I_{3\times 3}q_0 + \vec{q}^\times \end{pmatrix}\vec{\omega} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^\times J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$
$$q^T q = q_0^2 + \vec{q}^T\vec{q} = 1$$

$$\vec{\omega} = W^{-1}\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
  - Close to $\phi = \theta = \psi = 0$, $W \approx I$
  - Therefore for small angles and angular velocities, the attitude behaves like three double integrators on each roll, pitch and yaw axis
  - Easy PID, optimal control, etc.
- Nonlinear control

Attitude control

Full attitude control

$$\begin{cases} \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases} \qquad \begin{cases} \dot{q} &= \frac{1}{2}\begin{pmatrix} -\vec{q}^T \\ I_{3\times 3}q_0 + \vec{q}^{\times} \end{pmatrix}\vec{\omega} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \\ q^T q = q_0^2 + \vec{q}^T\vec{q} = 1 \end{cases} \qquad \vec{\omega} = W^{-1}\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
- Nonlinear control
  - Take $V = \frac{1}{2}\vec{\omega}^T J\vec{\omega} + k((1-q_0)^2 + \vec{q}^T\vec{q})$ as Lyapunov function
  - Note that $V = \frac{1}{2}\vec{\omega}^T J\vec{\omega} + 2k(1-q_0)$
  - its time derivative gives:

$$\begin{aligned} \dot{V} &= \vec{\omega}^T J\dot{\vec{\omega}} - 2k\dot{q}_0 \\ &= \vec{\omega}^T\vec{\Gamma}_c + k\vec{q}^T\vec{\omega} = \vec{\omega}^T\left[\vec{\Gamma}_c + k\vec{q}\right] \end{aligned}$$

  - Any control of the form $\vec{\Gamma}_c = -k_P\vec{\omega} - k\vec{q}$ with $k_P > 0$:
    - is such that $\dot{V} < 0$ for $\vec{\omega} \neq 0$
    - stabilizes $\vec{\omega}$ and $\vec{q}$ to zero
  - A PI-like controller stabilizes the attitude

Attitude control

Full attitude control

$$\begin{cases} \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases} \qquad \begin{cases} \dot{q} &= \frac{1}{2}\begin{pmatrix} -\vec{q}^T \\ I_{3\times 3} q_0 + \vec{q}^{\times} \end{pmatrix}\vec{\omega} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases} \quad q^T q = q_0^2 + \vec{q}^T\vec{q} = 1 \qquad \vec{\omega} = W^{-1}\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

- $\vec{\Gamma}_c$ as control variable
- Linear approximation
- Nonlinear control
  - Take $V = \frac{1}{2}\vec{\omega}^T J\vec{\omega} + k((1 - q_0$
  - Note that $V = \frac{1}{2}\vec{\omega}^T J\vec{\omega} + 2k(1$
  - its time derivative gives:

$$\begin{aligned} \dot{V} &= \vec{\omega}^T J\dot{\vec{\omega}} - 2k\dot{q}_0 \\ &= \vec{\omega}^T\vec{\Gamma}_c + k\vec{q}^T\vec{\omega} = \vec{\omega}^T\left[\vec{\Gamma}_c + k\vec{q}\right] \end{aligned}$$

  - Any control of the form $\vec{\Gamma}_c = -k_P\vec{\omega} - k\vec{q}$ with $k_P > 0$:
    - is such that $\dot{V} < 0$ for $\vec{\omega} \neq 0$: $\mathcal{S} = \{(\vec{\omega}, \vec{q})$ s.t. $\vec{\omega} = 0\}$
    - stabilizes $\vec{\omega}$ and $\vec{q}$ to zero : $\mathcal{I} = \{(\vec{\omega}, \vec{q})$ s.t. $\vec{\omega} = \vec{q} = 0\}$
  - A PI-like controller stabilizes the attitude

LaSalle's Invariance principle

- V: a Lyapunov function
- $\dot{V} \leq 0$ everywhere
- $\mathcal{S} := \{x$ s.t. $V(x) = 0\}$
- If $\mathcal{I}$ is the largest invariant set in $\mathcal{S}$
- Then every trajectory converge to $\mathcal{I}$

Attitude control

Full bounded attitude control

$$\begin{cases} \dot{R} &= R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases} \qquad \begin{cases} \dot{q} &= \frac{1}{2}\begin{pmatrix} -\vec{q}^T \\ I_{3\times 3}q_0 + \vec{q}^{\times} \end{pmatrix}\vec{\omega} \\ J\dot{\vec{\omega}} &= -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases} \qquad \vec{\omega} = W^{-1}\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

$$q^T q = q_0^2 + \vec{q}^T \vec{q} = 1$$

- $\vec{\Gamma}_c$ as control variable
- We want a control such that $\Gamma_{c_{1,2,3}} \in [-\bar{\Gamma}_{c_{1,2,3}}, \bar{\Gamma}_{c_{1,2,3}}]$
- Nonlinear control
  - Take $V = \frac{1}{2}\vec{\omega}^T J\vec{\omega} + 2k(1 - q_0)$ as Lyapunov function
  - its time derivative gives:

$$\begin{aligned} \dot{V} &= \vec{\omega}^T \left[\vec{\Gamma}_c + k\vec{q}\right] \\ &= \omega_1\Gamma_{c_1} + kq_1\omega_1 + \omega_2\Gamma_{c_2} + kq_2\omega_2 + \omega_3\Gamma_{c_3} + kq_3\omega_3 \\ &= \dot{V}_1 + \dot{V}_2 + \dot{V}_3 \end{aligned}$$

Attitude control

Full bounded attitude control

$$\begin{cases} \dot{R} = R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \end{cases} \qquad \begin{cases} \dot{q} = \frac{1}{2}\begin{pmatrix} -\vec{q}^T \\ I_{3\times3}q_0 + \vec{q}^{\times} \end{pmatrix}\vec{\omega} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times} J\vec{\omega} + \vec{\Gamma}_c \\ q^T q = q_0^2 + \vec{q}^T\vec{q} = 1 \end{cases} \qquad \vec{\omega} = W^{-1}\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

- $\vec{\Gamma}_c$ as control variable
- We want a control such that $\Gamma_{c_{1,2,3}} \in [-\bar{\Gamma}_{c_{1,2,3}}, \bar{\Gamma}_{c_{1,2,3}}]$
- Nonlinear control
  - Take $V = \frac{1}{2}\vec{\omega}^T J\vec{\omega} + 2k(1-q_0)$ as Lyapunov function
  - its time derivative gives:

$$\begin{aligned} \dot{V} &= \vec{\omega}^T\left[\vec{\Gamma}_c + k\vec{q}\right] \\ &= \omega_1\Gamma_{c_1} + kq_1\omega_1 + \omega_2\Gamma_{c_2} + kq_2\omega_2 + \omega_3\Gamma_{c_3} + kq_3\omega_3 \\ &= \dot{V}_1 + \dot{V}_2 + \dot{V}_3 \end{aligned}$$

  - Any control of the form $\Gamma_{c_i} = -\text{sat}_{\bar{\Gamma}_{c_i}}(k_P\omega_i + kq_i)$ with $k_P > 0$:
    - is such that $\dot{V}_i < 0$ for $\omega_i \notin \mathcal{J} := \left[-\frac{k}{k_P}, \frac{k}{k_P}\right] \Rightarrow \omega_i \to \mathcal{J}$
    - $\omega_i \in \mathcal{J} \Rightarrow k_P\omega_i \in [-k,k] \Rightarrow k_P\omega_i + kq_i \in [-2k, 2k] \xRightarrow{\text{if } 2k \leq \bar{\Gamma}_{c_i}} k_P\omega_i + kq_i \in [-\bar{\Gamma}_{c_i}, \bar{\Gamma}_{c_i}]$
    - The control is not saturated: $\Gamma_{c_i} = -(k_P\omega_i + kq_i)$
    - $\dot{V} = -k_P\vec{\omega}^T\vec{\omega} < 0$ for $\omega \neq 0$

Attitude control

Full bounded attitude control

$$\begin{cases} \dot{R} = R\vec{\omega}^{\times} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times}J\vec{\omega} + \vec{\Gamma}_c \end{cases}$$

$$\begin{cases} \dot{q} = \frac{1}{2}\begin{pmatrix} -\vec{q}^T \\ I_{3\times3}q_0 + \vec{q}^{\times} \end{pmatrix}\vec{\omega} \\ J\dot{\vec{\omega}} = -\vec{\omega}^{\times}J\vec{\omega} + \vec{\Gamma}_c \end{cases} \quad q^Tq = q_0^2 + \vec{q}^T\vec{q} = 1$$

$$\vec{\omega} = W^{-1}\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

- $\vec{\Gamma}_c$ as control variable
- We want a control such that $\Gamma_{c_{1,2,}}$
- Nonlinear control
  - Take $V = \frac{1}{2}\vec{\omega}^T J\vec{\omega} + 2k(1 - q_0)$
  - its time derivative gives:

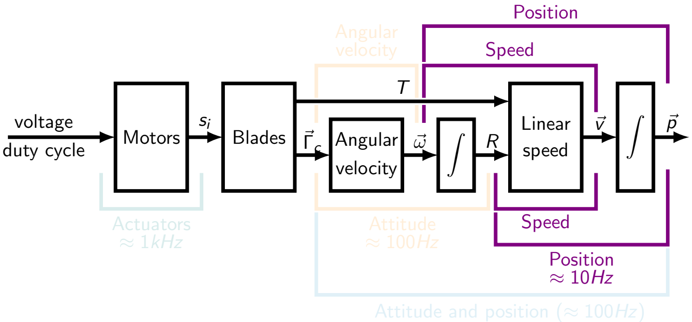
$$\dot{V} = \vec{\omega}^T\left[\vec{\Gamma}_c\right.$$
$$= \omega_1\Gamma_{c_1} +$$
$$= \dot{V}_1 + \dot{V}_2 + V_3$$

LaSalle's Invariance principle

- V: a Lyapunov function
- $\dot{V} \leq 0$ everywhere
- $\mathcal{S} := \{x \text{ s.t. } V(x) = 0\}$
- If $\mathcal{I}$ is the largest invariant set in $\mathcal{S}$
- Then every trajectory converge to $\mathcal{I}$

- Any control of the form $\Gamma_{c_i} = -\text{sat}_{\bar{\Gamma}_{c_i}}(k_P\omega_i + kq_i)$ with $k_P > 0$:
  - is such that $\dot{V}_i < 0$ for $\omega_i \notin \mathcal{J} := \left[-\frac{k}{k_P}, \frac{k}{k_P}\right] \Rightarrow \omega_i \to \mathcal{J}$
  - $\omega_i \in \mathcal{J} \Rightarrow k_P\omega_i \in [-k, k] \Rightarrow k_P\omega_i + kq_i \in [-2k, 2k] \xrightarrow{\text{if } 2k \leq \bar{\Gamma}_{c_i}} k_P\omega_i + kq_i \in \left[-\bar{\Gamma}_{c_i}, \bar{\Gamma}_{c_i}\right]$
  - The control is not saturated: $\Gamma_{c_i} = -(k_P\omega_i + kq_i)$
  - $\dot{V} = -k_P\vec{\omega}^T\vec{\omega} < 0$ for $\omega \neq 0$
  - stabilizes $\vec{\omega}$ and $\vec{q}$ to zero
- A saturated PI-like controller stabilizes the attitude

Speed/position control



Speed control

- We assume that the attitude loop is *infinitely* fast: $R \rightarrow R_d$ (subscript d stands for "desired")
- The model becomes:

$$\left\{ m\dot{\vec{v}} = -mg\vec{e}_3 + R_d\vec{T} \right.$$

- R_d is a control variable (as T)
- Detailing the equations:

$$\left\{ \begin{array}{lcl} \dot{v}_x & = & -\left[\cos\varphi_d \sin\theta_d \cos\psi_d + \sin\varphi_d \sin\psi_d)\right] T/m := u_x \\ \dot{v}_y & = & -\left[\cos\varphi_d \sin\theta_d \sin\psi_d - \sin\varphi_d \cos\psi_d)\right] T/m := u_y \\ \dot{v}_z & - & -g + \left[\cos\varphi_d \cos\theta_d\right] T/m := u_z \end{array} \right.$$

- Assume i could choose (u_x, u_y, u_z) the way i would like :

$$\left\{ \begin{array}{lcl} u_x & = & -k_x(v_x - v_x^d) \\ u_y & = & -k_y(v_y - v_y^d) \\ u_z & = & -k_z(v_z - v_z^d) \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} v_x \rightarrow v_x^d \\ v_y \rightarrow v_y^d \\ v_z \rightarrow v_z^d \end{array} \right.$$

with $k > 0$
- Take

$$\begin{array}{lcl} \alpha := \sin\varphi_d & \Rightarrow & \cos\varphi_d = \pm\sqrt{1-\alpha^2} \\ \beta := \sin\theta_d & \Rightarrow & \cos\theta_d = \pm\sqrt{1-\beta^2} \end{array}$$

Speed control

- Take

$$\alpha := \sin\varphi_d \quad \Rightarrow \quad \cos\varphi_d = \pm\sqrt{1-\alpha^2}$$
$$\beta := \sin\theta_d \quad \Rightarrow \quad \cos\theta_d = \pm\sqrt{1-\beta^2}$$

- It gives:

$$\begin{aligned} u_x &:= \pm\sqrt{1-\alpha^2}\beta T/m \\ u_y &:= \alpha T/m \\ u_z &:= \left(\pm\sqrt{1-\alpha^2}\cdot\mp\sqrt{1-\beta^2}\cdot T/m\right) - g \end{aligned}$$

- If $u_x \neq 0$:

$$\begin{aligned} \beta &:= \pm\left[\left(\frac{g+u_z}{u_x}\right)^2 + 1\right]^{\frac{1}{2}} \\ T &= \pm m\sqrt{\frac{u_x^2}{\beta^2} + u_y^2} \\ \alpha &= u_y \cdot \frac{m}{T} \end{aligned}$$

Speed control

- If $u_x \neq 0$:

$$\beta := \pm\left[\left(\frac{g+u_z}{u_x}\right)^2 + 1\right]^{\frac{1}{2}}$$

$$T = \pm m\sqrt{\frac{u_x^2}{\beta^2} + u_y^2}$$

$$\alpha = u_y \cdot \frac{m}{T}$$

- T is always positive therefore:

$$T = \pm m\sqrt{\frac{u_x^2}{\beta^2} + u_y^2}$$

- α is also unique and $\varphi_d = \arcsin\alpha \in [-\pi/2, \pi/2]$
- $\theta_d = \arcsin\beta \in [-\pi/2, \pi/2]$ of opposite sign of u_x

Position control

- We assume that the attitude loop is *infinitely* fast: $R \to R_d$ (subscript d stands for "desired")
- The model becomes:

$$\left\{\begin{array}{rcl} \dot{\vec{p}} & = & \vec{v} \\ m\dot{\vec{v}} & = & -mg\vec{e}_3 + R_d\vec{T} \end{array}\right.$$

- R_d is a control variable (as T)
- Detailing the equations:

$$\left\{\begin{array}{rcl} \ddot{x} & = & -\left[\cos\varphi_d \sin\theta_d \cos\psi_d + \sin\varphi_d \sin\psi_d\right)\right] T/m := u_x \\ \ddot{y} & = & -\left[\cos\varphi_d \sin\theta_d \sin\psi_d - \sin\varphi_d \cos\psi_d)\right] T/m := u_y \\ \ddot{z} & - & -g + \left[\cos\varphi_d \cos\theta_d\right] T/m := u_z \end{array}\right.$$

- Assume i could choose (u_x, u_y, u_z) the way i would like :

$$\left\{\begin{array}{rcl} u_x & = & k_{\dot{x}} v_x + k_x(x - x^d) \\ u_y & = & k_{\dot{y}} v_y + k_y(y - y^d) \\ u_z & = & k_{\dot{z}} v_z + k_z(z - z^d) \end{array}\right. \Leftrightarrow \left\{\begin{array}{l} x \to x^d \\ y \to y^d \\ z \to z^d \end{array}\right.$$

  with an appropriate choice of the controller parameters
- Take

$$\begin{array}{rcl} \alpha := \sin\varphi_d & \Rightarrow & \cos\varphi_d = \pm\sqrt{1-\alpha^2} \\ \beta := \sin\theta_d & \Rightarrow & \cos\theta_d = \pm\sqrt{1-\beta^2} \end{array}$$

Position control

- Take

$$\alpha := \sin\varphi_d \quad \Rightarrow \quad \cos\varphi_d = \pm\sqrt{1-\alpha^2}$$
$$\beta := \sin\theta_d \quad \Rightarrow \quad \cos\theta_d = \pm\sqrt{1-\beta^2}$$

- It gives:

$$\begin{aligned} u_x &:= \pm\sqrt{1-\alpha^2}\beta T/m \\ u_y &:= \alpha T/m \\ u_z &:= \left(\pm\sqrt{1-\alpha^2}\cdot\mp\sqrt{1-\beta^2}\cdot T/m\right) - g \end{aligned}$$

- If $u_x \neq 0$:

$$\begin{aligned} \beta &:= \pm\left[\left(\frac{g+u_z}{u_x}\right)^2 + 1\right]^{\frac{1}{2}} \\ T &= \pm m\sqrt{\frac{u_x^2}{\beta^2} + u_y^2} \\ \alpha &= u_y \cdot \frac{m}{T} \end{aligned}$$

Position control

- If $u_x \neq 0$:

$$\beta := \pm\left[\left(\frac{g+u_z}{u_x}\right)^2+1\right]^{\frac{1}{2}}$$

$$T = \pm m\sqrt{\frac{u_x^2}{\beta^2}+u_y^2}$$

$$\alpha = u_y \cdot \frac{m}{T}$$

- T is always positive therefore:

$$T = \pm m\sqrt{\frac{u_x^2}{\beta^2}+u_y^2}$$

- α is also unique and $\varphi_d = \arcsin\alpha \in [-\pi/2, \pi/2]$
- $\theta_d = \arcsin\beta \in [-\pi/2, \pi/2]$ of opposite sign of u_x

Trajectory tracking

- We assume that the attitude loop is *infinitely* fast: $R \to R_d$ (subscript d stands for "desired")
- The model becomes:

$$\begin{cases} \dot{\vec{p}} &= \vec{v} \\ m\dot{\vec{v}} &= -mg\vec{e}_3 + R_d\vec{T} \end{cases}$$

- R_d is a control variable (as T)
- Detailing the equations:

$$\begin{cases} \ddot{x} &= -\left[\cos\varphi_d \sin\theta_d \cos\psi_d + \sin\varphi_d \sin\psi_d\right)\right] T/m := u_x \\ \ddot{y} &= -\left[\cos\varphi_d \sin\theta_d \sin\psi_d - \sin\varphi_d \cos\psi_d\right)\right] T/m := u_y \\ \ddot{z} &- -g + \left[\cos\varphi_d \cos\theta_d\right] T/m := u_z \end{cases}$$

- Given a trajectory to track: $(x^d(t), y^d(t), z^d(t))$ and its time derivative $(v_x^d(t), v_y^d(t), v_z^d(t))$
- Assume i could choose (u_x, u_y, u_z) the way i would like :

$$\begin{cases} u_x &= k_{\dot{x}}(v_x - v_x^d) + k_x(x - x^d) \\ u_y &= k_{\dot{y}}(v_y - v_y^d) + k_y(y - y^d) \\ u_z &= k_{\dot{z}}(v_z - v_z^d) + k_z(z - z^d) \end{cases} \Leftrightarrow \begin{cases} x \to x^d(t) \\ y \to y^d(t) \\ z \to z^d(t) \end{cases}$$

with an appropriate choice of the controller parameters

- Take

$$\alpha := \sin\varphi_d \quad\Rightarrow\quad \cos\varphi_d = \pm\sqrt{1-\alpha^2}$$
$$\beta := \sin\theta_d \quad\Rightarrow\quad \cos\theta_d = \pm\sqrt{1-\beta^2}$$

Position control

- Take

$$\begin{aligned}\alpha := \sin\varphi_d &\Rightarrow \cos\varphi_d = \pm\sqrt{1-\alpha^2}\\ \beta := \sin\theta_d &\Rightarrow \cos\theta_d = \pm\sqrt{1-\beta^2}\end{aligned}$$

- It gives:

$$\begin{aligned}u_x &:= \pm\sqrt{1-\alpha^2}\beta T/m\\ u_y &:= \alpha T/m\\ u_z &:= \left(\pm\sqrt{1-\alpha^2}\cdot\mp\sqrt{1-\beta^2}\cdot T/m\right)-g\end{aligned}$$

- If $u_x \neq 0$:

$$\begin{aligned}\beta &:= \pm\left[\left(\frac{g+u_z}{u_x}\right)^2+1\right]^{\frac{1}{2}}\\ T &= \pm m\sqrt{\frac{u_x^2}{\beta^2}+u_y^2}\\ \alpha &= u_y\cdot\frac{m}{T}\end{aligned}$$

Position control

- If $u_x \neq 0$:

$$\beta := \pm\left[\left(\frac{g+u_z}{u_x}\right)^2 + 1\right]^{\frac{1}{2}}$$

$$T = \pm m\sqrt{\frac{u_x^2}{\beta^2} + u_y^2}$$

$$\alpha = u_y \cdot \frac{m}{T}$$

- T is always positive therefore:

$$T = \pm m\sqrt{\frac{u_x^2}{\beta^2} + u_y^2}$$

- α is also unique and $\varphi_d = \arcsin \alpha \in [-\pi/2, \pi/2]$
- $\theta_d = \arcsin \beta \in [-\pi/2, \pi/2]$ of opposite sign of u_x