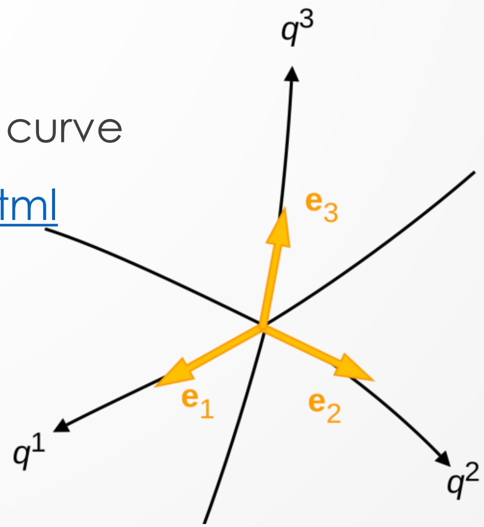
CS491/691: Introduction to Aerial Robotics

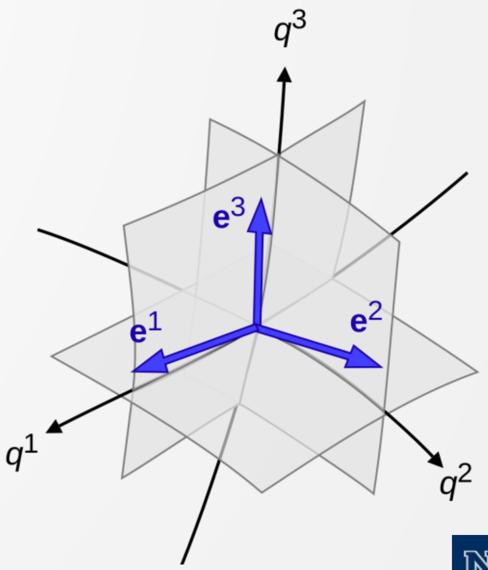
Topic: MAV Dynamics Recap

Dr. Kostas Alexis (CSE)

Elementary Background

- **Linear Algebra**
  - Critical background
  - Good overview: CS479/679: Pattern Recognition :: Review of Linear Algebra
  - We will summarize a few very common operations
- **Differential Equations**
  - Critical background
  - Good online course: Introduction to Diff Eq :: Boston University
- **Numerical tools**
  - Mostly MATLAB or Python for a smooth learning curve
  - http://www.kostasalexis.com/simulation-tools.html





Dot Product of 2 vectors

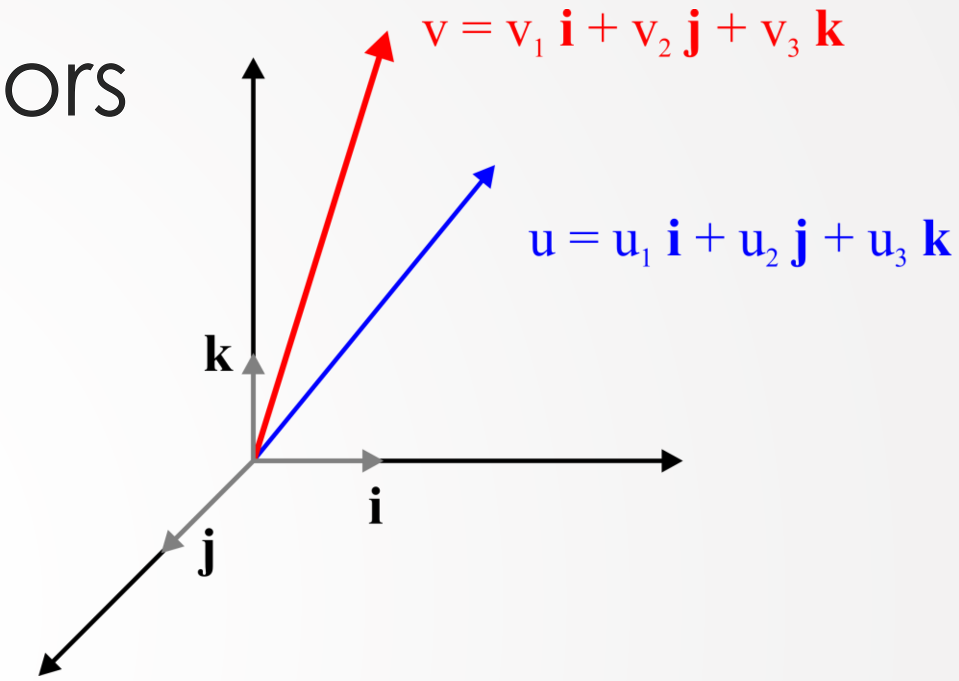
- Let:

$$\mathbf{u} = u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}$$
$$\mathbf{v} = v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k}$$

- Then:

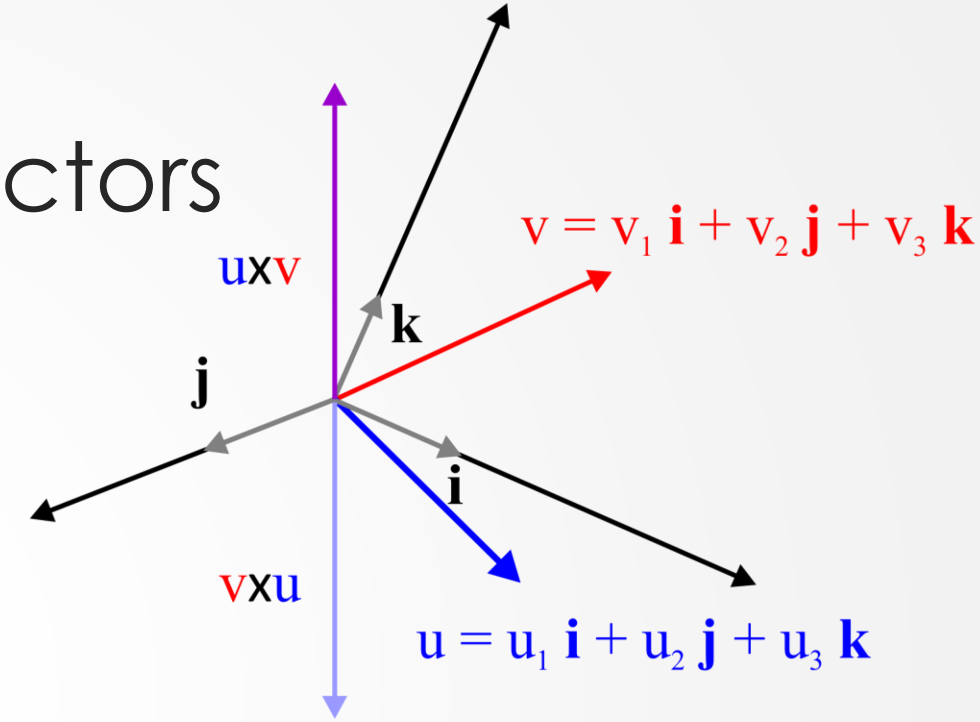
$$\mathbf{uv} = \|\mathbf{u}\|\,\|\mathbf{v}\|\cos\theta$$

$$\mathbf{uv} = u_1v_1 + u_2v_2 + u_3v_3$$



The dot product or scalar product (sometimes inner product in the context of Euclidean space, or rarely projection product for emphasizing the geometric significance), is an algebraic operation that takes two equal-length sequences of numbers (usually coordinate vectors) and returns a single number.

Cross Product of 2 vectors



- Let:

$$\mathbf{u} = u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}$$
$$\mathbf{v} = v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k}$$

- Then:

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} \Rightarrow \mathbf{u} \times \mathbf{v} = \begin{vmatrix} u_2 & u_3 \\ v_2 & v_3 \end{vmatrix}\mathbf{i} - \begin{vmatrix} u_1 & u_3 \\ v_1 & v_3 \end{vmatrix}\mathbf{j} + \begin{vmatrix} u_1 & u_2 \\ v_1 & v_2 \end{vmatrix}\mathbf{k} \Rightarrow$$

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix}$$

The cross product or vector product (occasionally directed area product to emphasize the geometric significance) is a binary operation on two vectors in three-dimensional space (R3) and is denoted by the symbol ×. Given two linearly independent vectors u and v, the cross product, u × v, is a vector that is perpendicular to both and therefore normal to the plane containing them. It has many applications in mathematics, physics, engineering, and computer programming. It should not be confused with dot product (projection product).

Outer Product of 2 vectors

- Let:

$$\mathbf{u} = [u_1, u_2, \ldots, u_m]$$
$$\mathbf{v} = [v_1, v_2, \ldots, v_n]$$

- Then:

$$\mathbf{u} \otimes \mathbf{v} = \begin{bmatrix} u_1v_1 & u_1v_2 & \cdots & u_1v_n \\ u_2v_1 & u_2v_2 & \cdots & u_2v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_mv_1 & u_mv_2 & \cdots & u_mv_n \end{bmatrix}$$

The outer product $\mathbf{u} \otimes \mathbf{v}$ is equivalent to a matrix multiplication $\mathbf{uv}^\mathbf{T}$, provided that **u** is represented as a m × 1 column vector and **v** as a n × 1 column vector (which makes vT a row vector). For instance, if m = 4 and n = 3, then

Transpose of a Matrix

In linear algebra, the transpose of a matrix A is another matrix AT (also written **A**') created by any one of the following equivalent actions:

- reflect **A** over its main diagonal (which runs from top-left to bottom-right) to obtain $\mathbf{A}^{\mathbf{T}}$
- write the rows of **A** as the columns of $\mathbf{A}^{\mathbf{T}}$
- write the columns of A as the rows of $\mathbf{A}^{\mathbf{T}}$

Formally, the i-th row, j-th column element of $\mathbf{A}^{\mathbf{T}}$ is the j-th row, i-th column element of **A**:

$$\left[\mathbf{A}^T\right]_{ij} = \left[\mathbf{A}^T\right]_{ji}$$

Example:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

Determinant of a Matrix

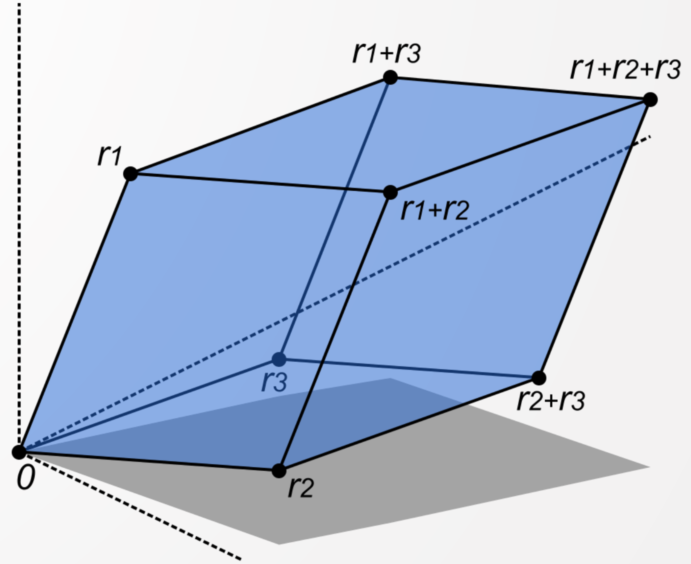
In linear algebra, the determinant is a useful value that can be computed from the elements of a square matrix. The determinant of a matrix **A** is denoted det(**A**) or |A|.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} =$$

$$a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{32}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

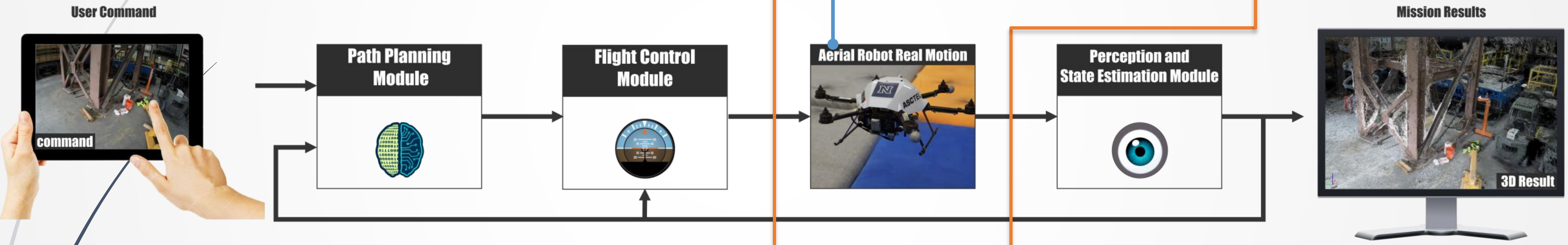
$$\begin{array}{ccc:cc} a_{11} & a_{12} & a_{13} & a_{11} & a_{12} \\ a_{21} & a_{22} & a_{23} & a_{21} & a_{22} \\ a_{31} & a_{32} & a_{33} & a_{31} & a_{32} \end{array}$$

Rule of Sarrus



The Aerial Robot Loop

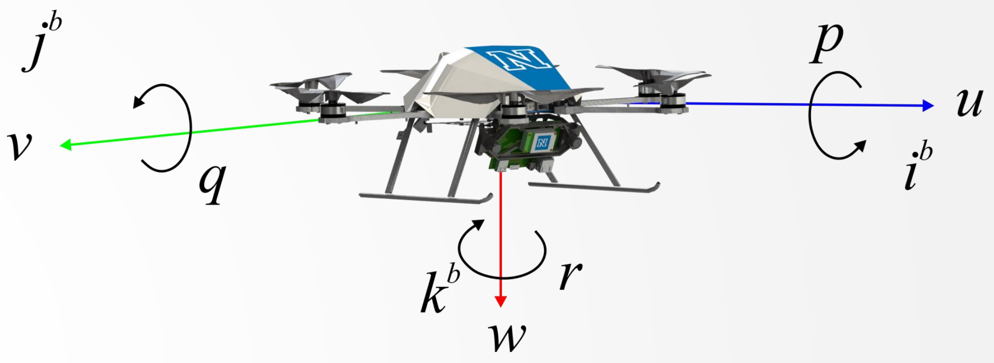
- Real-life Aerial Robot expressing its dynamic behavior in response to the control inputs and external disturbances



Section 1 of our course

Goal of this lecture

- The goal of this lecture is to derive the equations of motion that describe the motion of a multirotor Micro Aerial Vehicle.
- The MAV has 6 Degrees of Freedom but only 4 distinct inputs (it can generate 3 moments and thrust force).
  - It is an **underactuated** system.
- To achieve this goal, we rely on:
  - A model of the Aerodynamic Forces & Moments
  - A model of the motion of the vehicle body as actuated by the forces and moments acting on it.



Recap of previous lecture: http://goo.gl/MfhLNR

The MAV Propeller

- Rotor modeling is a very complicated process.
- **A Rotor is different than a propeller.** It is not-rigid and contains degrees of freedom. Among them blade flapping allows the control of the rotor tip path plane and therefore control the helicopter.

- Used to produce thrust.
- Propeller plane perpendicular to shaft.
- Rigid blade. No flapping.
- Fixed blade pitch angle or collective changes only.

- Used to produce lift and directional control.
- Elastic element between blade and shaft.
- Blade flapping used to change tip path plane.
- Blade pitch angle controlled by swashplate.

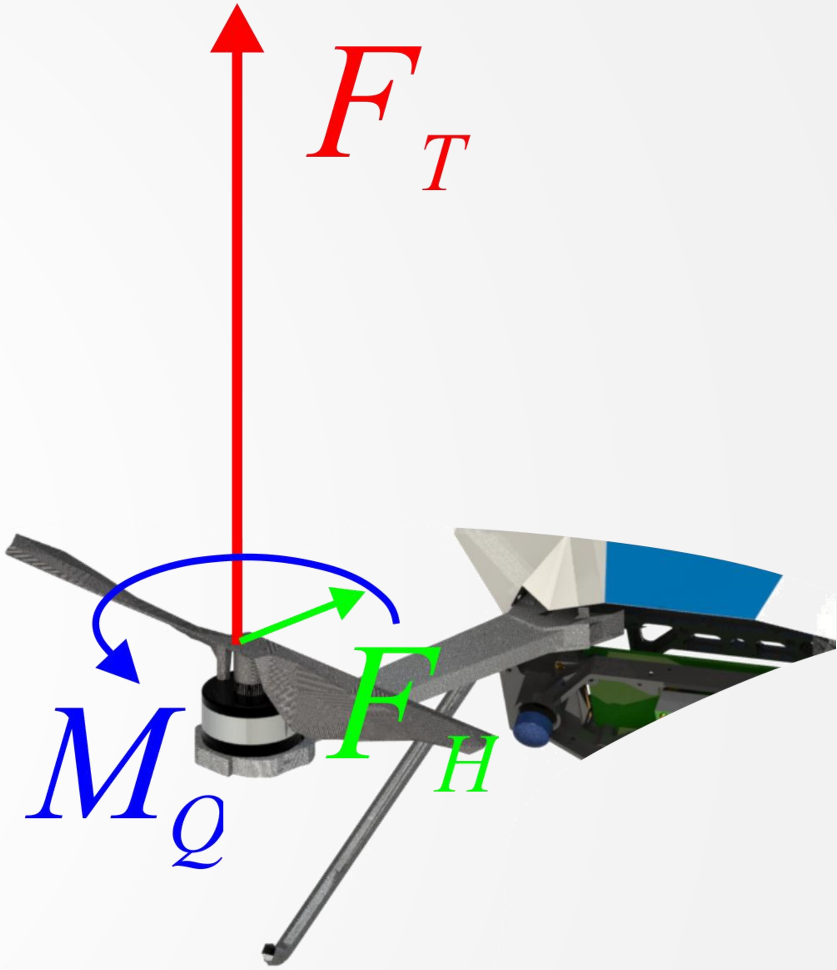
The MAV Propeller

- Simplified model forces and moments:
  - **Thrust Force:** the resultant of the vertical forces acting on all the blade elements.

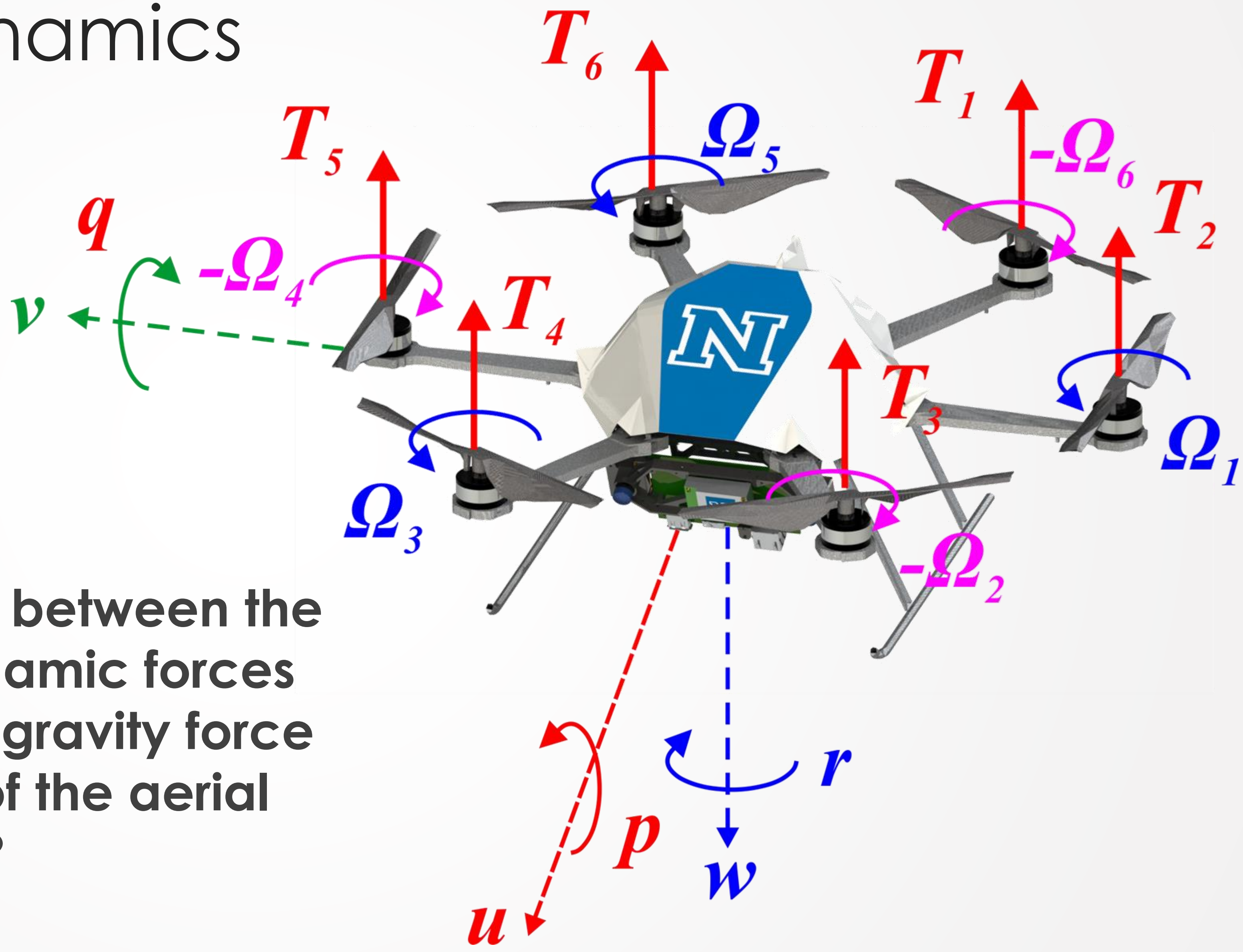
$$F_T = T = C_T \rho A (\Omega R)^2$$

  - **Drag Moment:** This moment about the rotor shaft is caused by the aerodynamic forces acting on the blade elements. The horizontal forces acting on the rotor are multiplied by the moment arm and integrated over the rotor. Drag moment determines the power required to spin the rotor.

$$M_Q = Q = C_Q \rho A (\Omega R)^2 R$$



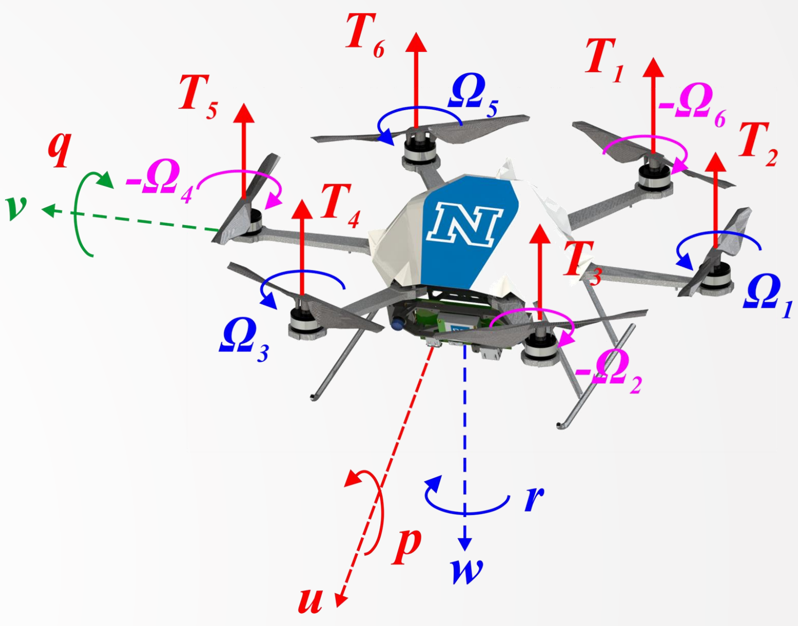
MAV Dynamics



What is the relation between the propeller aerodynamic forces and moments, the gravity force and the motion of the aerial robot?

MAV Dynamics

- Assumption 1: the Micro Aerial Vehicle is flying as a rigid body with negligible aerodynamic effects on it – for the employed airspeeds.
- The propeller is considered as a simple propeller disc that generates thrust and a moment around its shaft.



- Recall:

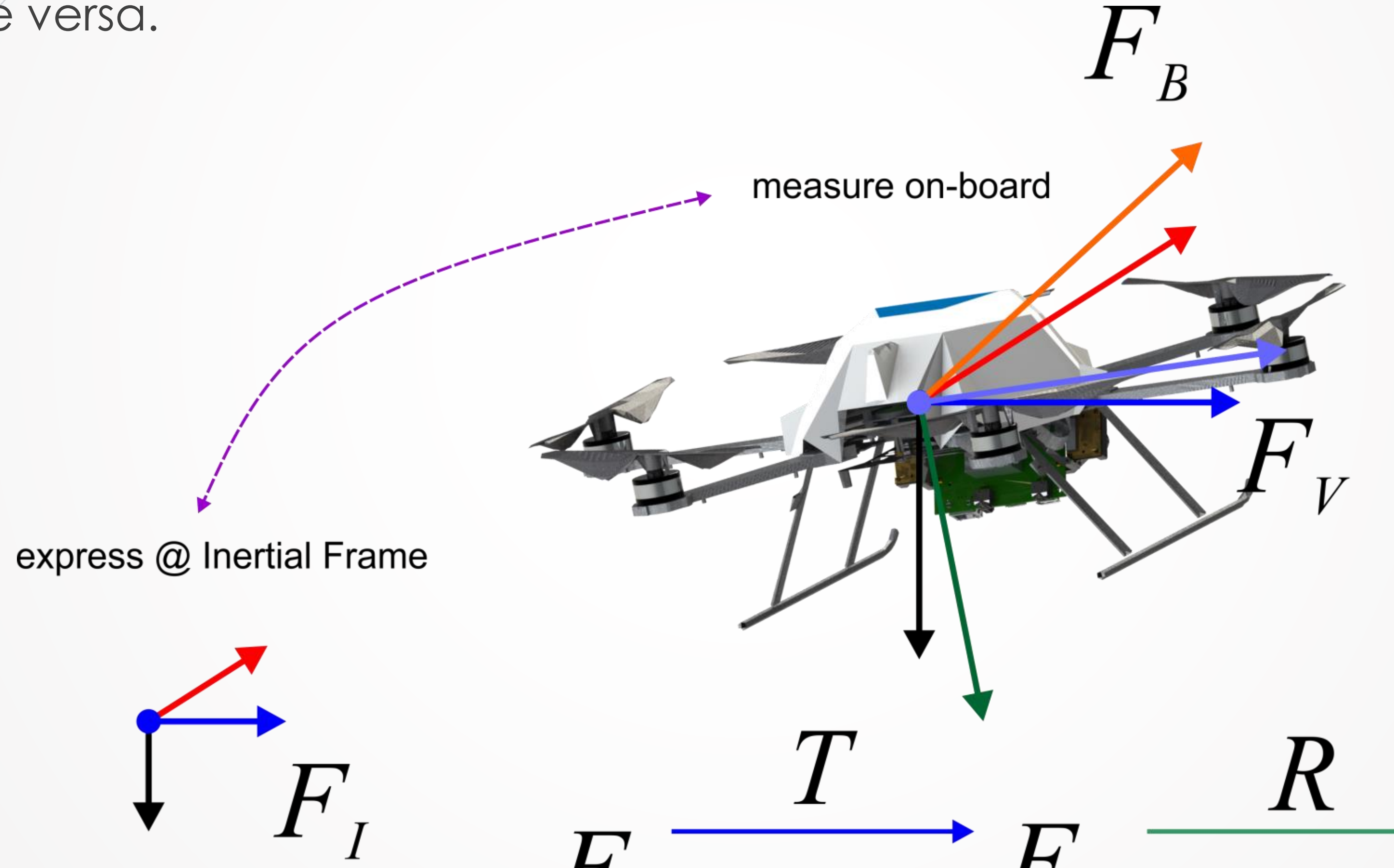
$$F_T = T = C_T \rho A (\Omega R)^2$$
$$M_Q = Q = C_Q \rho A (\Omega R)^2 R$$

- And let us write:

$$T_i = k_n \Omega_i^2$$
$$M_i = (-1)^{i-1} k_m T_i$$

MAV Dynamics

- We need to vehicle states to the inertial frame and vice versa.



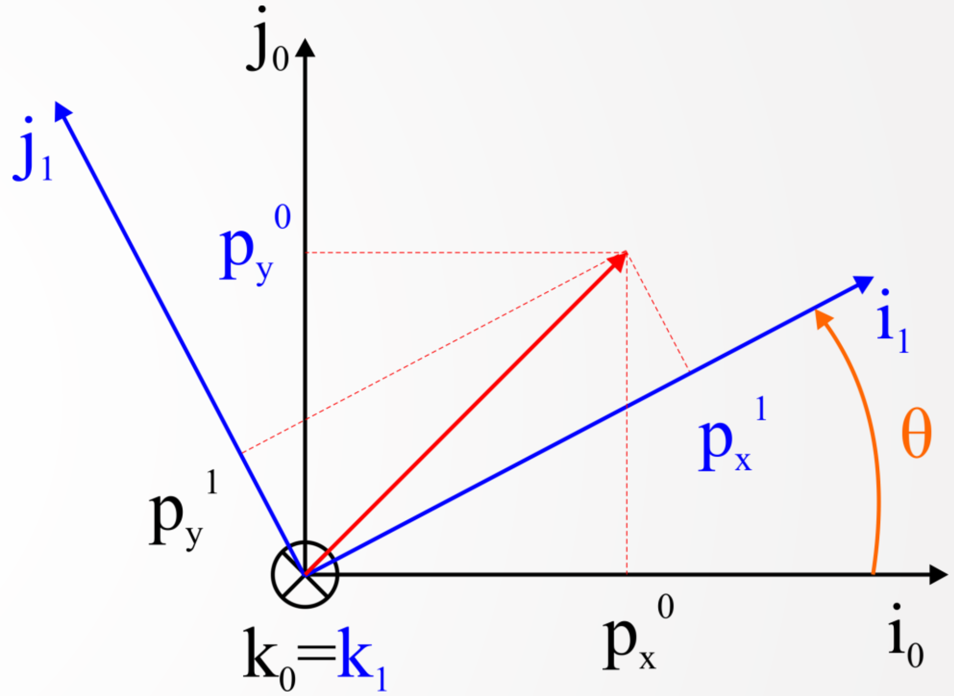
$$F_I \underset{\longleftarrow}{\xrightarrow{T}} F_V \underset{\longleftarrow}{\xrightarrow{R}} F_B$$

Reference Frame Rotatations

- Rotation around the k-axis

$$\mathbf{p} = p_x^0 \mathbf{i}^0 + p_y^0 \mathbf{j}^0 + p_z^0 \mathbf{k}^0$$
$$\mathbf{p} = p_x^1 \mathbf{i}^1 + p_y^1 \mathbf{j}^1 + p_z^1 \mathbf{k}^1$$
$$\mathbf{p}^1 = \begin{pmatrix} p_x^1 \\ p_y^1 \\ p_z^1 \end{pmatrix} = \begin{pmatrix} \mathbf{i}^1\mathbf{i}^0 & \mathbf{i}^1\mathbf{j}^0 & \mathbf{i}^1\mathbf{k}^0 \\ \mathbf{j}^1\mathbf{i}^0 & \mathbf{j}^1\mathbf{j}^0 & \mathbf{j}^1\mathbf{k}^0 \\ \mathbf{k}^1\mathbf{i}^0 & \mathbf{k}^1\mathbf{j}^0 & \mathbf{k}^1\mathbf{k}^0 \end{pmatrix} \begin{pmatrix} p_x^0 \\ p_y^0 \\ p_z^0 \end{pmatrix}$$

$$\mathbf{p}^1 = \mathcal{R}_0^1 \mathbf{p}^0, \ \mathcal{R}_0^1 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

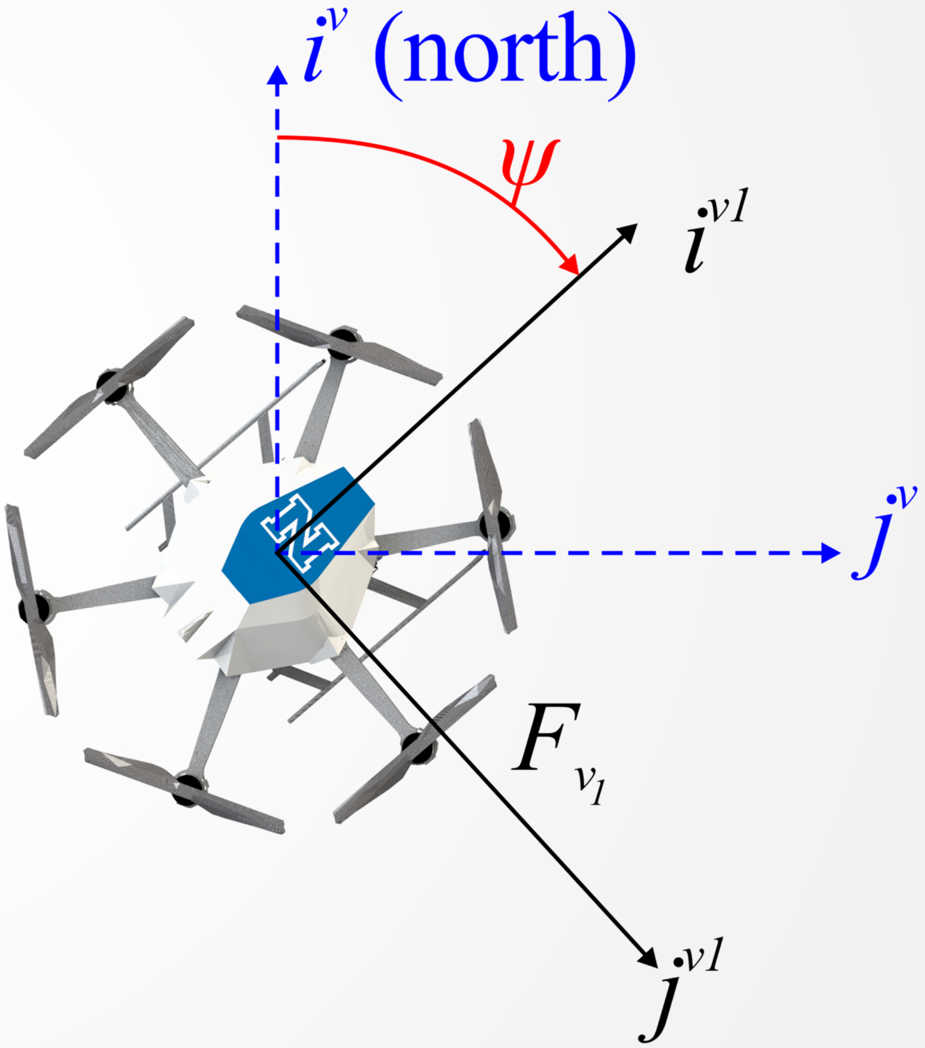


Vehicle-1 Frame

$$\mathbf{p}^{v_1} = \mathcal{R}_v^{v_1}\mathbf{p}^v,$$

$$\mathcal{R}_v^{v_1} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- ψ represents the yaw angle

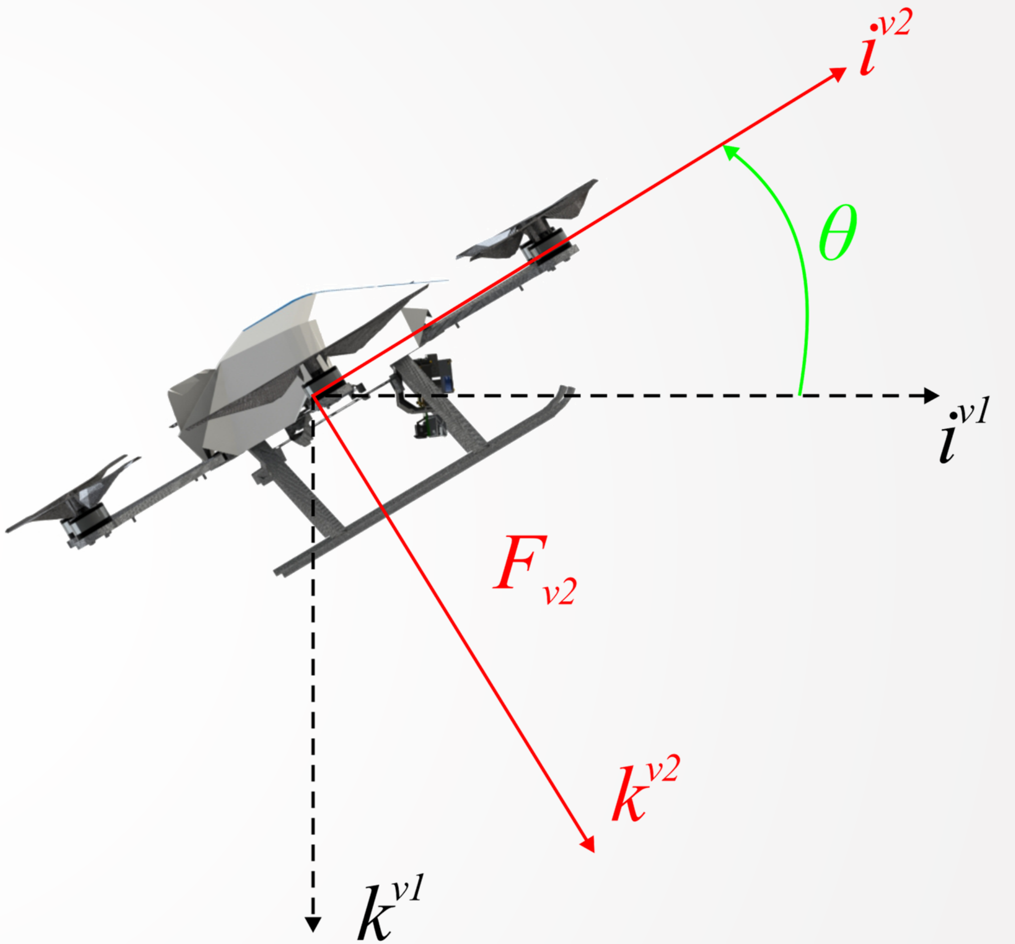


Vehicle-2 Frame

$$\mathbf{p}^{v_2} = \mathcal{R}_{v_1}^{v_2}\mathbf{p}^{v_1},$$

$$\mathcal{R}_{v_1}^{v_2} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

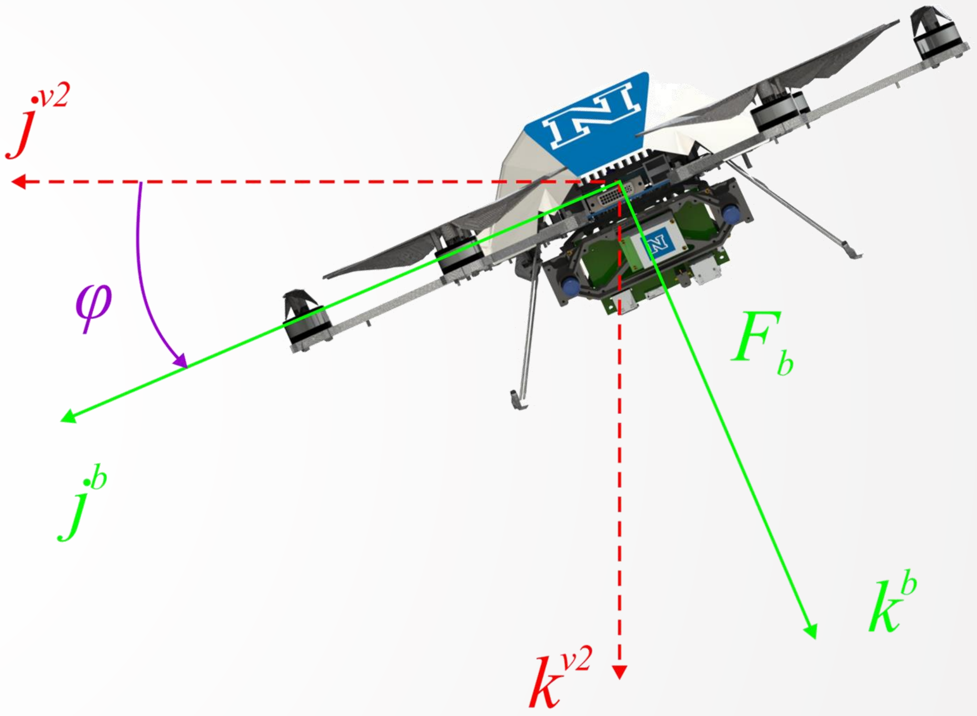
- θ represents the pitch angle



Body Frame

$$\mathbf{p}^b = \mathcal{R}_{v_2}^b \mathbf{p}^{v_2},$$

$$\mathcal{R}_{v_2}^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$



- φ represents the roll angle

Inertial Frame to Body Frame

- Let:

$$\mathcal{R}_v^b(\phi,\theta,\psi) = \mathcal{R}_{v_2}(\phi)\mathcal{R}_{v_1}^{v_2}(\theta)\mathcal{R}_v^{v_1}(\psi)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

- Then:

$$\mathbf{p}^b = \mathcal{R}_v^b \mathbf{p}^v$$

Rotation of Reference Frame

- Rotation around the i-axis

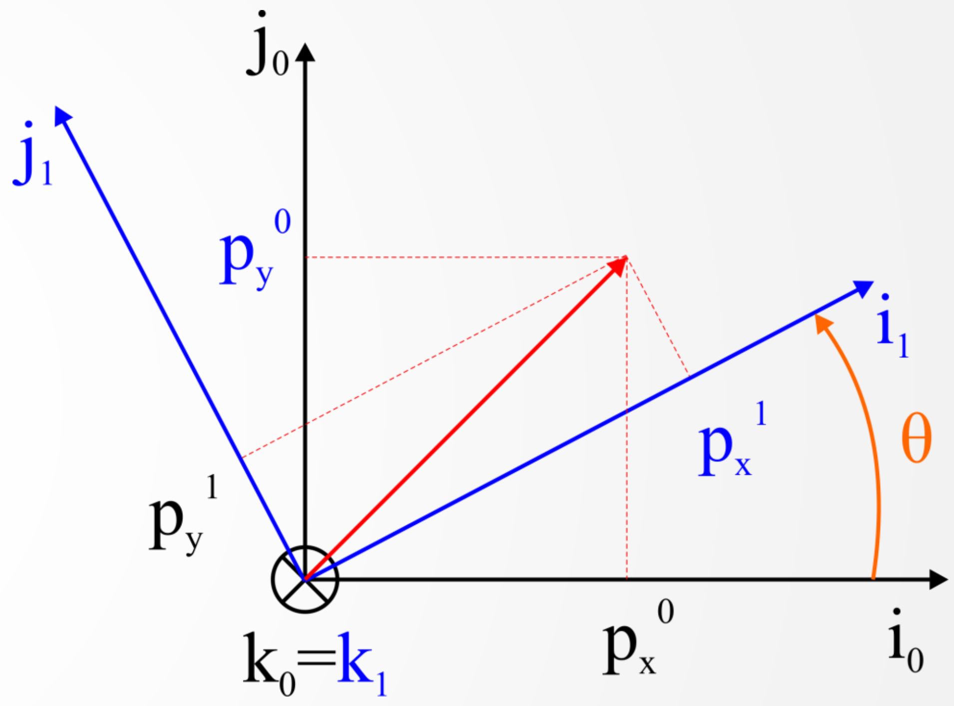
$$\mathcal{R}_0^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

- Rotation around the j-axis

$$\mathcal{R}_0^1 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

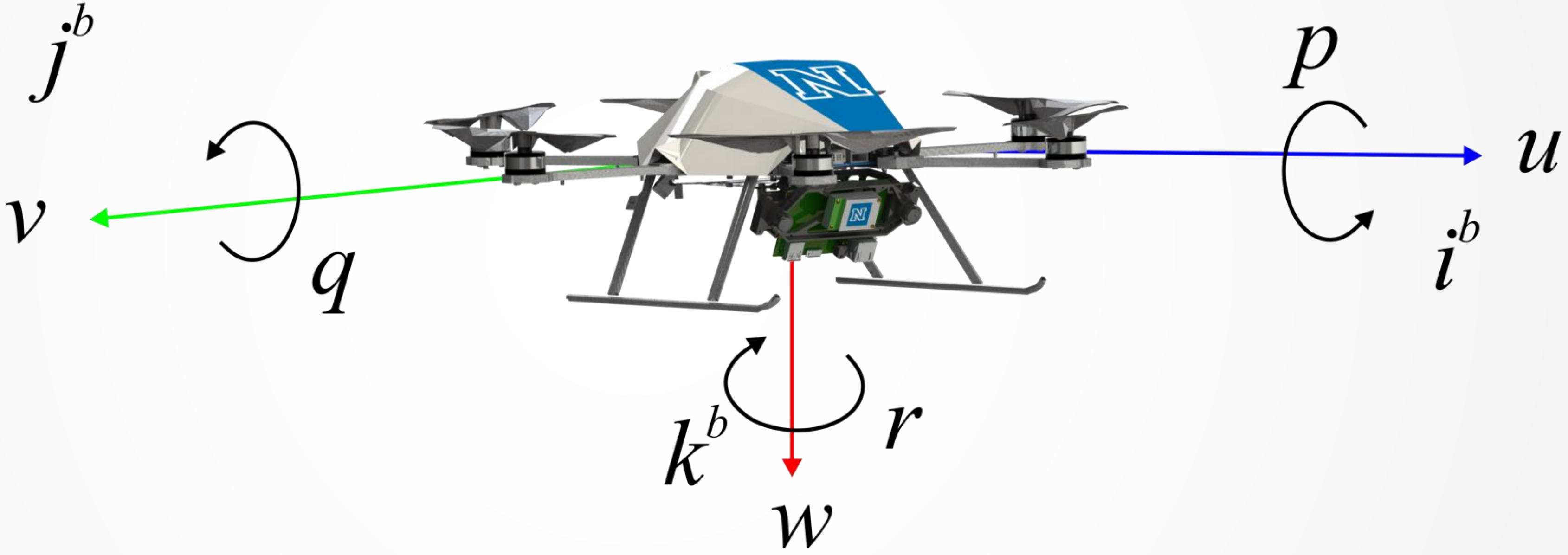
- Rotation around the k-axis

$$\mathcal{R}_0^1 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- Orthonormal matrix properties
  - $(\mathcal{R}_a^b)^{-}1 = (\mathcal{R}_a^b)^T = \mathcal{R}_b^a$
  - $\mathcal{R}_b^c \mathcal{R} a^b = \mathcal{R}_a^c$
  - $\det(\mathcal{R}_a^b) = 1$

Further Application to Robot Kinematics



- *[p,q,r]* : body angular rates
- *[u,v,w]* : body linear velocities

Relate Translational Velocity-Position

- Let *[u,v,w]* represent the body linear velocities

$$\frac{d}{dt}\begin{bmatrix} p_n \\ p_e \\ p_d \end{bmatrix} = \mathcal{R}_b^v \begin{bmatrix} u \\ v \\ w \end{bmatrix} = (\mathcal{R}_v^b)^T \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

- Which gives:

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\phi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_{c\psi} \\ -s_\theta & s_\phi c_\theta & c_{c\theta} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Body Rates – Euler Rates

- Let *[p,q,r]* denote the body angular rates

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathcal{R}_{v_2}^{b}(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathcal{R}_{v_2}^{b}(\phi)\mathcal{R}_{v_1}^{v_2}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} =$$

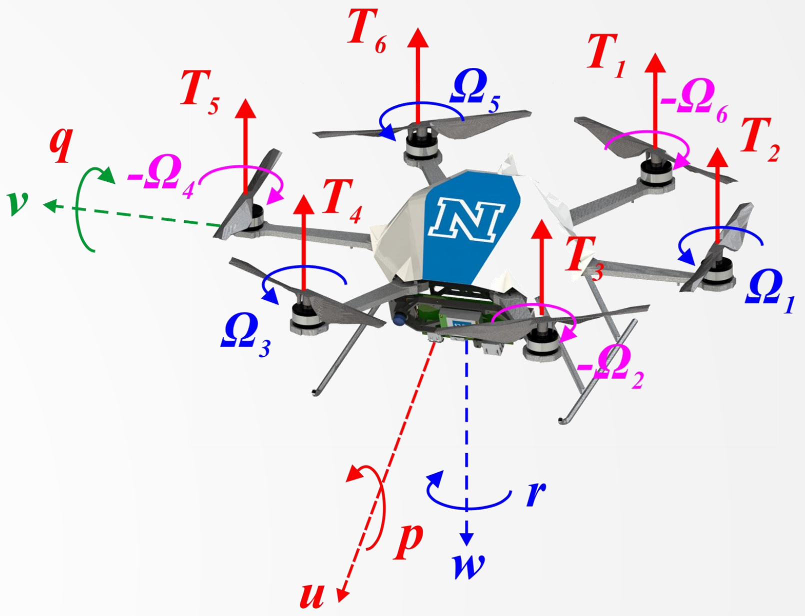
$$\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

- Inverting this expression:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Translational Dynamics



- Recall Newton's 2nd Law:

$$m\frac{d\mathbf{V}_g}{dt_i} = \mathbf{f}$$

  - f is the summary of all external forces
  - m is the mass of the robot
  - Time derivative is taken wrt the interial frame
- Using the expression:

$$\frac{d\mathbf{V}_g}{dti} = \frac{d\mathbf{V}_g}{dt_b} + \omega_{b/i} \times \mathbf{V}_g \quad \Rightarrow \quad m(\frac{d\mathbf{V}_g}{dt_b} + \omega_{b/i} \times \mathbf{V}_g) = \mathbf{f}$$

- Which expressed in the body frame:

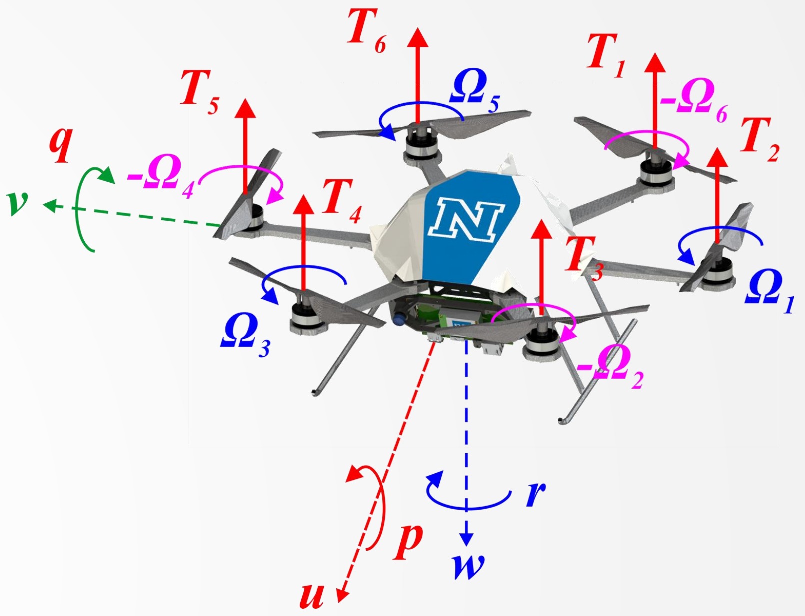
$$m(\frac{d\mathbf{V}_g^b}{dt_b} + \omega_{b/i}^b \times \mathbf{V}_g^b) = \mathbf{f}^b$$

Translational Dynamics

- Recall Newton's 2nd Law:

$$\boxed{m\frac{d\mathbf{V}_g}{dt_i} = \mathbf{f}}$$

  - f is the summary of all external forces
  - m is the mass of the robot
  - **Time derivative is taken wrt the interial frame**



- Time Derivatives in a Rotating Frame:
  - Introduce the unit vectors ***i,j,k*** representing standard unit basis vectors in the rotating frame. As they rotate they will remain normalized. If we let them rotate at a speed Ω about an axis $\boldsymbol{\Omega}$ then each unit vector $\boldsymbol{u}$ of the rotating coordinate system abides by the rule:

$$\frac{d}{dt}\mathbf{u} = \boldsymbol{\Omega} \times \mathbf{u}$$

Translational Dynamics

- Time Derivatives in a Rotating Frame:
  - Introduce the unit vectors ***i,j,k*** representing standard unit basis vectors in the rotating frame. As they rotate they will remain normalized. If we let them rotate at a speed Ω about an axis $\boldsymbol{\Omega}$ then each unit vector ***u*** of the rotating coordinate system abides by the rule:

$$\frac{d}{dt}\mathbf{u} = \boldsymbol{\Omega} \times \mathbf{u}$$

  - Then if we have a unit vector:

$$\mathbf{f}(t) = \mathbf{f}_x(t)\mathbf{i} + \mathbf{f}_y(t)\mathbf{j} + \mathbf{f}_z(t)\mathbf{k}$$

  - To examine its first derivative – we have to use the product rule of differentiation:

$$\frac{d}{dt}\mathbf{f} = \frac{df_x}{dt}\mathbf{i} + \frac{d\mathbf{i}}{dt}f_x + \frac{df_y}{dt}\mathbf{j} + \frac{d\mathbf{j}}{dt}f_y + \frac{df_z}{dt}\mathbf{k} + \frac{d\mathbf{k}}{dt}f_z \Rightarrow$$

$$\frac{d}{dt}\mathbf{f} = \left[\left(\tfrac{d}{dt}\right)_r + \boldsymbol{\Omega}\times\right]\mathbf{f}$$

Translational Dynamics

- Time Derivatives in a Rotating Frame:
  - Introduce the unit vectors ***i,j,k*** representing standard unit basis vectors in the rotating frame. As they rotate they will remain normalized. If we let them rotate at a speed Ω about an axis $\boldsymbol{\Omega}$ then each unit vector ***u*** of the rotating coordinate system abides by the rule:

$$\frac{d}{dt}\mathbf{u} = \boldsymbol{\Omega} \times \mathbf{u}$$

  - Then if we have a unit vector:

$$\mathbf{f}(t) = \mathbf{f}_x(t)\mathbf{i} + \mathbf{f}_y(t)\mathbf{j} + \mathbf{f}_z(t)\mathbf{k}$$

  - To examine its first derivative – we have to use the product rule of differentiation:

$$\frac{d}{dt}\mathbf{f} = \frac{df_x}{dt}\mathbf{i} + \frac{d\mathbf{i}}{dt}f_x + \frac{df_y}{dt}\mathbf{j} + \frac{d\mathbf{j}}{dt}f_y + \frac{df_z}{dt}\mathbf{k} + \frac{d\mathbf{k}}{dt}f_z \Rightarrow$$

$$\frac{d}{dt}\mathbf{f} = \left[\left(\frac{d}{dt}\right)_r + \boldsymbol{\Omega}\times\right]\mathbf{f}$$

Translational Dynamics

- As a result: **Relation Between Velocities in the Inertial & Rotating Frame**
  - Let $\boldsymbol{v}$ be the position of an object's position:

$$\mathbf{v} = \frac{d}{dt}\mathbf{p}$$

  - Then the relation of the velocity as expressed in the inertial frame and as expressed in the rotating frame becomes:

$$\mathbf{v}_i = \mathbf{v}_r + \boldsymbol{\Omega} \times \mathbf{p}$$

- Similarly: **Relation Between Accelerations in the Inertial & Rotating Frame**
  - Let $\boldsymbol{a}$ be the acceleration of an object's position. Then:

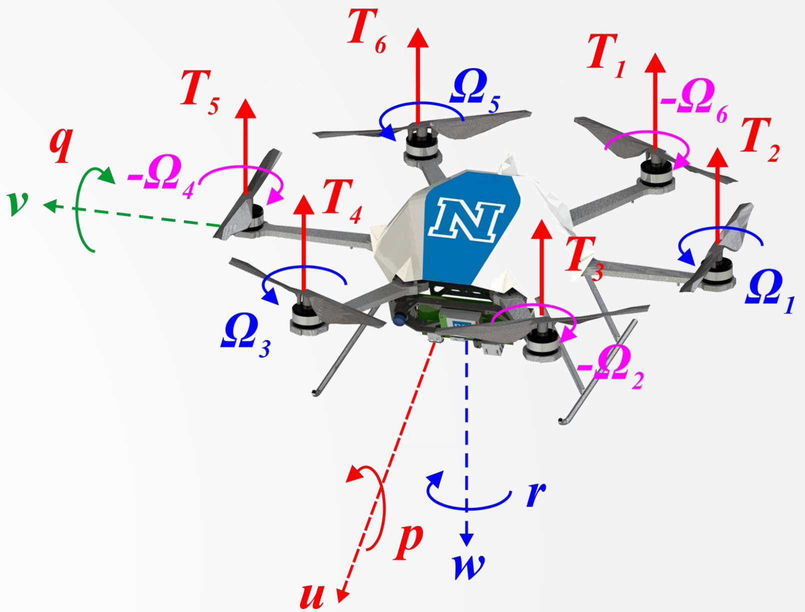
$$\mathbf{a}_i = \left(\frac{d\mathbf{p}}{dt}\right)_i = \left(\frac{d\mathbf{v}}{dt}\right)_i = \left[\left(\frac{d}{dt}_r + \boldsymbol{\Omega}\times\right)\right]\left[\left(\frac{d\mathbf{p}}{dt}\right)_r + \boldsymbol{\Omega} \times \mathbf{p}\right]$$

  - Carrying out the differentiations:

$$\mathbf{a}_r = \mathbf{a}_i - 2\boldsymbol{\Omega} \times \mathbf{v}_r - \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{p}) - \frac{d\boldsymbol{\Omega}}{dt} \times \mathbf{p}$$

- *Subscripts i, r represent the inertial frame and the rotating frame respectively.*

Translational Dynamics



- Recall Newton's 2nd Law:

$$m\frac{d\mathbf{V}_g}{dt_i} = \mathbf{f}$$

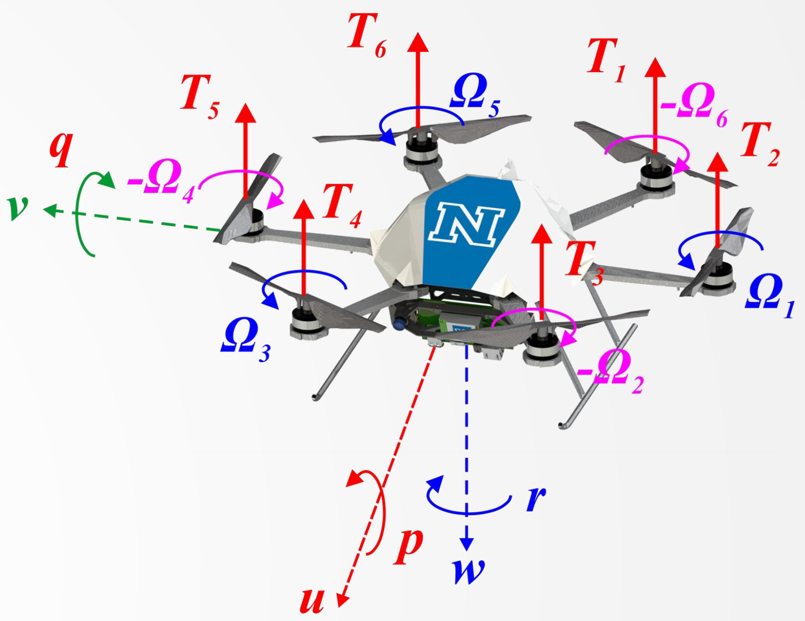
  - f is the summary of all external forces
  - m is the mass of the robot
  - Time derivative is taken wrt the interial frame
- Using the expression:

$$\frac{d\mathbf{V}_g}{dti} = \frac{d\mathbf{V}_g}{dt_b} + \omega_{b/i} \times \mathbf{V}_g \quad \Rightarrow \quad m(\frac{d\mathbf{V}_g}{dt_b} + \omega_{b/i} \times \mathbf{V}_g) = \mathbf{f}$$

- Which expressed in the body frame:

$$m(\frac{d\mathbf{V}_g^b}{dt_b} + \omega_{b/i}^b \times \mathbf{V}_g^b) = \mathbf{f}^b$$

Translational Dynamics



$$m\left(\frac{d\mathbf{V}_g^b}{dt_b} + \omega_{b/i}^b \times \mathbf{V}_g^b\right) = \mathbf{f}^b$$

- Where

$$\mathbf{V}_g^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \omega_{b/i}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \mathbf{f}^b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

- Therefore:

$$\boxed{\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m}\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}}$$

Rotational Dynamics

- Recall Newton's 2nd Law:

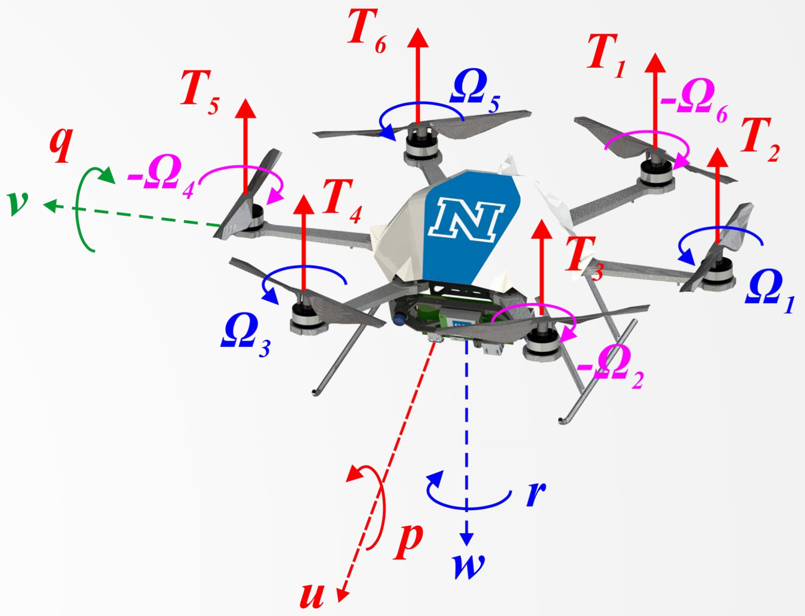
$$\frac{d\mathbf{h}}{dt_i} = \mathbf{m}$$

  - ***h*** is the angular momentum vector
  - ***m*** is the summary of all external moments
  - Time derivative is taken wrt the interial frame
- Therefore:

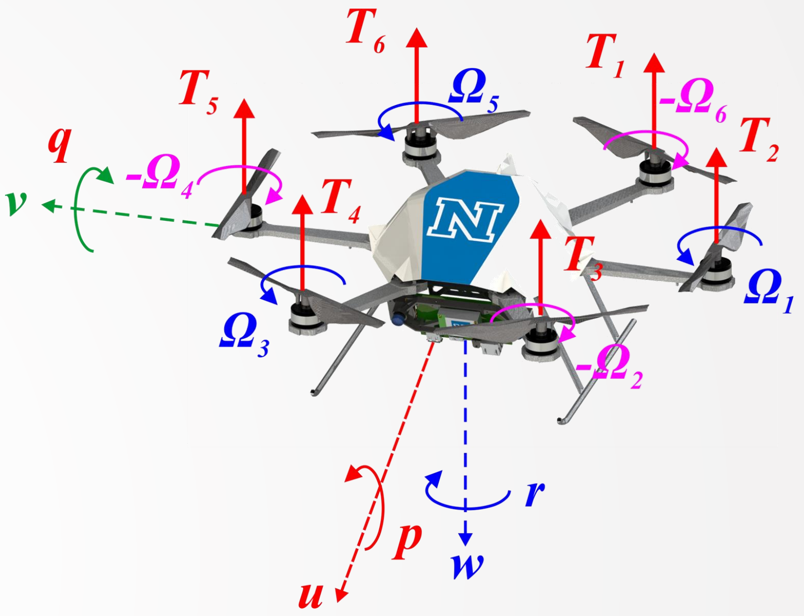
$$\frac{d\mathbf{h}}{dt_i} = \frac{d\mathbf{h}}{dt_b} + \omega_{b/i} \times \mathbf{h} = \mathbf{m}$$

- Which expressed in the body frame:

$$\frac{d\mathbf{h}^b}{dt_b} + \omega_{b/i}^b \times \mathbf{h}^b = \mathbf{m}^b$$



Rotational Dynamics



- For a rigid body, the **angular momentum** is defined as the product of the inertia matrix and the angular velocity vector:

$$\mathbf{h}^b = \mathbf{J}\omega_{b/i}^b$$

- where

$$\mathbf{J} = \begin{bmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{bmatrix}$$

- But as the multirotor MAV is symmetric:

$$\mathbf{J} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}$$

Rotational Dynamics

- Replacing in:

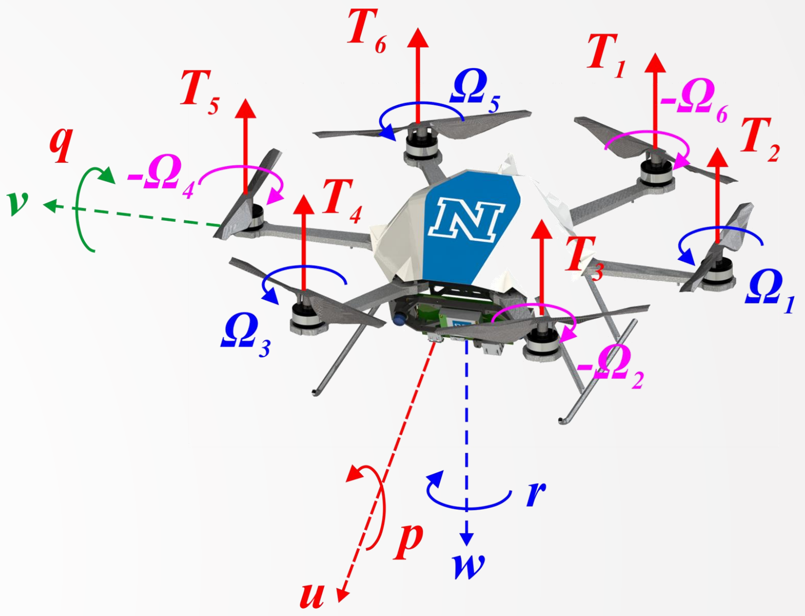
$$\frac{d\mathbf{h}^b}{dt_b} + \omega^b_{b/i} \times \mathbf{h}^b = \mathbf{m}^b$$

- Gives:

$$\mathbf{J}\frac{d\omega^b_{b/i}}{dt_b} + \omega^b_{b/i} \times (\mathbf{J}\omega^b_{b/i}) = \mathbf{m}^b \Rightarrow$$
$$\dot{\omega}^b_{b/i} = \mathbf{J}^{-1}[-\omega^b_{b/i} \times (\mathbf{J}\omega^b_{b/i}) + \mathbf{m}^b]$$

- where

$$\dot{\omega}^b_{b/i} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}$$



Rotational Dynamics

- By setting the moments vector:

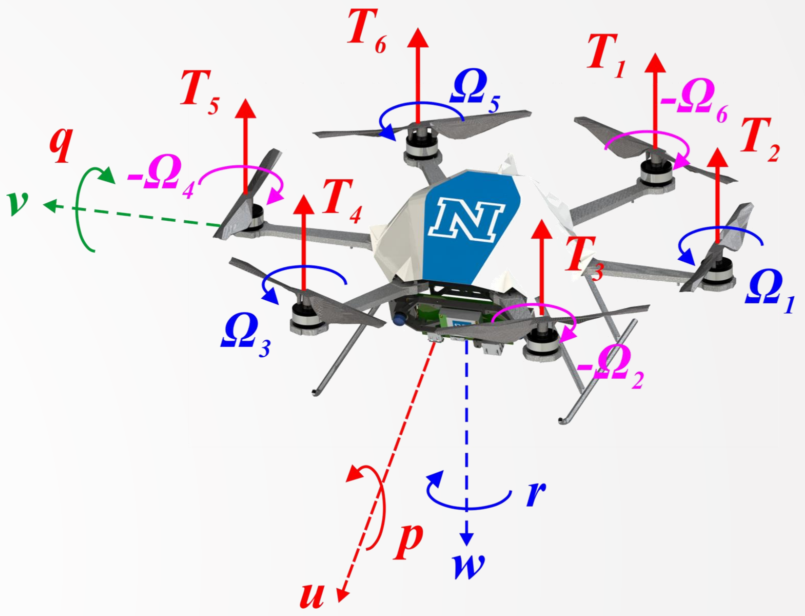
$$\mathbf{m}^b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}$$

- Then for the symmetric MAV, equation:

$$\dot{\omega}^b_{b/i} = \mathbf{J}^{-1}[-\omega^b_{b/i} \times (\mathbf{J}\omega^b_{b/i}) + \mathbf{m}^b]$$

- Becomes:

$$\boxed{\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x} M_x \\ \frac{1}{J_y} M_y \\ \frac{1}{J_z} M_z \end{bmatrix}}$$



MAV Dynamics

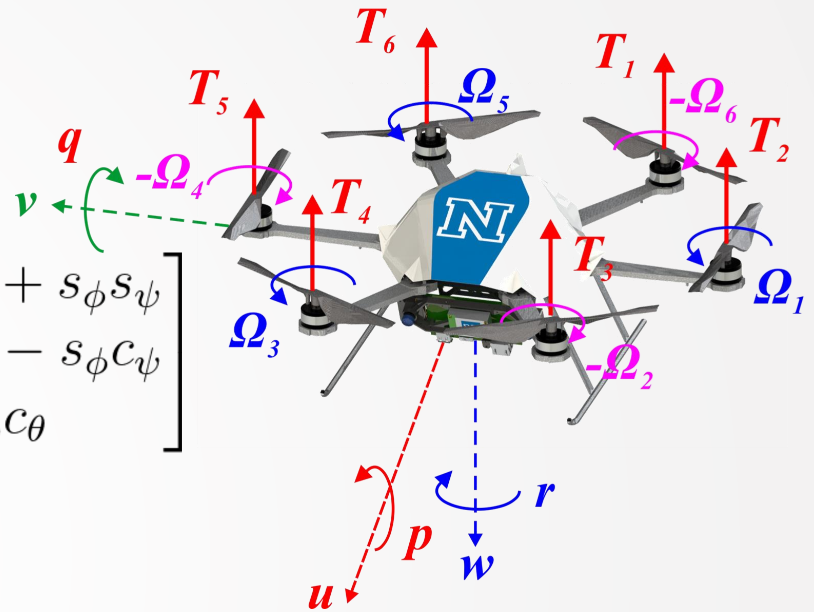
- To append the forces and moments we need to combine their formulation with

$$\begin{bmatrix}\dot{p}_n\\ \dot{p}_e\\ \dot{p}_d\end{bmatrix} = \mathcal{R}_b^v \begin{bmatrix}u\\ v\\ w\end{bmatrix}, \quad \mathcal{R}_b^v = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m}\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x} M_x \\ \frac{1}{J_y} M_y \\ \frac{1}{J_z} M_z \end{bmatrix}$$

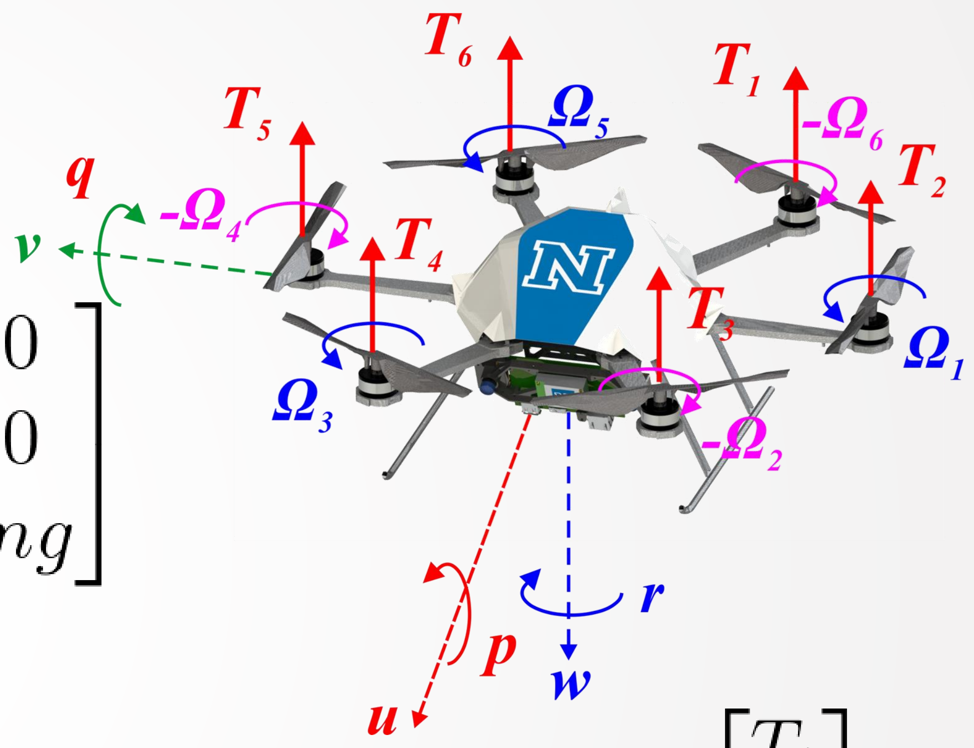


- ***Next step: append the MAV forces and moments***

MAV Dynamics

- MAV forces in the body frame:

$$\mathbf{f}_b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{6} T_i \end{bmatrix} - \mathcal{R}_v^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$



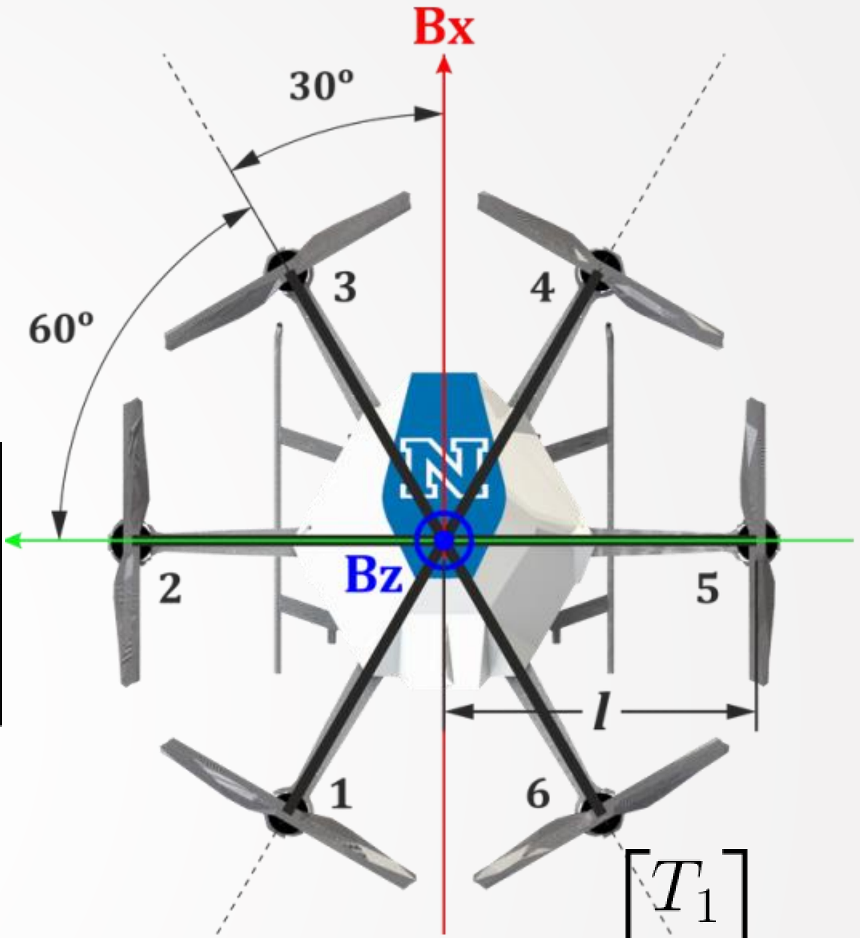
- Moments in the body frame:

$$\mathbf{m}_b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} lc_{60} & l & lc_{60} & -lc_{60} & -l & -lc_{60} \\ -ls_{60} & 0 & ls_{60} & ls_{60} & 0 & -ls_{60} \\ -k_m & k_m & -k_m & k_m & -k_m & k_m \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

MAV Dynamics

- MAV forces in the body frame:

$$\mathbf{f}_b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{6} T_i \end{bmatrix} - \mathcal{R}_v^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

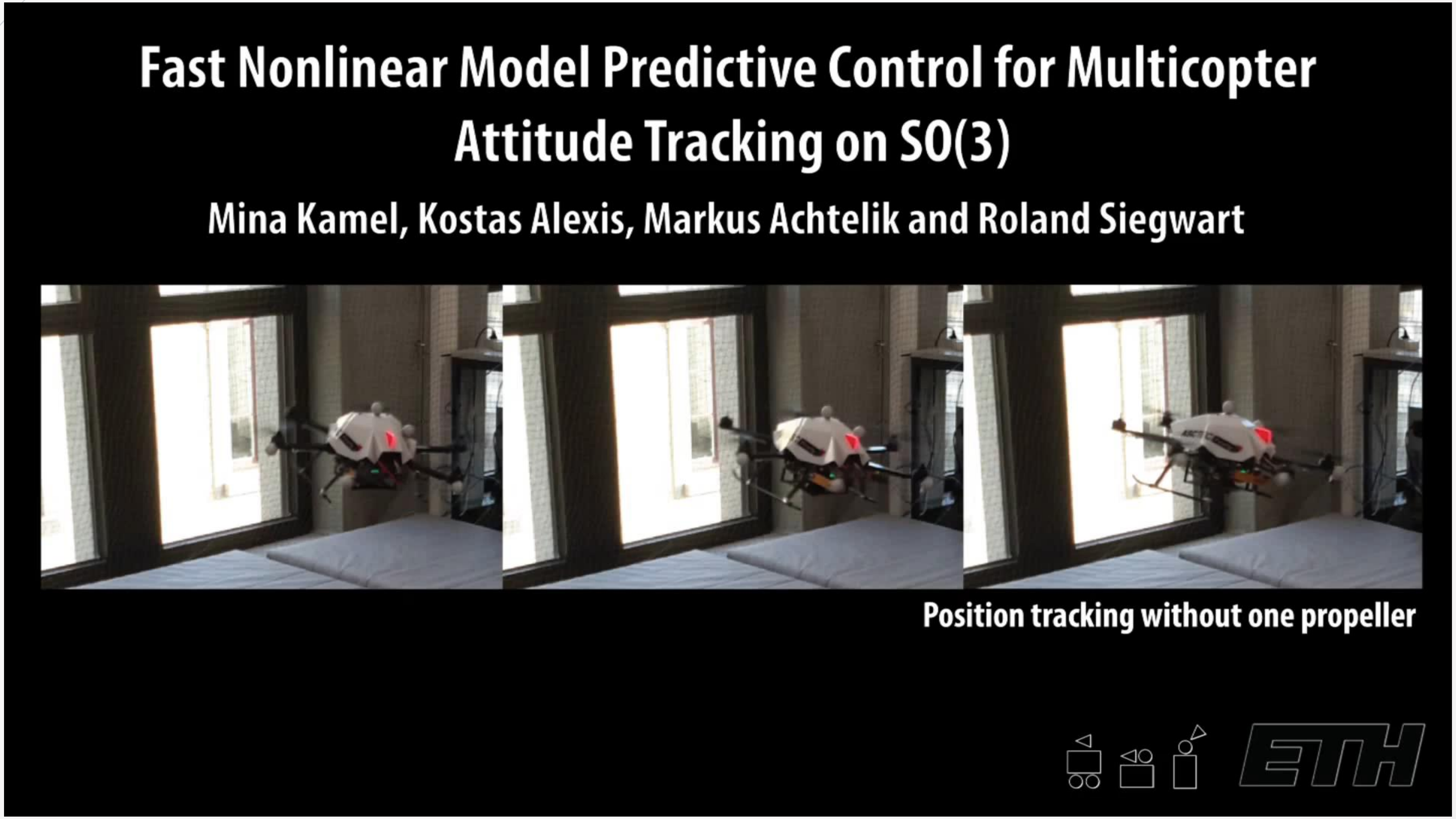


- Moments in the body frame:

$$\mathbf{m}_b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} lc_{60} & l & lc_{60} & -lc_{60} & -l & -lc_{60} \\ -ls_{60} & 0 & ls_{60} & ls_{60} & 0 & -ls_{60} \\ -k_m & k_m & -k_m & k_m & -k_m & k_m \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

Relevant Research Study

- Can we exploit the control allocation of a hexacopter MAV to provide tolerance against a motor failure?



Mina Kamel, Kostas Alexis, Markus Wilhelm Achtelik, Roland Siegwart, ***"Fast Nonlinear Model Predictive Control for Multicopter Attitude Tracking on SO(3)"***, Multiconference on Systems and Control (MSC), 2015, Novotel Sydney Manly Pacific, Sydney Australia. 21-23 September, 2015

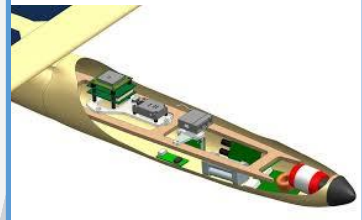
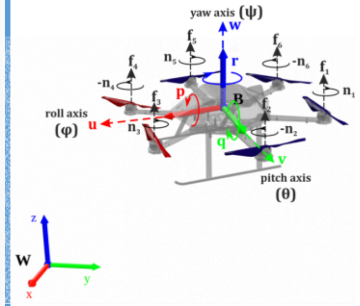
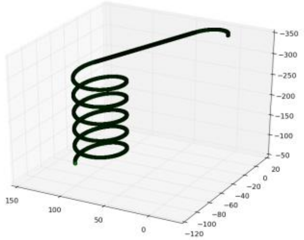
Find out more

- http://www.autonomousrobotslab.com/frame-rotations-and-representations.html
- http://page.math.tu-berlin.de/~plaue/plaue_intro_quats.pdf
- http://mathworld.wolfram.com/RotationMatrix.html
- http://mathworld.wolfram.com/EulerAngles.html
- http://blog.wolframalpha.com/2011/08/25/quaternion-properties-and-interactive-rotations-with-wolframalpha/
- http://www.mathworks.com/discovery/rotation-matrix.html
- http://www.mathworks.com/discovery/quaternion.html?refresh=true
- http://www.cprogramming.com/tutorial/3d/rotationMatrices.html
- http://www.cprogramming.com/tutorial/3d/quaternions.html
- http://www.kostasalexis.com/multirotor-dynamics.html
- S. Leutenegger,C. Huerzeler, A.K. Stowers, K. Alexis, M. Achtelik, D. Lentink, P. Oh, and R. Siegwart. **"Flying Robots",** *Handbook of Robotics*.
- http://www.kostasalexis.com/simulations-with-simpy.html
- MATLAB Demo: http://www.mathworks.com/help/aeroblks/examples/quadcopter-project.html?refresh=true
- **Quick Help with Linear Algebra?** https://www.khanacademy.org/math/linear-algebra
- **Quick Help with Differential Equations?** https://www.khanacademy.org/math/differential-equations
- **Always check:** http://www.kostasalexis.com/literature-and-links.html

Course Projects

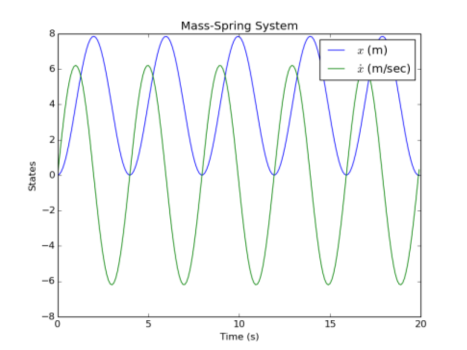
Those arranged: an extra e-mail to finalize the topic of the project

Those that have not arranged yet: hurry up ☺

System Design	Modeling	Estimation	Flight Control	Path Planning	Perception
					

Relevant Web link: http://www.autonomousrobotslab.com/student-projects.html

Simulation Tools



http://www.autonomousrobotslab.com/rotors-simulator2.html

Thank you!

Please ask your question!