

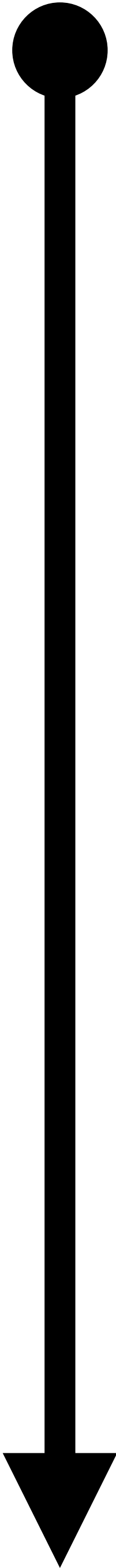
Estimating Airplane Aerodynamics from Data

MSI Guest Lecture

Jakob Harzer, 21.01.2026



Jakob Harzer



HOCH
SCHULE
OFFEN
BURG



Jakob Harzer



2014 - 2018: Electrical Engineering



Jakob Harzer



2014 - 2018: Electrical Engineering

universität freiburg



Jakob Harzer



2014 - 2018: Electrical Engineering

universität freiburg

2018 - 2022: Embedded Systems Engineering
Since 2022: PhD in with Prof. Diehl



Jakob Harzer

Airborne Wind Energy



Image from Kitepower

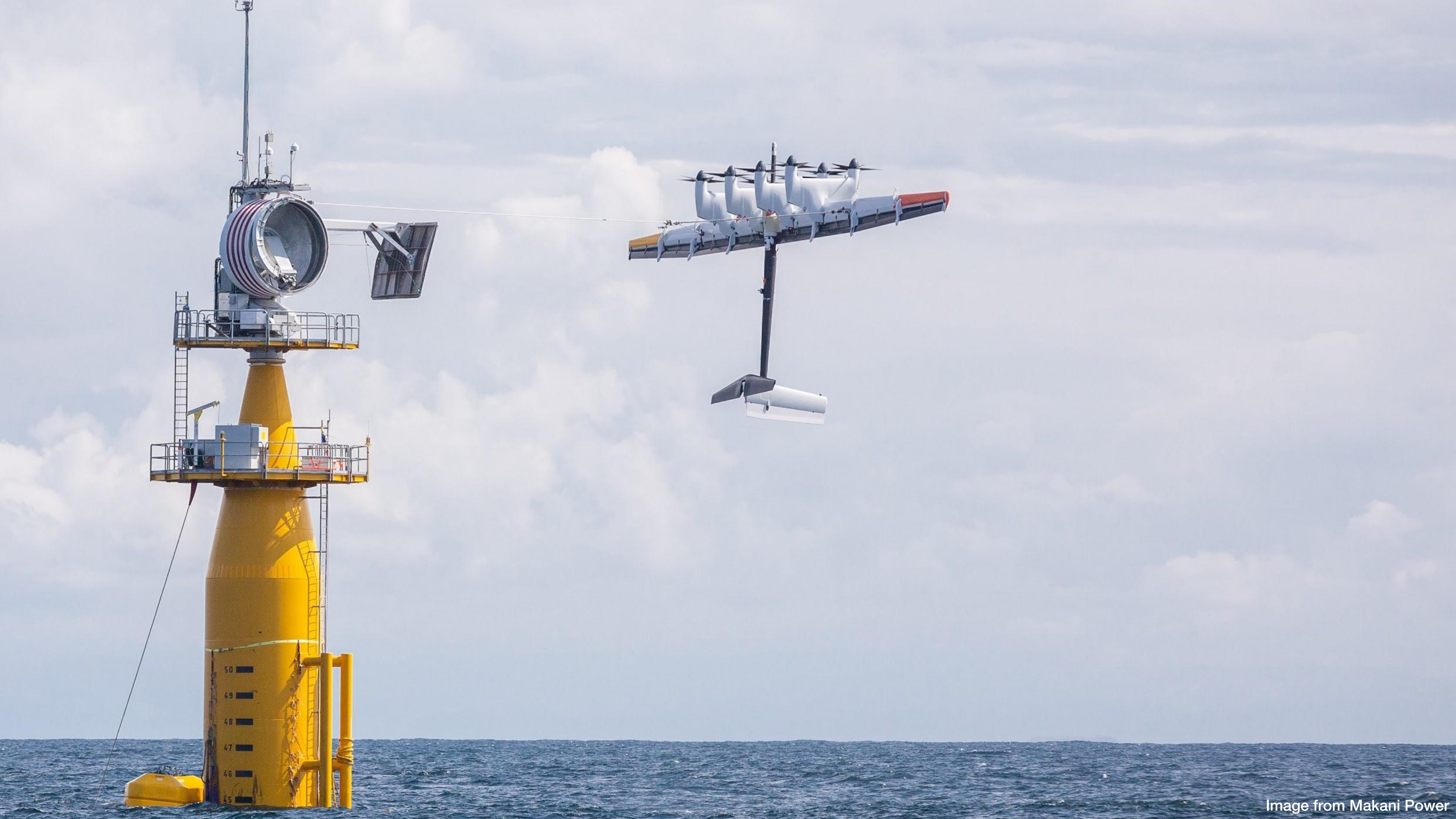
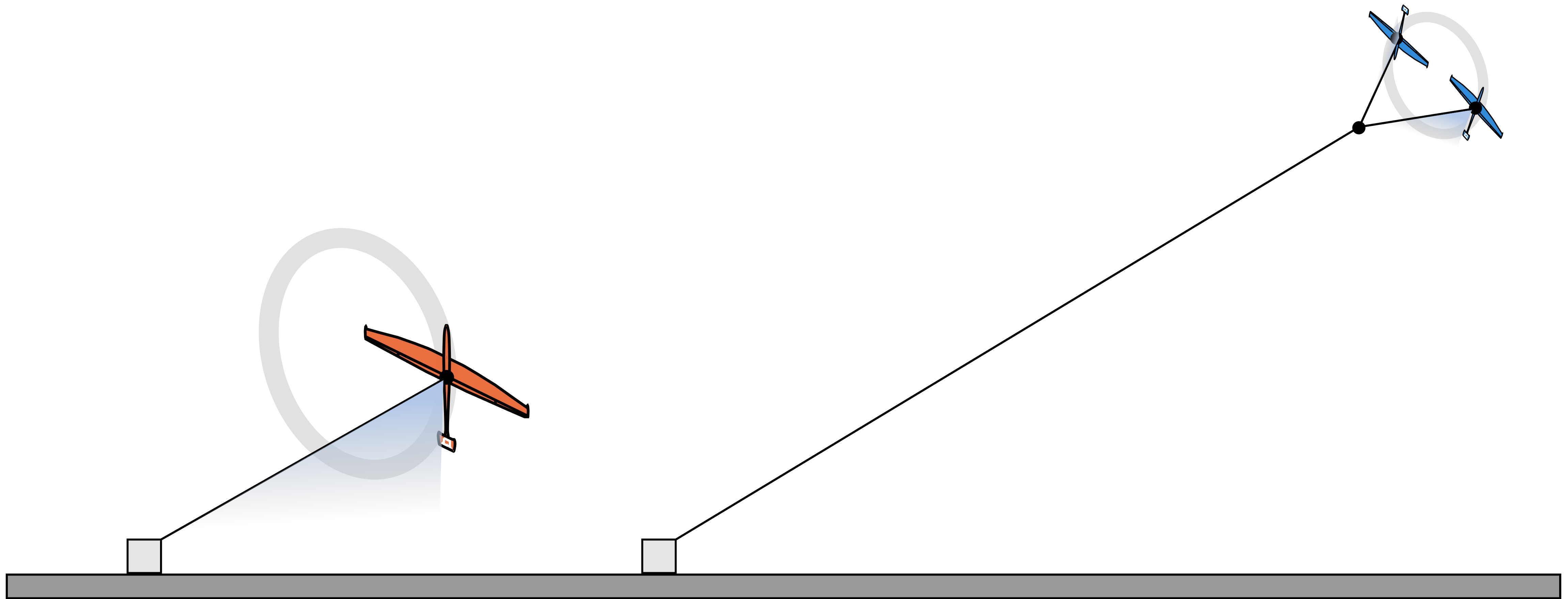
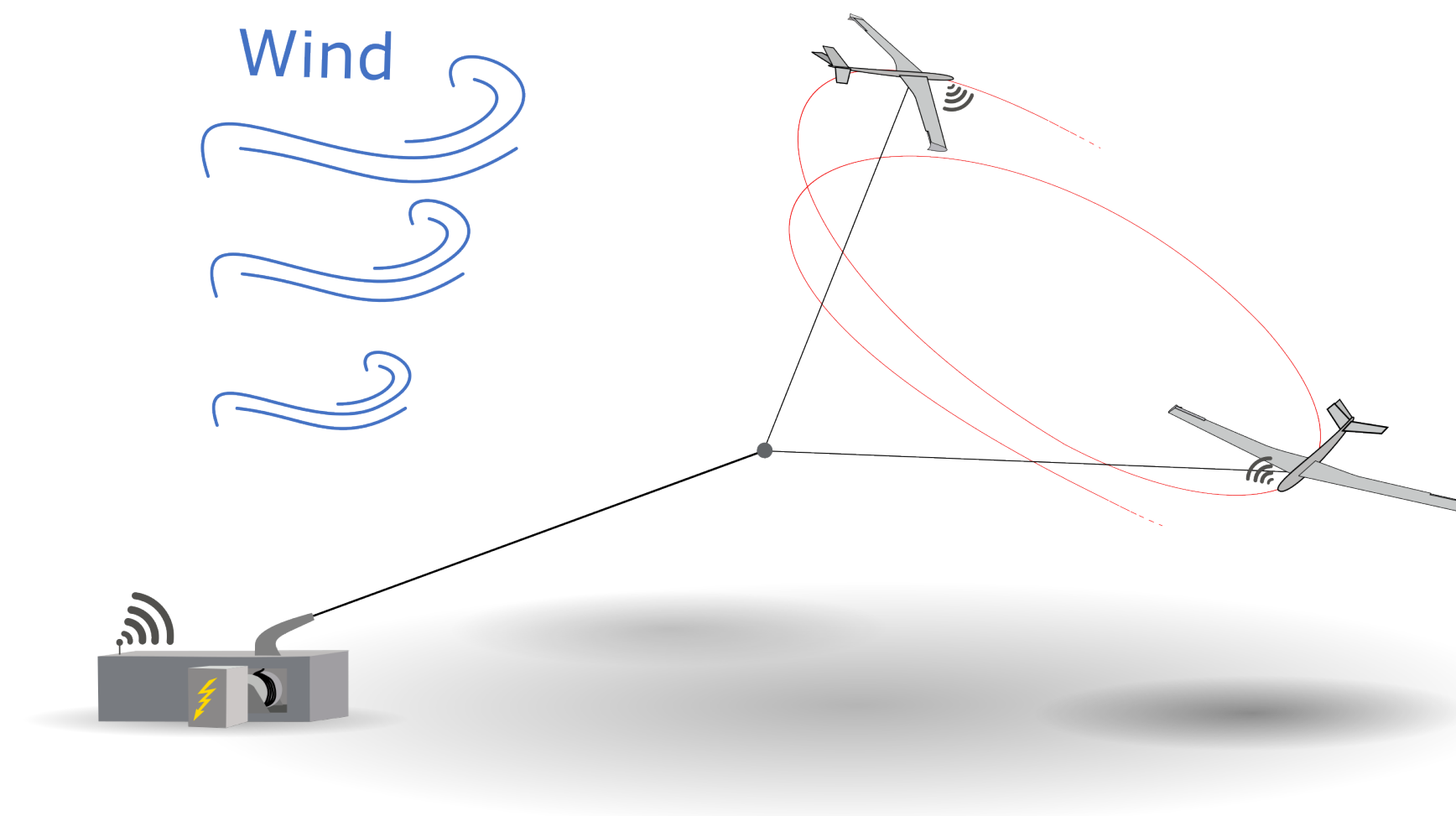


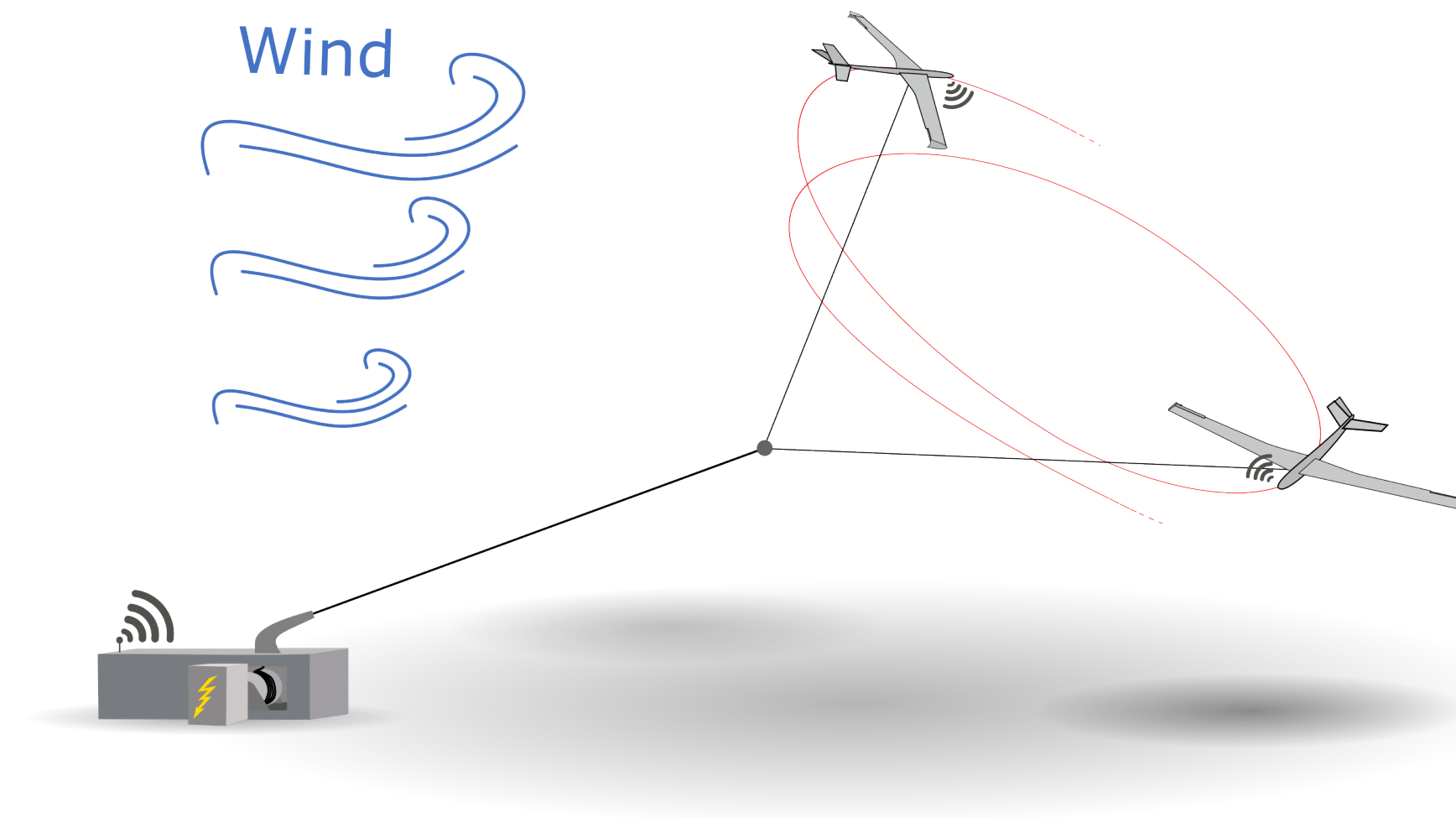
Image from Makani Power

Single vs Dual-Kite Systems



The MAWERO Project





universität freiburg

Systems Control and Optimization Laboratory



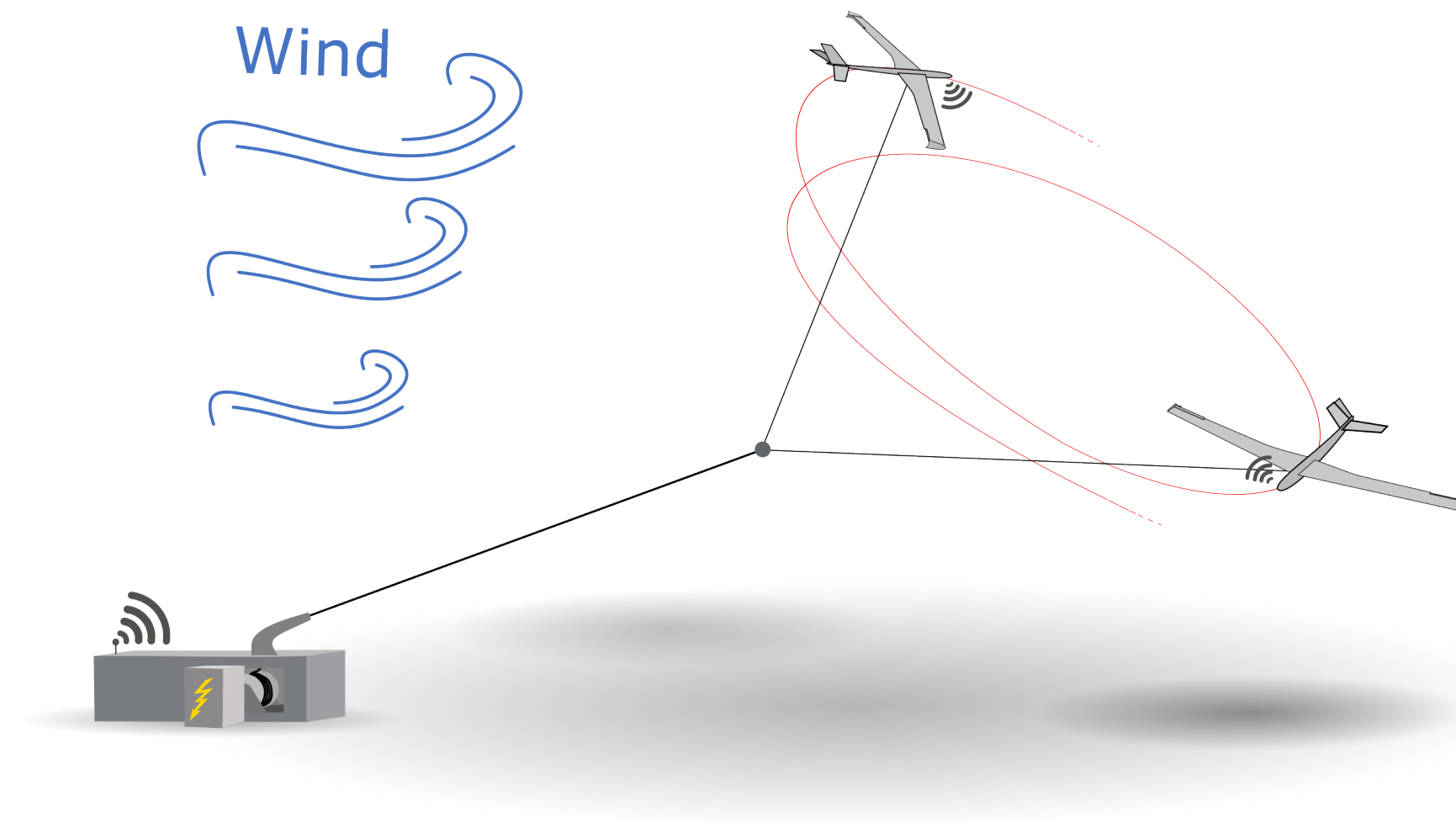
Prof. Moritz Diehl



Jakob Harzer



Jonas Keplinger



RWTHAACHEN
UNIVERSITY

Chair and Institute of Flight System Dynamics

universität freiburg

Systems Control and Optimization Laboratory



Prof.
Dieter Moormann



Kim Polkläser



Sven Jackisch



Johannes Schuler



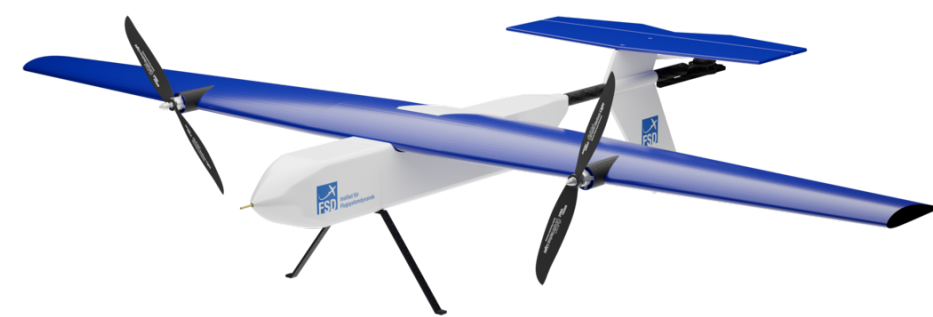
Prof. Moritz Diehl



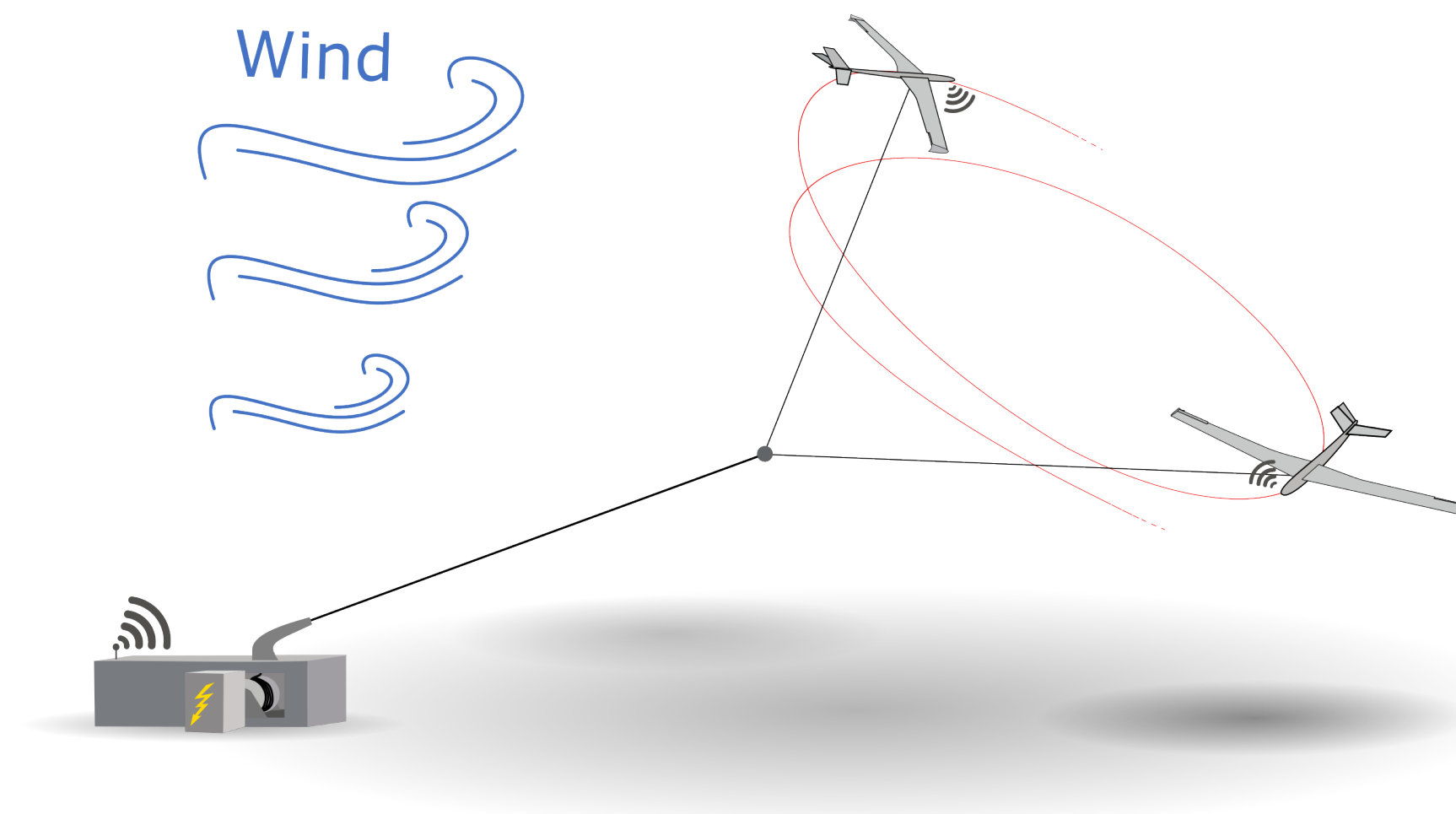
Jakob Harzer



Jonas Keplinger



Maverix Flight System



RWTHAACHEN
UNIVERSITY

Chair and Institute of Flight System Dynamics

universität freiburg

Systems Control and Optimization Laboratory



Prof.
Dieter Moormann



Kim Polkläser



Sven Jackisch



Johannes Schuler



Prof. Moritz Diehl



Jakob Harzer

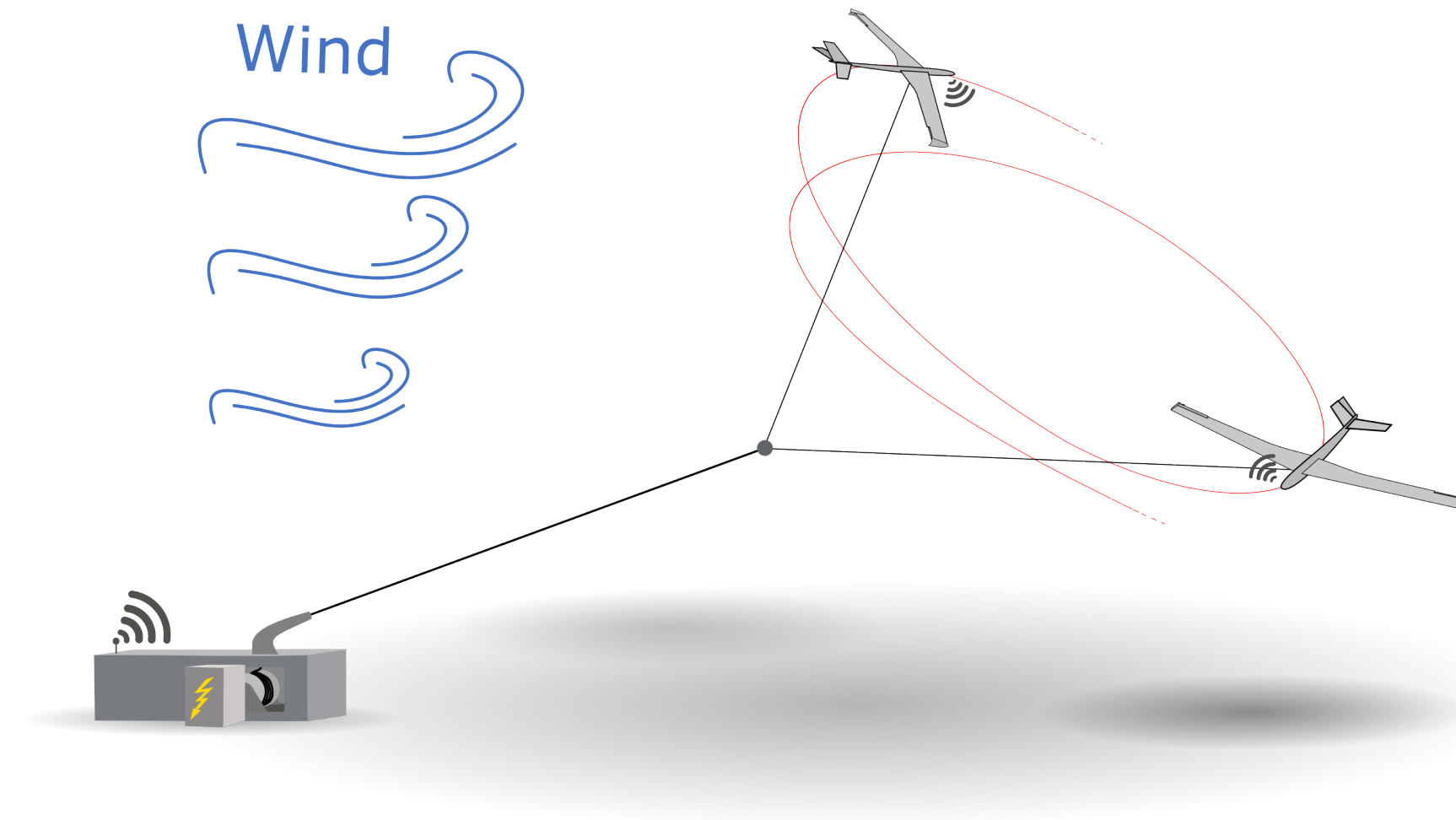


Jonas Keplinger



Maverix Flight System

$$m = 2.5 \text{ kg}, \quad A = 0.27 \text{ m}^2$$



RWTHAACHEN
UNIVERSITY

Chair and Institute of Flight System Dynamics

universität freiburg

Systems Control and Optimization Laboratory



Prof.
Dieter Moormann



Kim Polkläser



Sven Jackisch



Johannes Schuler



Prof. Moritz Diehl

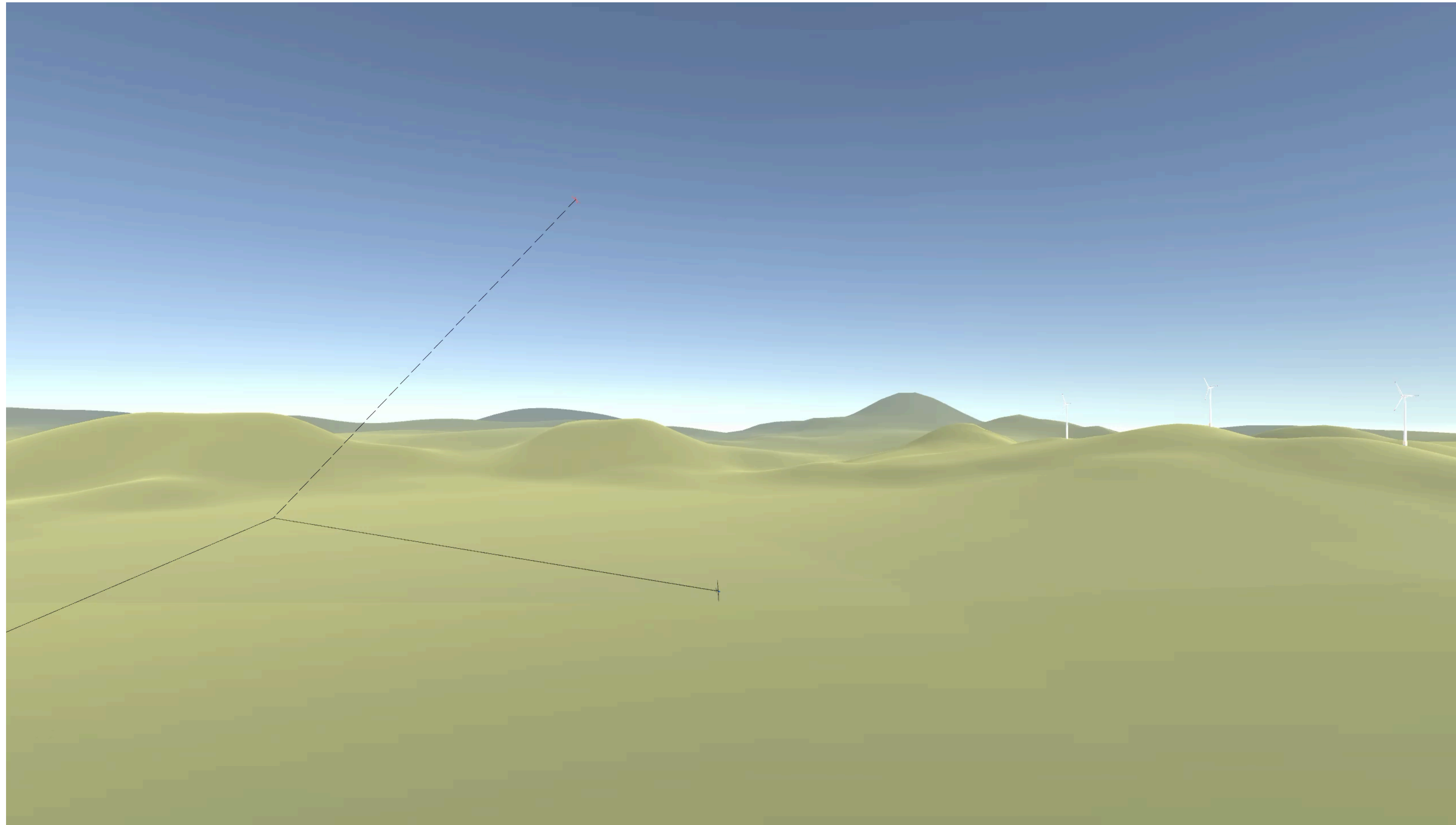


Jakob Harzer

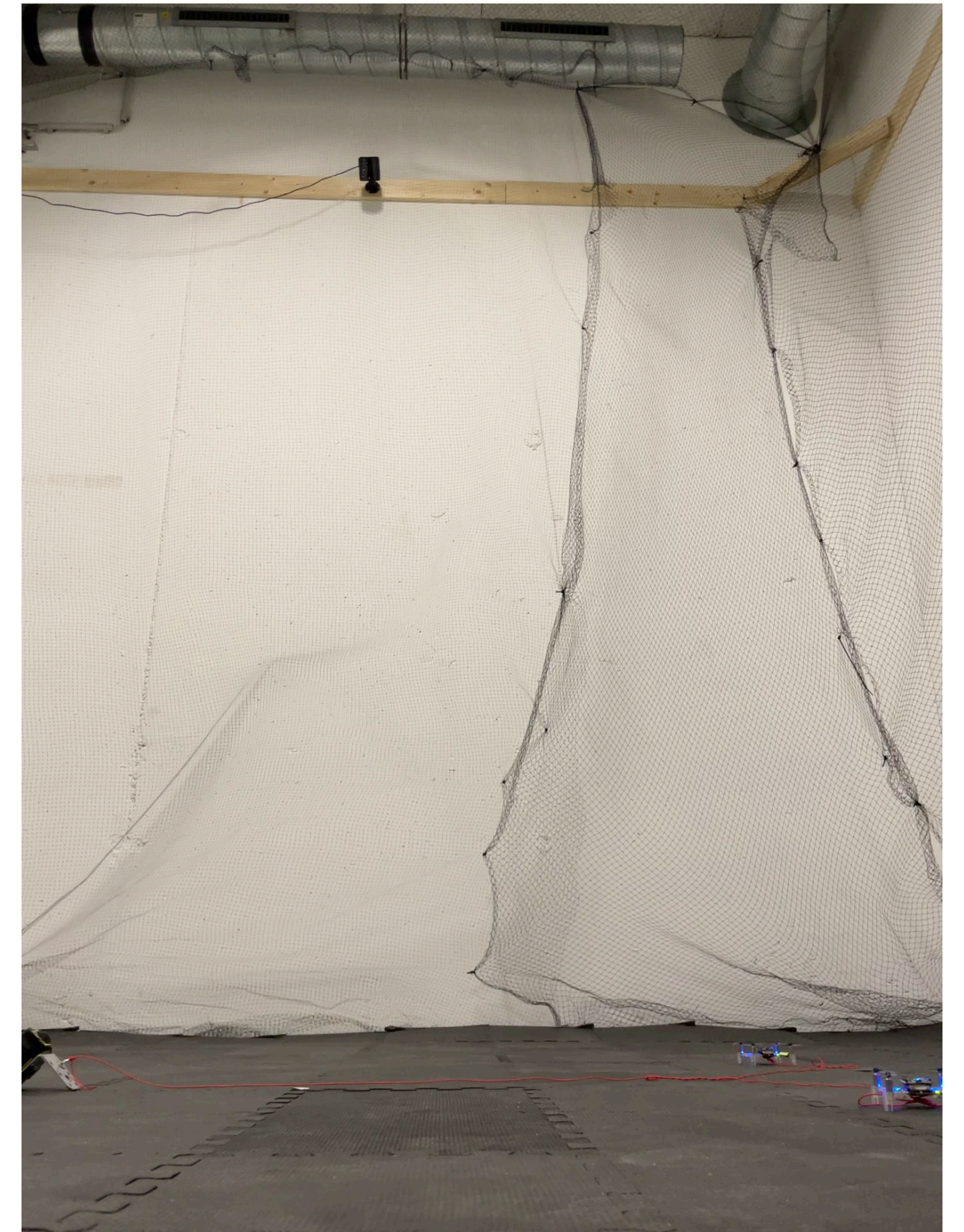


Jonas Keplinger

UFR: Trajectory Optimization & Control

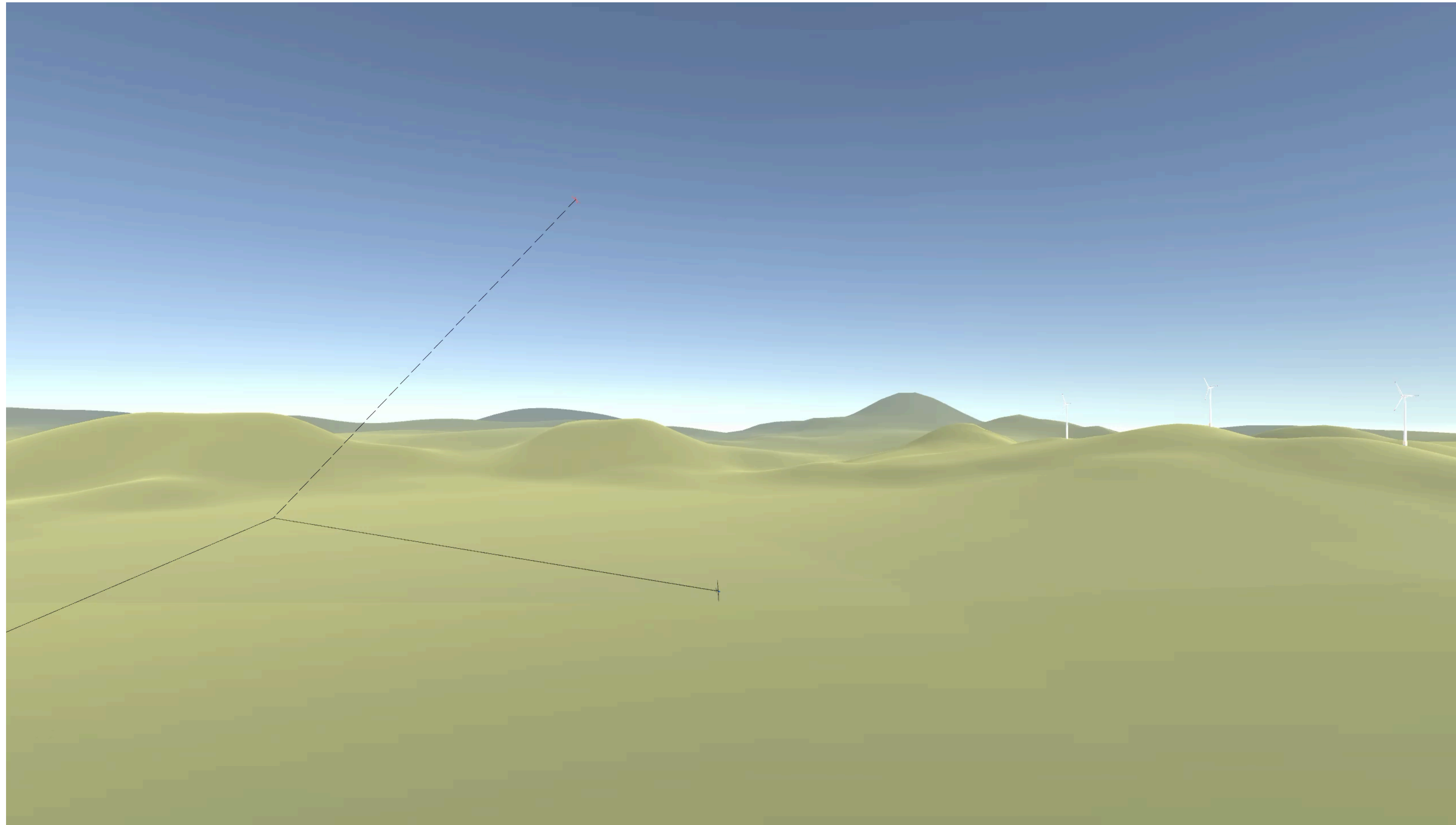


Power-Optimal Trajectory



Control Experiments in the Drone Lab

UFR: Trajectory Optimization & Control

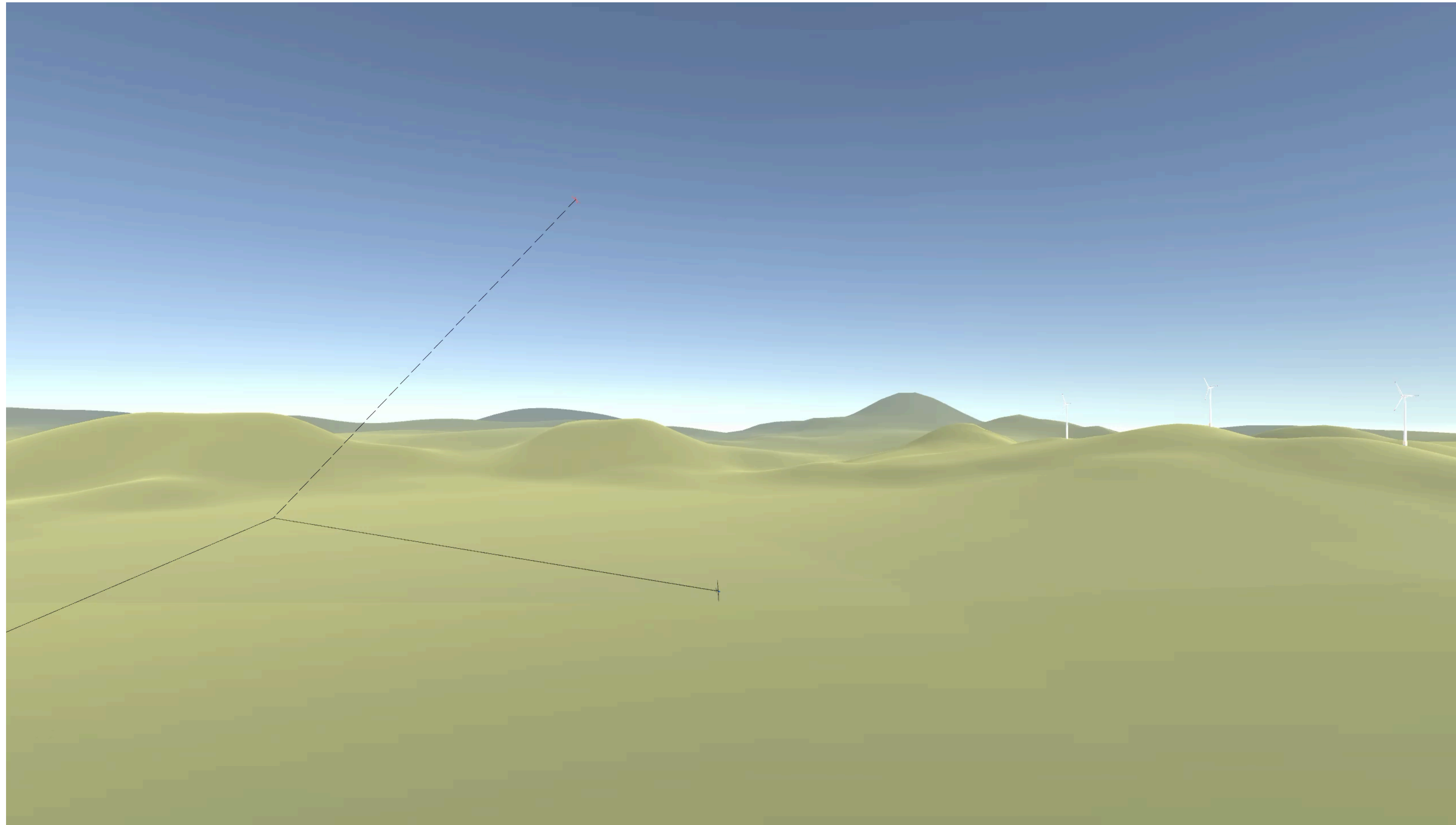


Power-Optimal Trajectory



Control Experiments in the Drone Lab

UFR: Trajectory Optimization & Control

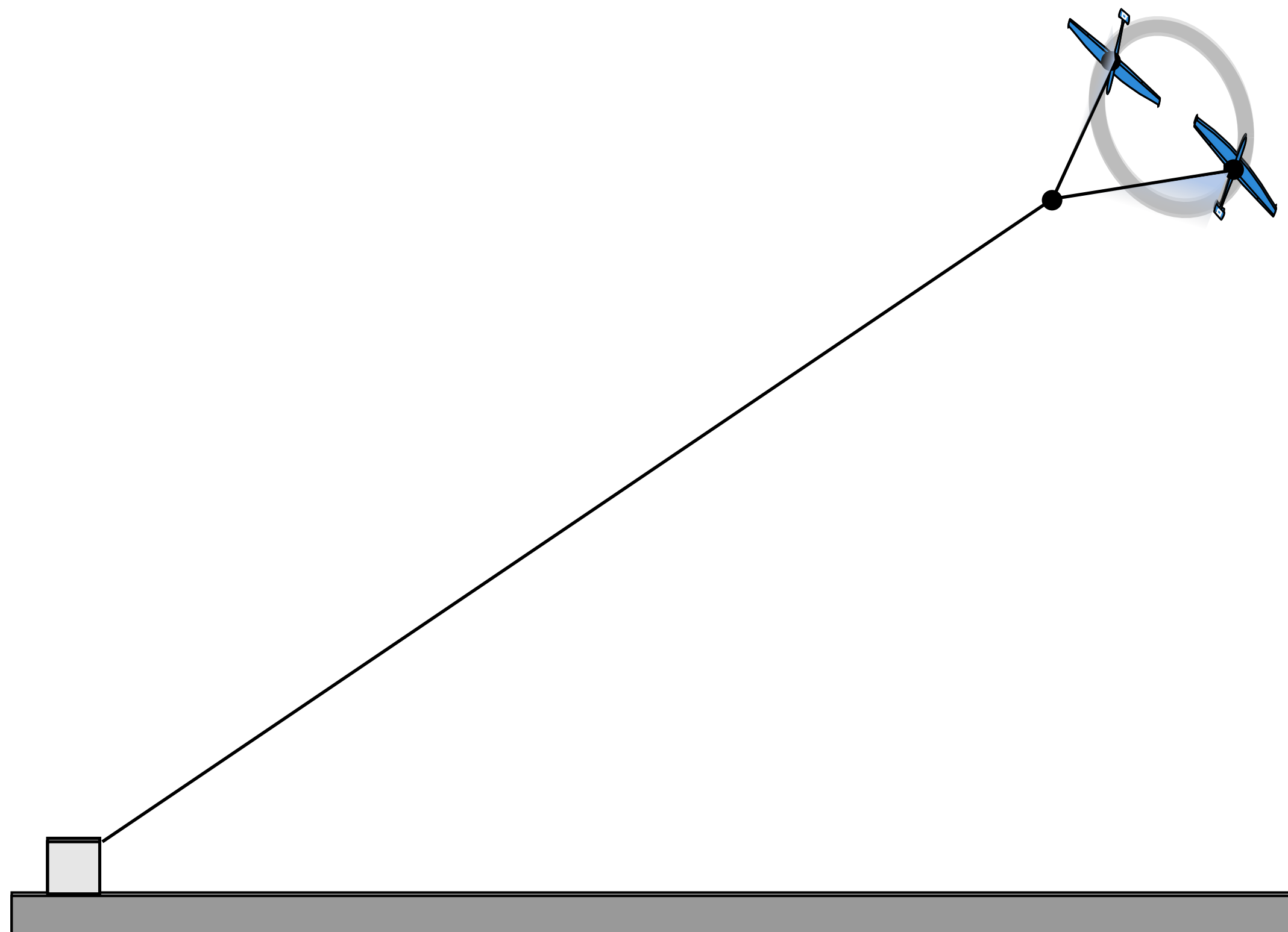


Power-Optimal Trajectory



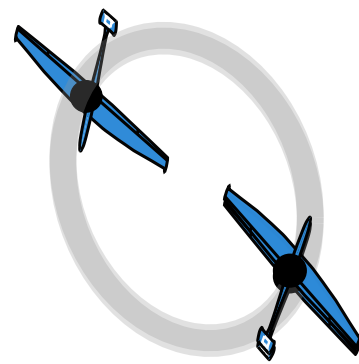
Control Experiments in the Drone Lab

For now: The Most Simple Case



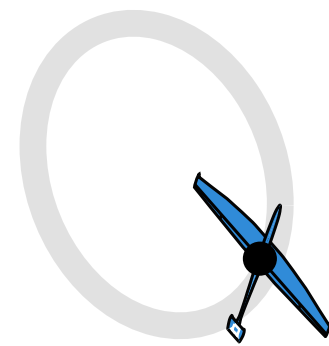
For now: The Most Simple Case

- No tether



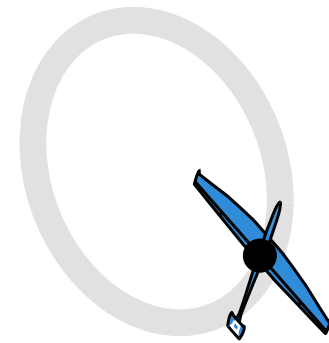
For now: The Most Simple Case

- No tether
- No wind



For now: The Most Simple Case

- No tether
- No wind



**We need
a model!**



A Dynamic Model for an Airplane

A Dynamic

3D Rigid Body Physics

airplane

State-Space Model



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

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

States

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

States

Position [m] $p \in \mathbb{R}^3$

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

Orientation $R \in \mathbb{R}^9$

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

Orientation $R \in \mathbb{R}^9$

Rot. Vel. [rad/s] $\omega \in \mathbb{R}^3$

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

$$u = \begin{bmatrix} \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^5$$

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

Orientation $R \in \mathbb{R}^9$

Rot. Vel. [rad/s] $\omega \in \mathbb{R}^3$

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

$$u = \begin{bmatrix} \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^5$$

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

Orientation $R \in \mathbb{R}^9$

Rot. Vel. [rad/s] $\omega \in \mathbb{R}^3$

Controls

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

$$u = \begin{bmatrix} \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^5$$

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

Orientation $R \in \mathbb{R}^9$

Rot. Vel. [rad/s] $\omega \in \mathbb{R}^3$

Controls

Propellor Thrusts [N] $\tau_L, \tau_R \in \mathbb{R}$

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

$$u = \begin{bmatrix} \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^5$$

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

Orientation $R \in \mathbb{R}^9$

Rot. Vel. [rad/s] $\omega \in \mathbb{R}^3$

Controls

Propellor Thrusts [N] $\tau_L, \tau_R \in \mathbb{R}$

Ailerons Position [rad] $\delta_L, \delta_R \in \mathbb{R}$

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

$$u = \begin{bmatrix} \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^5$$

States

Position [m]	$p \in \mathbb{R}^3$
Velocity [m/s]	$v \in \mathbb{R}^3$
Orientation	$R \in \mathbb{R}^9$
Rot. Vel. [rad/s]	$\omega \in \mathbb{R}^3$

Controls

Propellor Thrusts [N]	$\tau_L, \tau_R \in \mathbb{R}$
Ailerons Position [rad]	$\delta_L, \delta_R \in \mathbb{R}$
Elevator Position [rad]	$\gamma \in \mathbb{R}$

State-Space Model



$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \phantom{\dot{p}} \\ \phantom{\dot{v}} \\ \phantom{\dot{R}} \\ \phantom{\dot{\omega}} \end{bmatrix}$$

State-Space Model



$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ \vdots \end{bmatrix}$$

State-Space Model



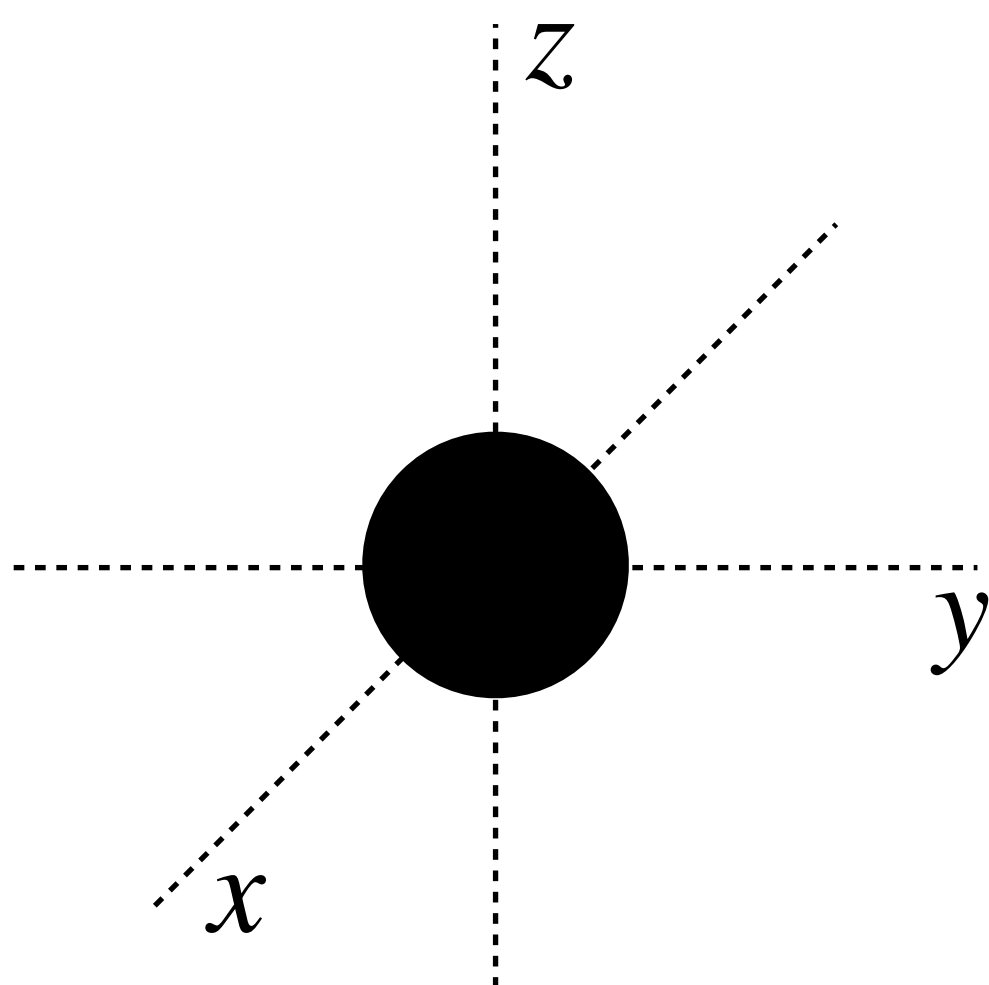
$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \end{bmatrix}$$

State-Space Model

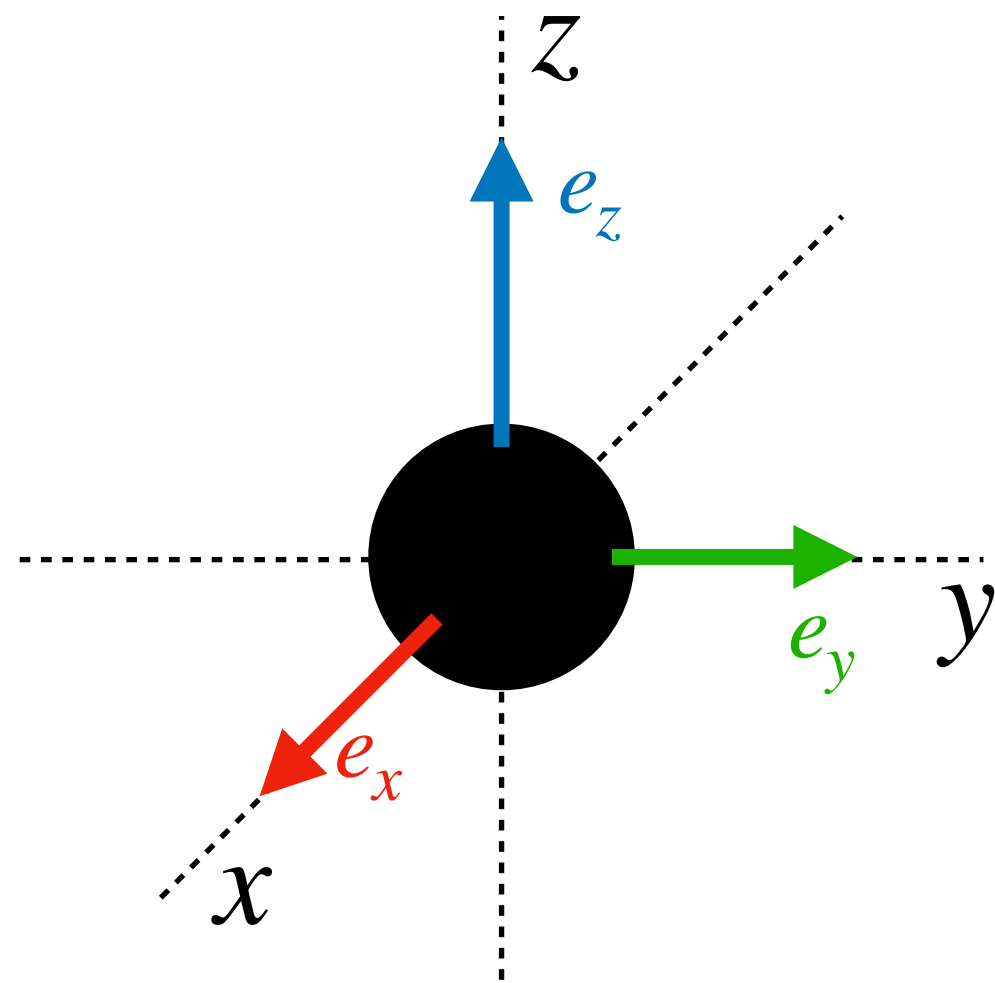


$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \\ ? \end{bmatrix}$$

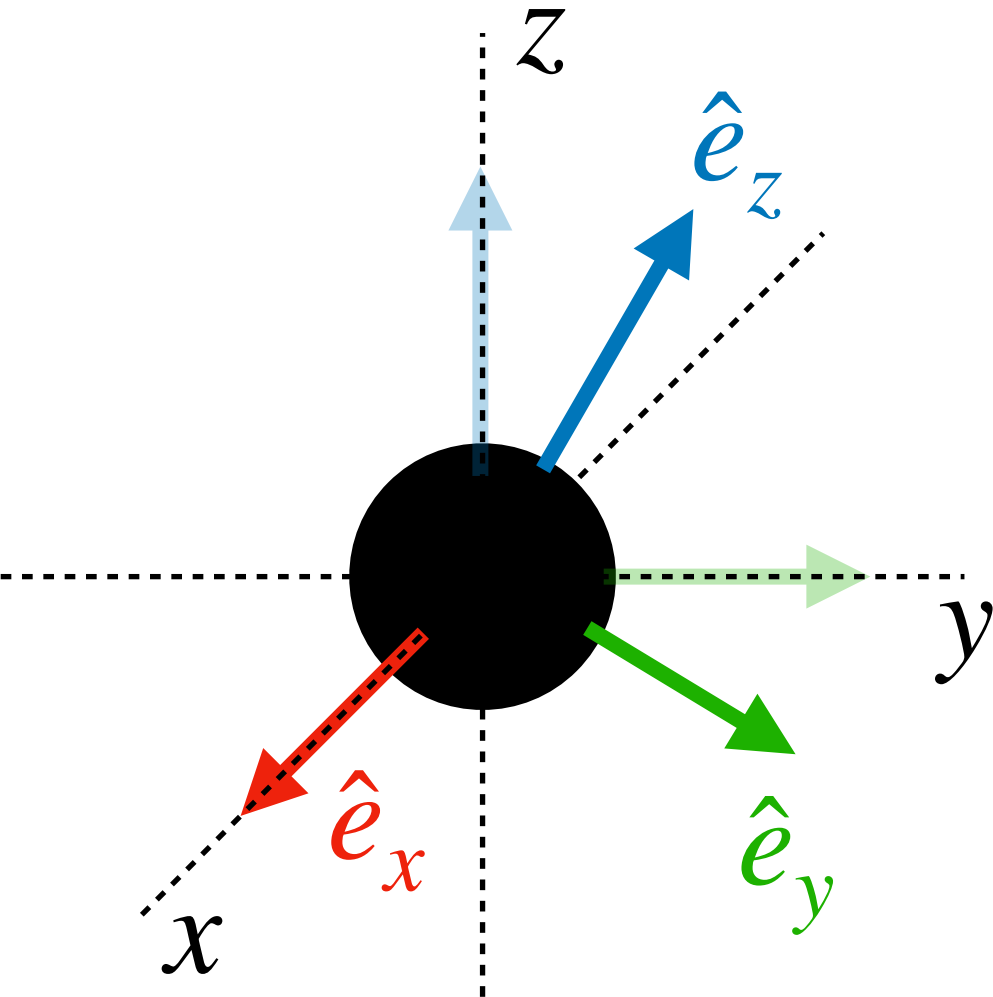
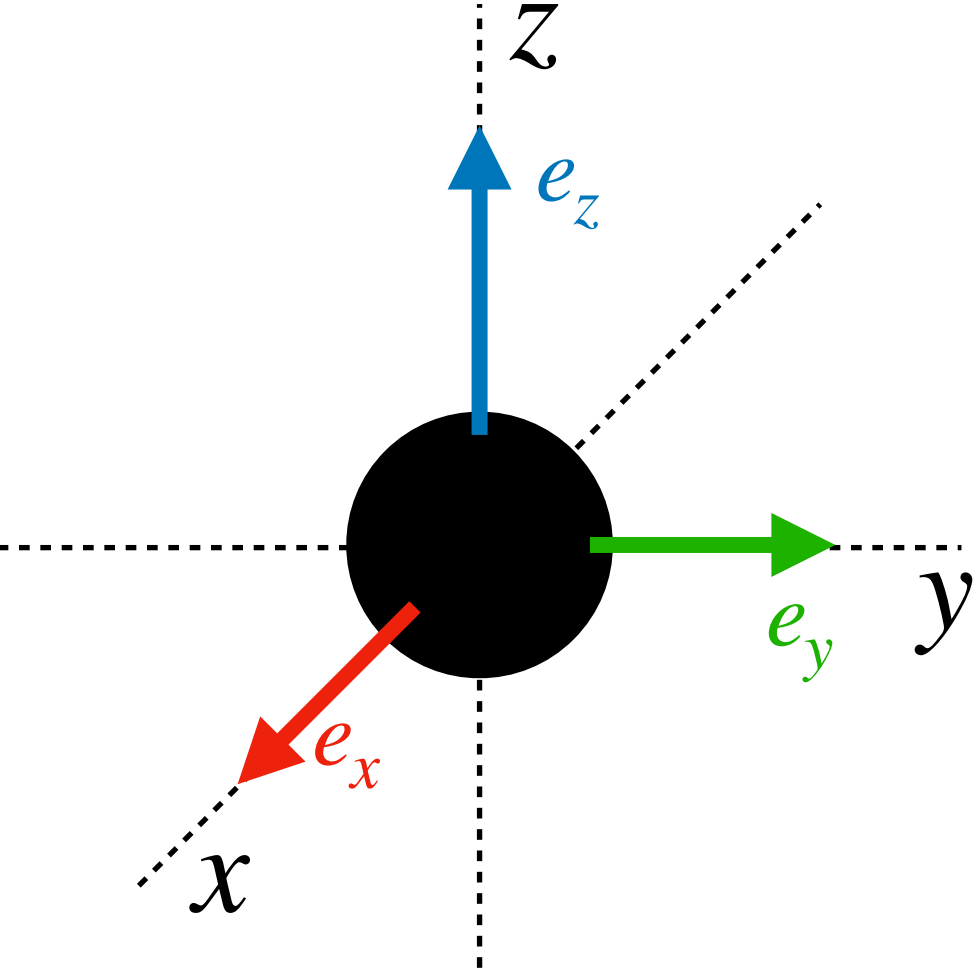
Rotation Representation



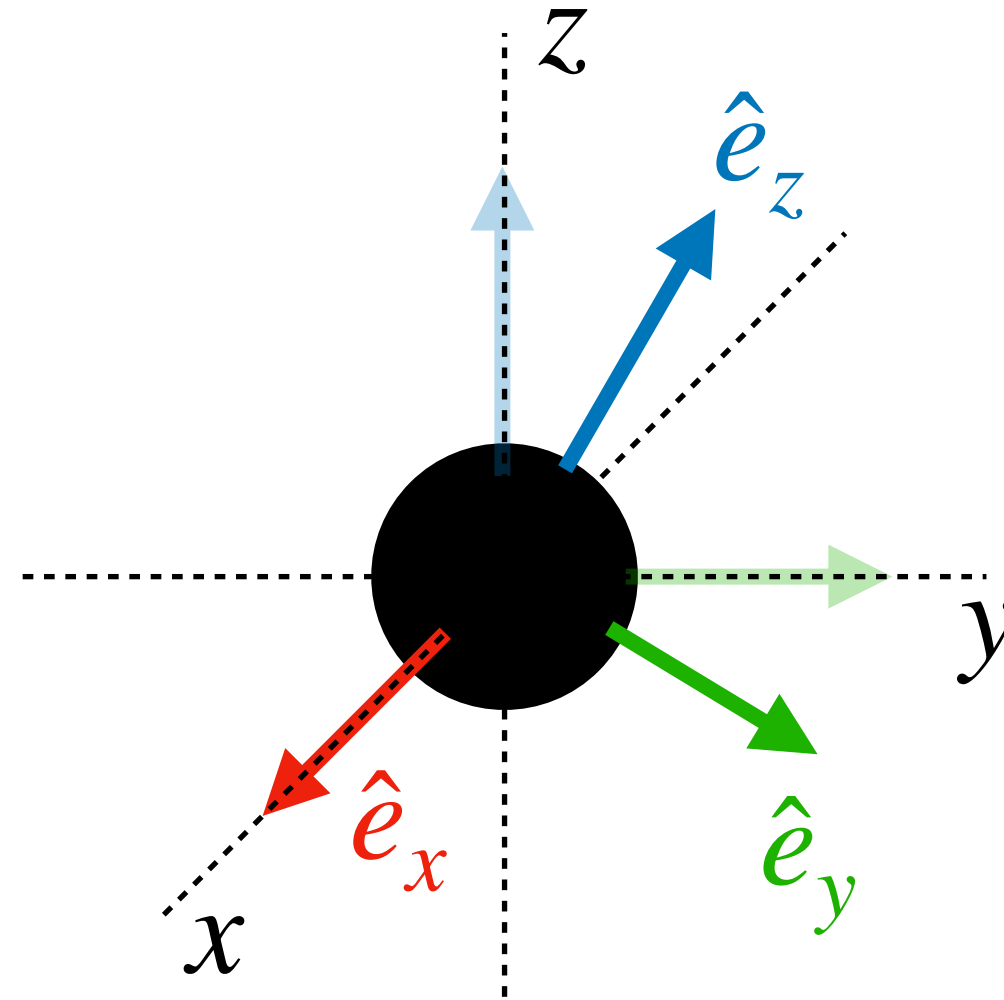
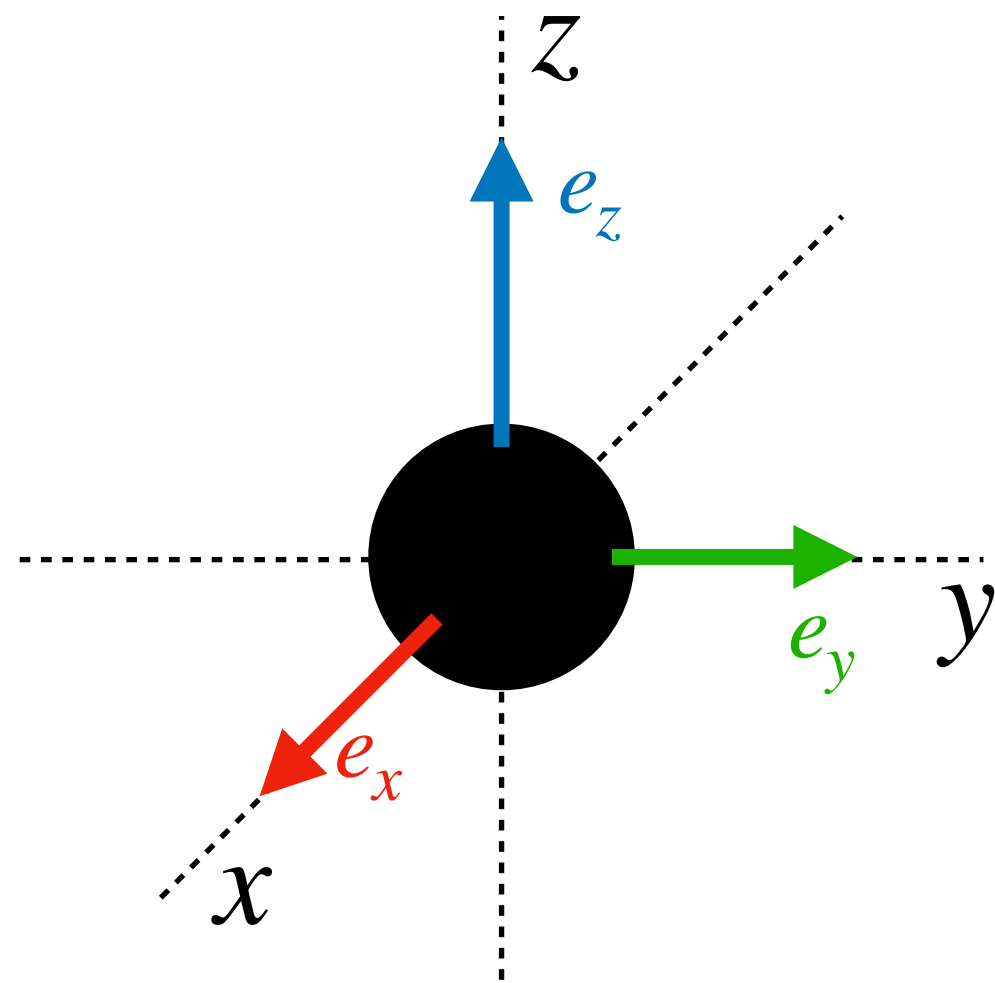
Rotation Representation



Rotation Representation

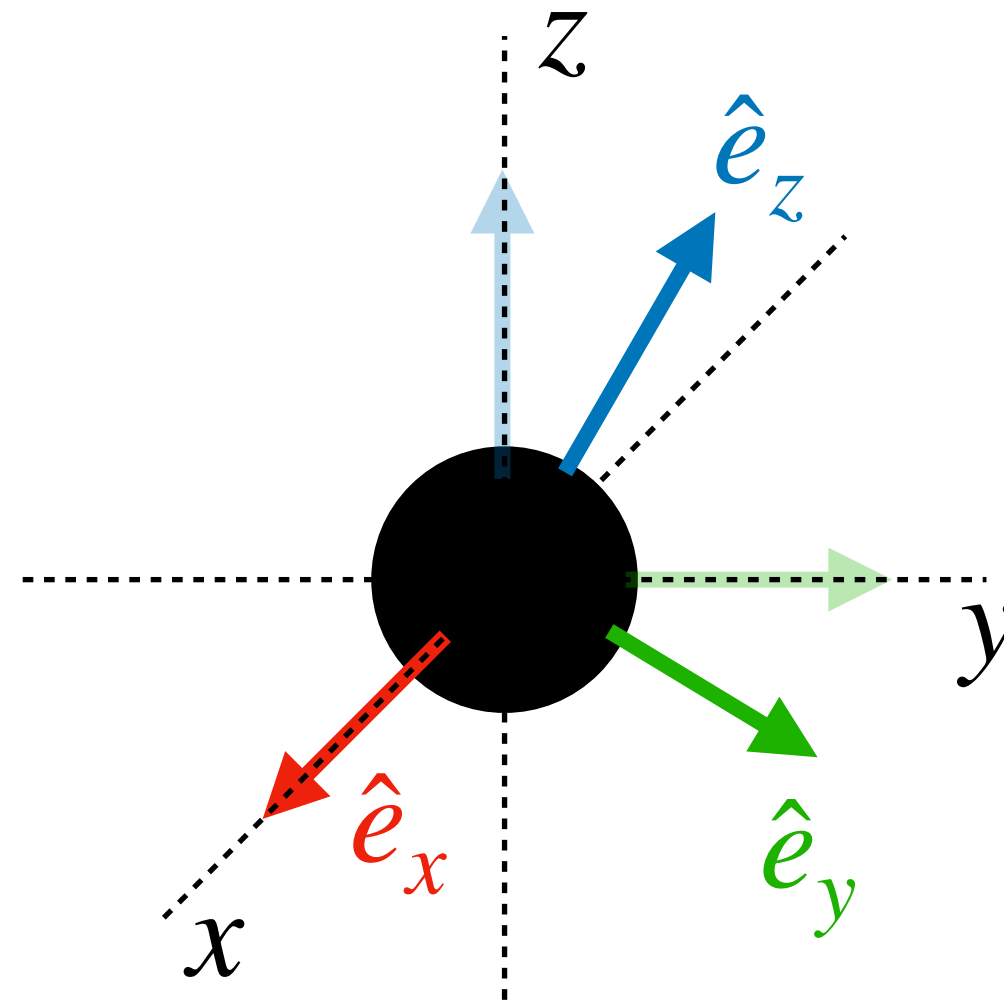
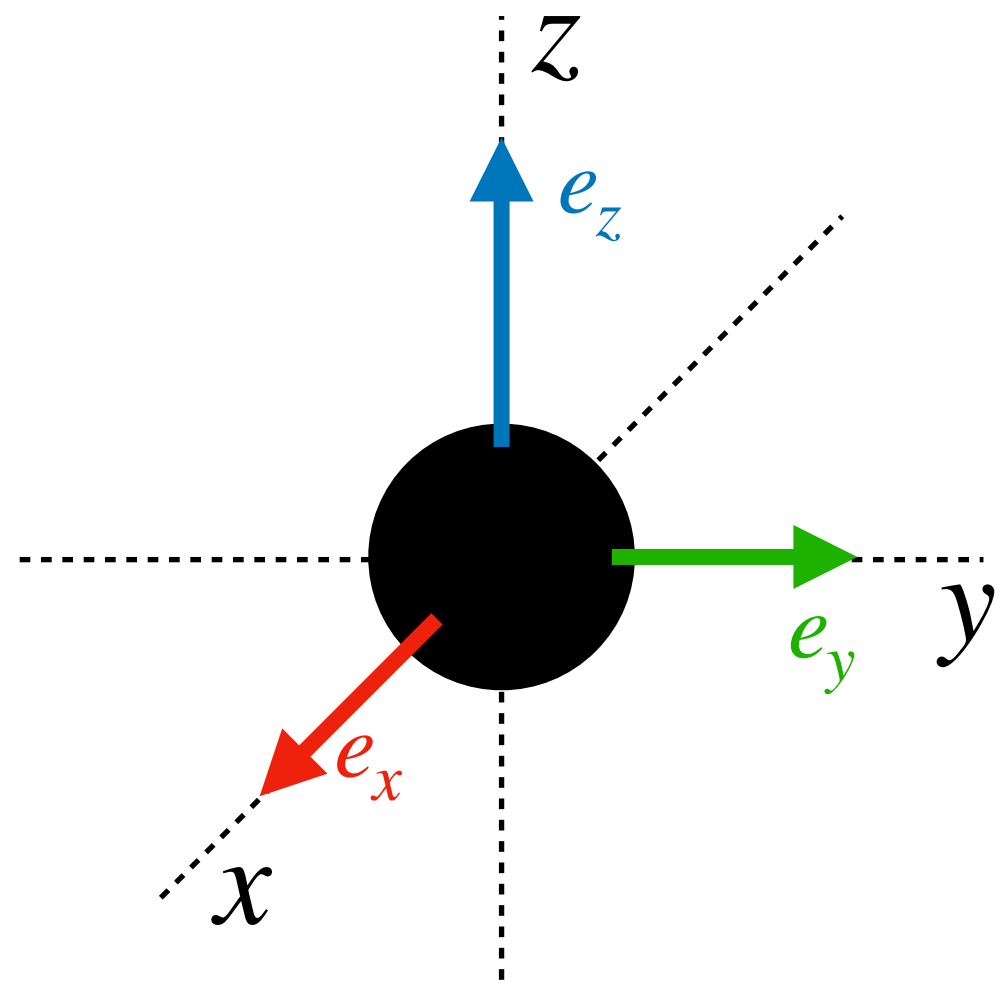


Rotation Representation



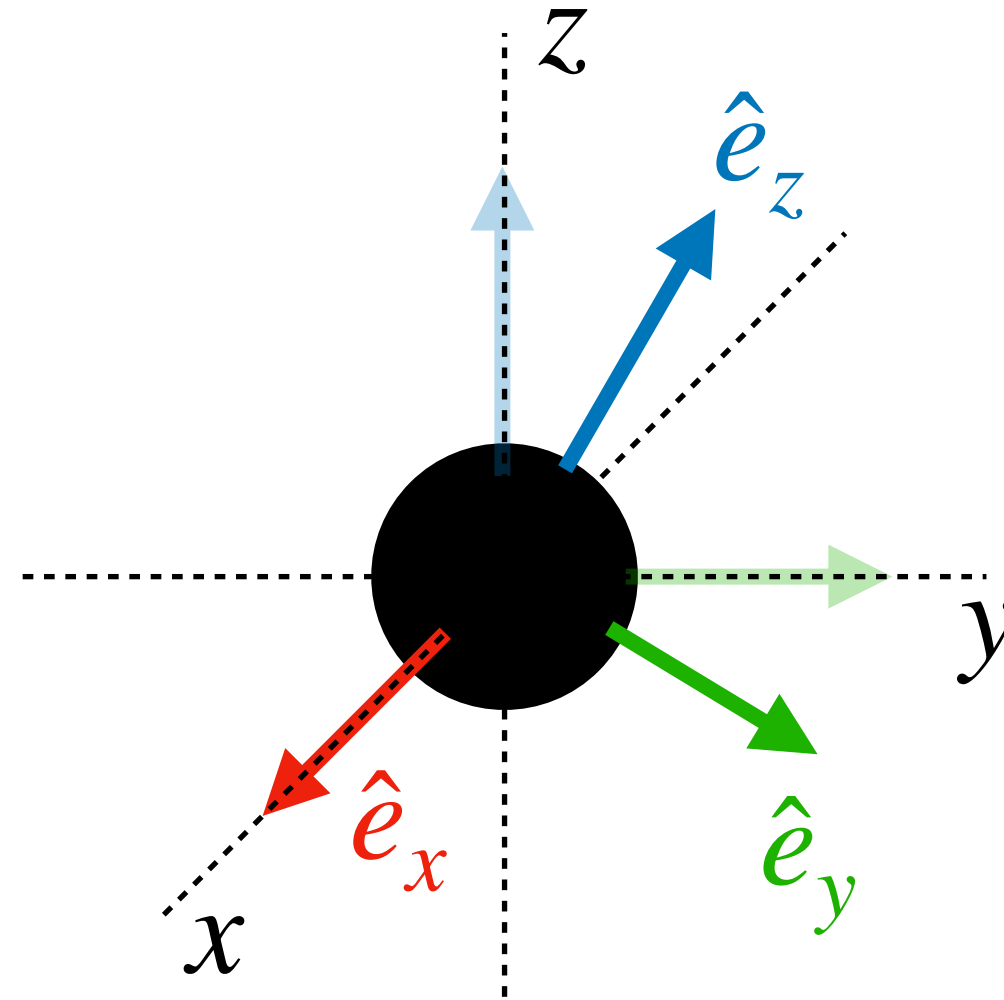
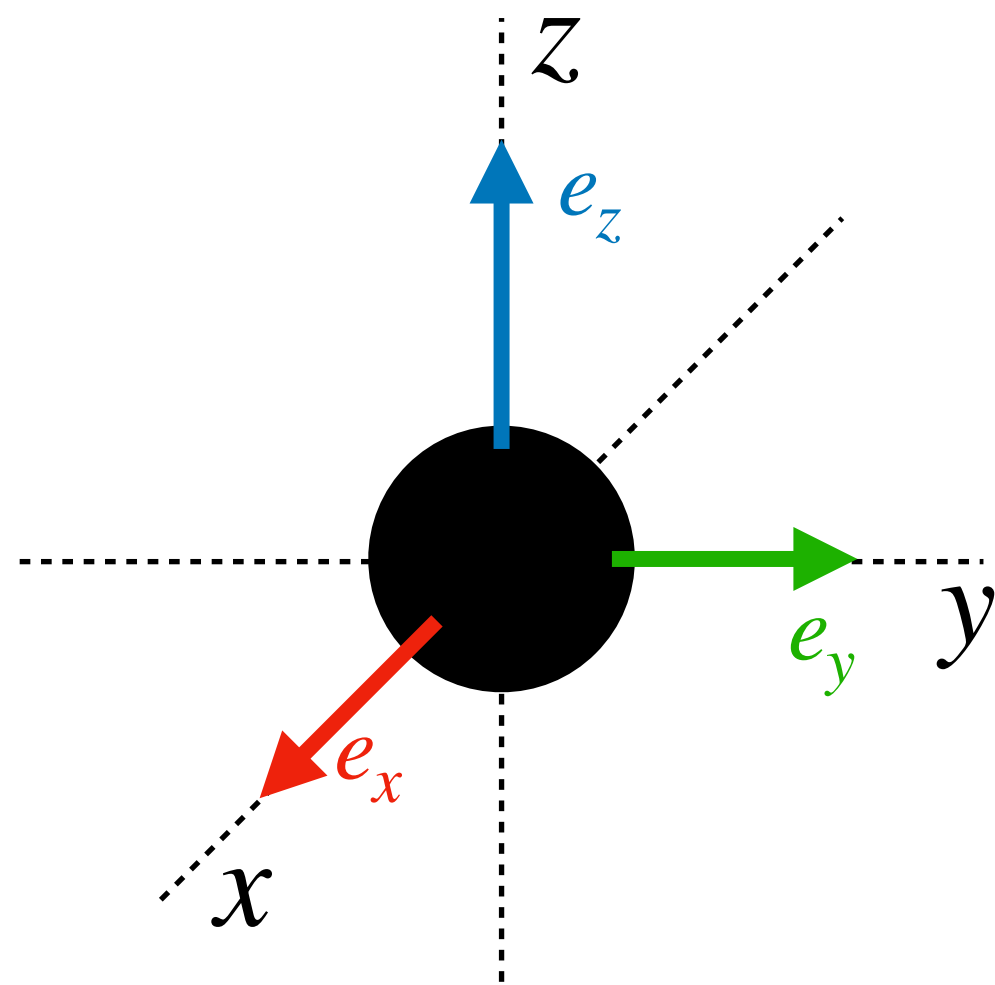
$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & 0.5 \\ 0 & -0.5 & 0.866 \end{bmatrix}$$

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

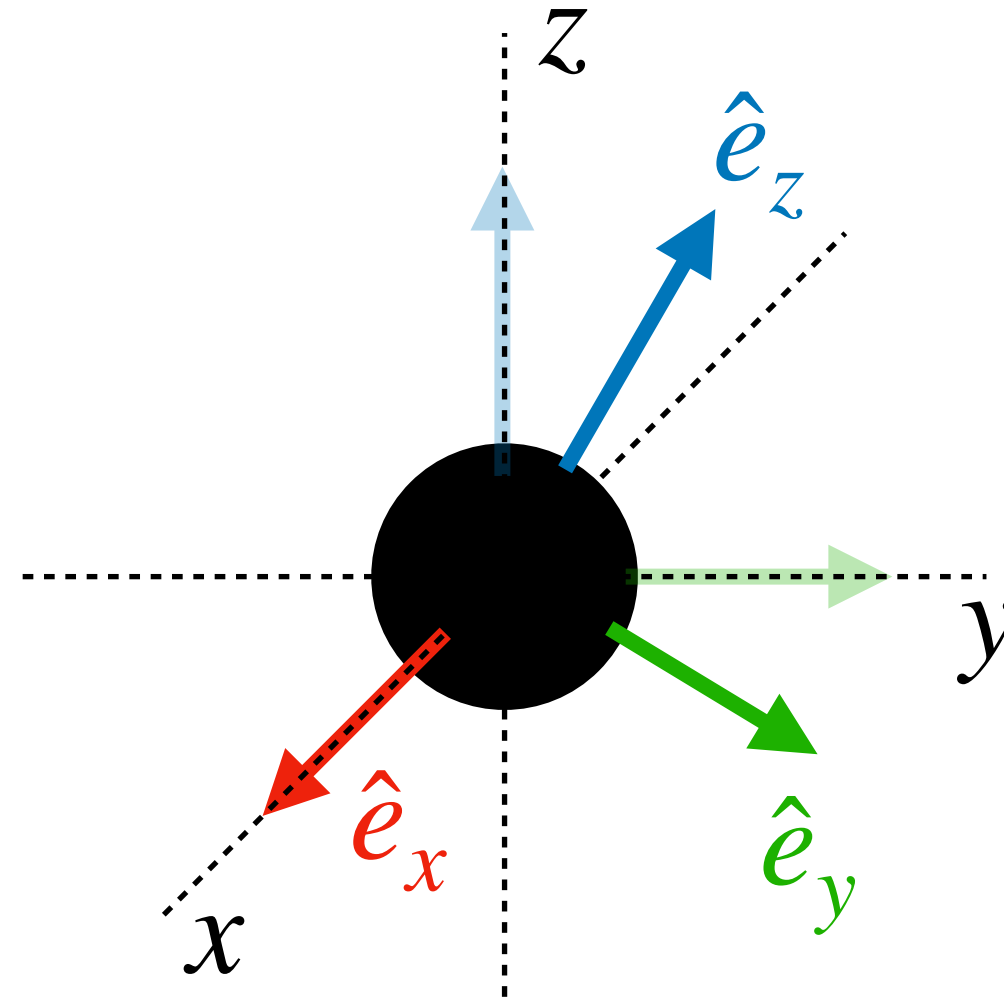
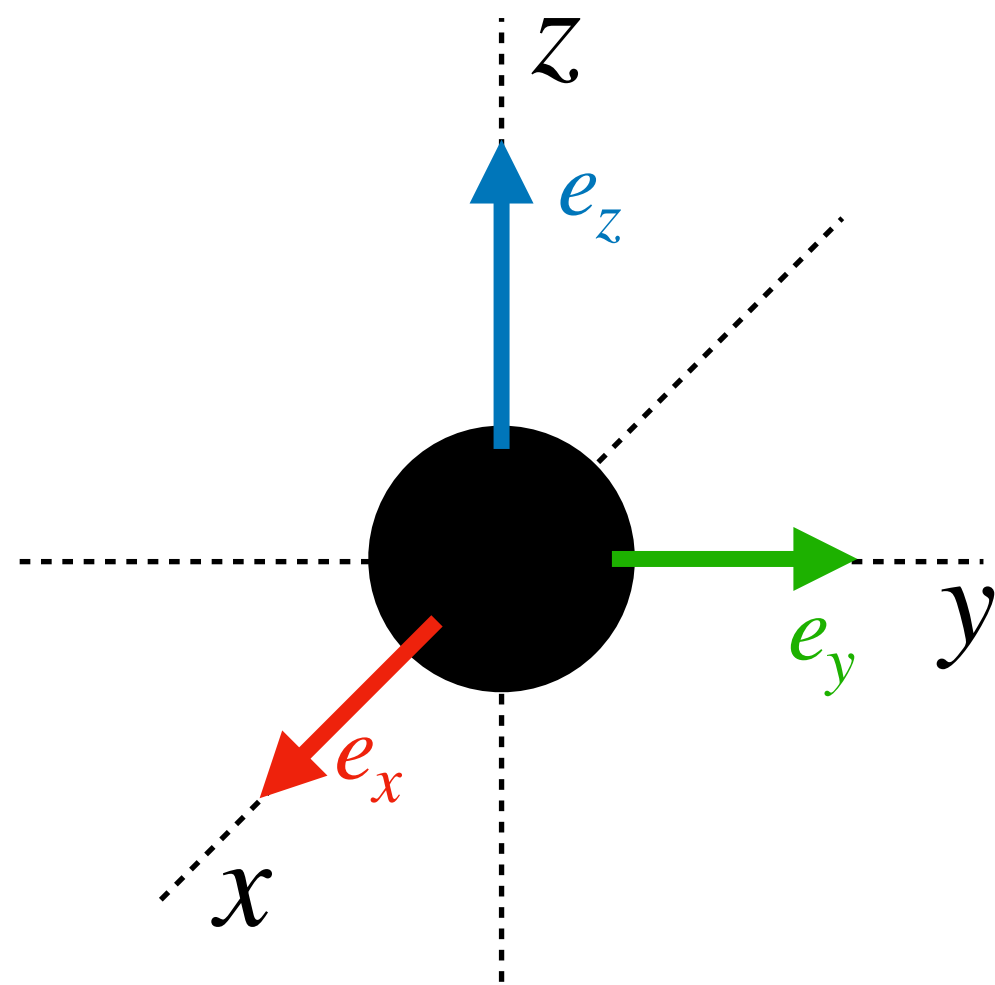
Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*

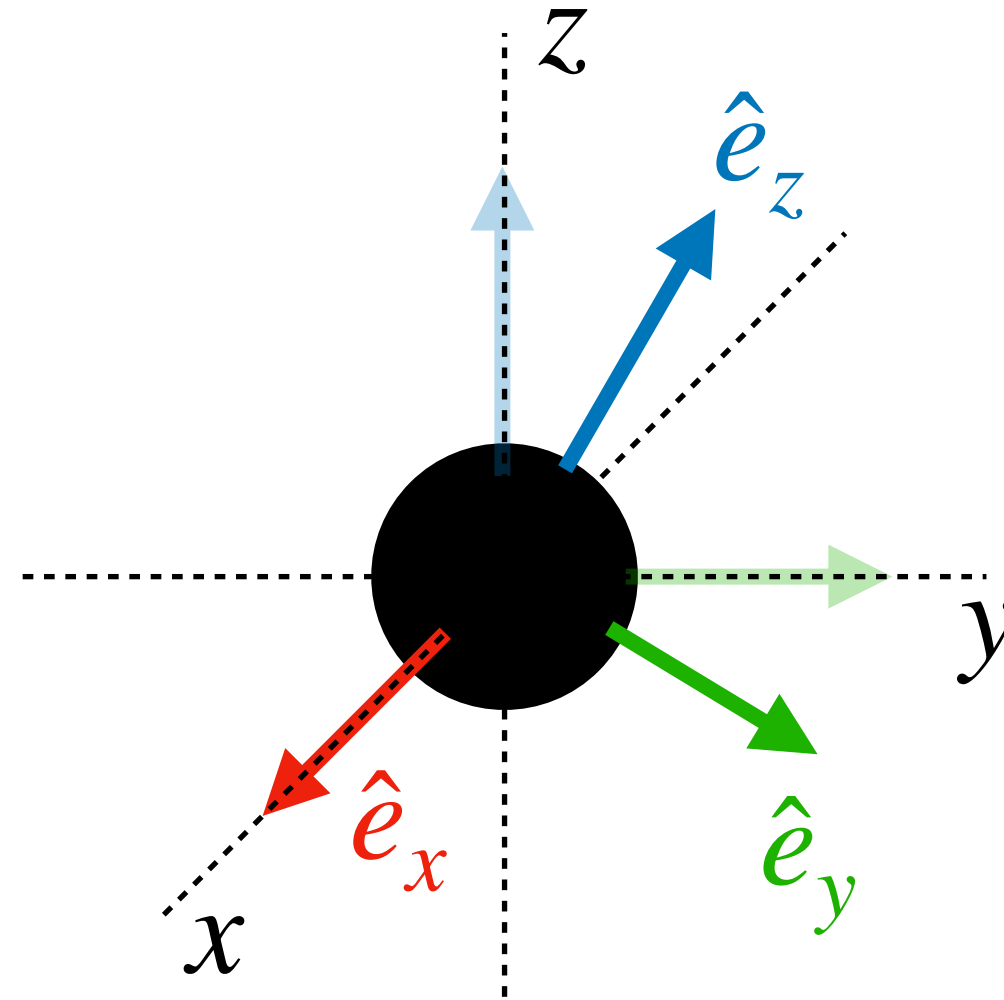
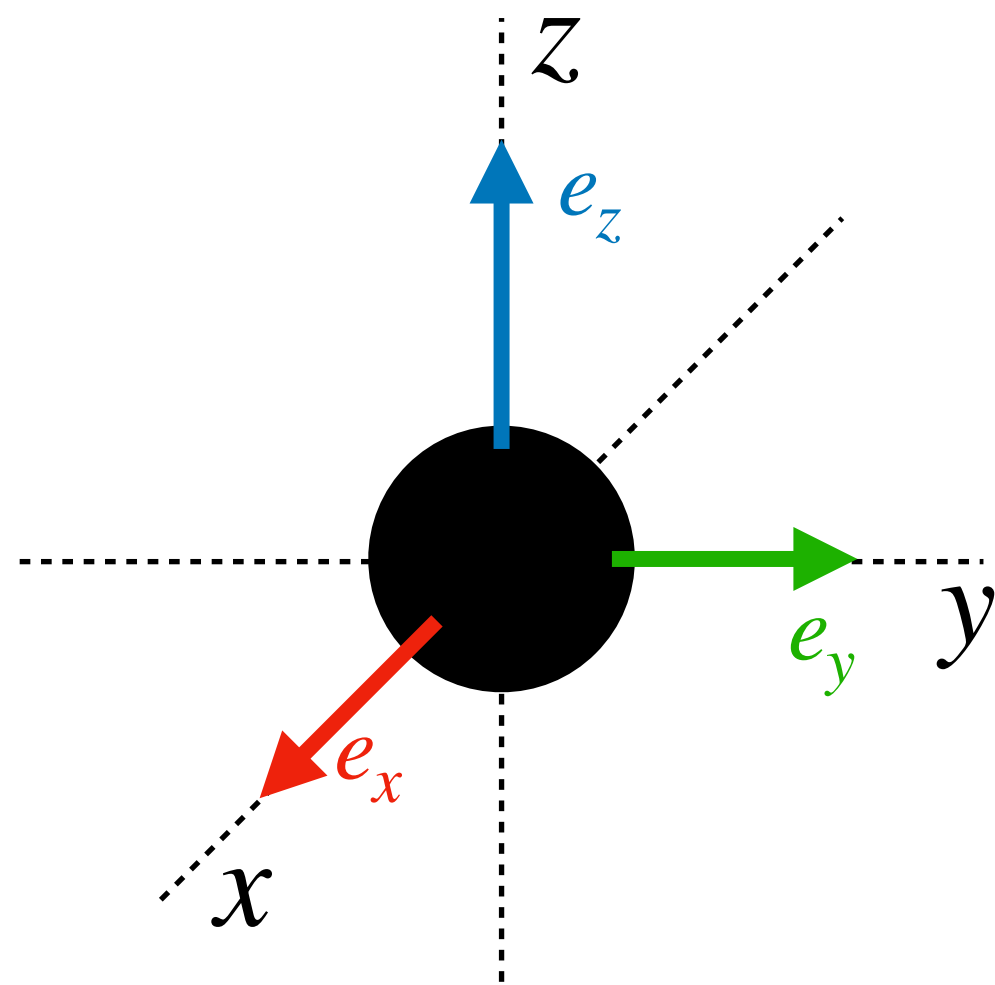
Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:

Group (mathematics)

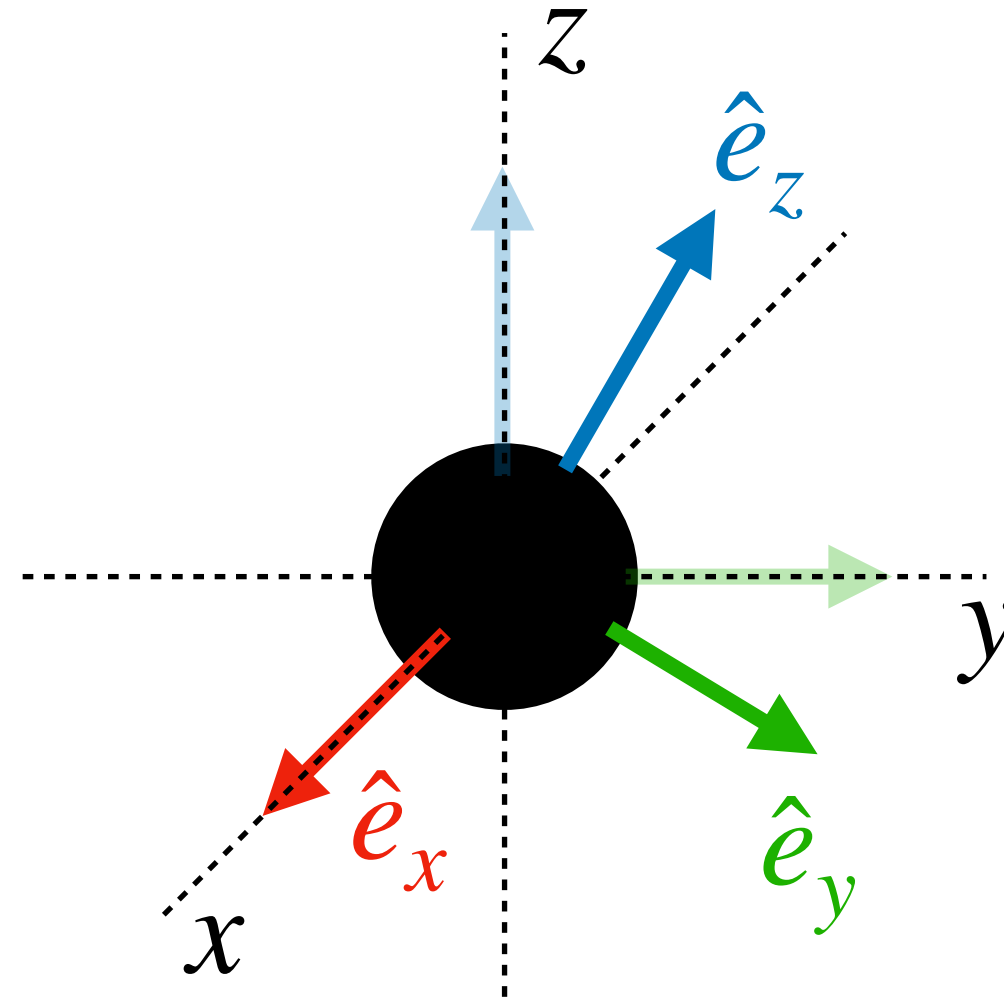
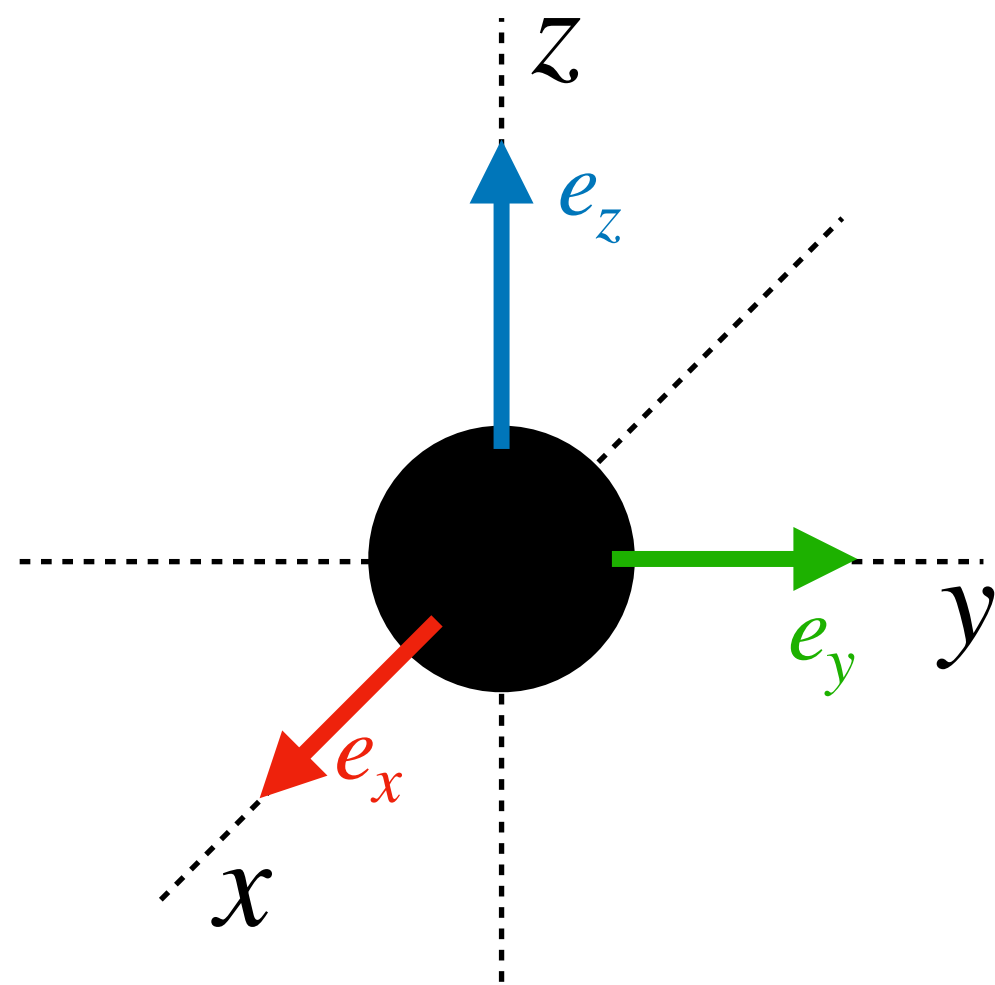
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is a [set](#) with an [operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:

Group (mathematics)

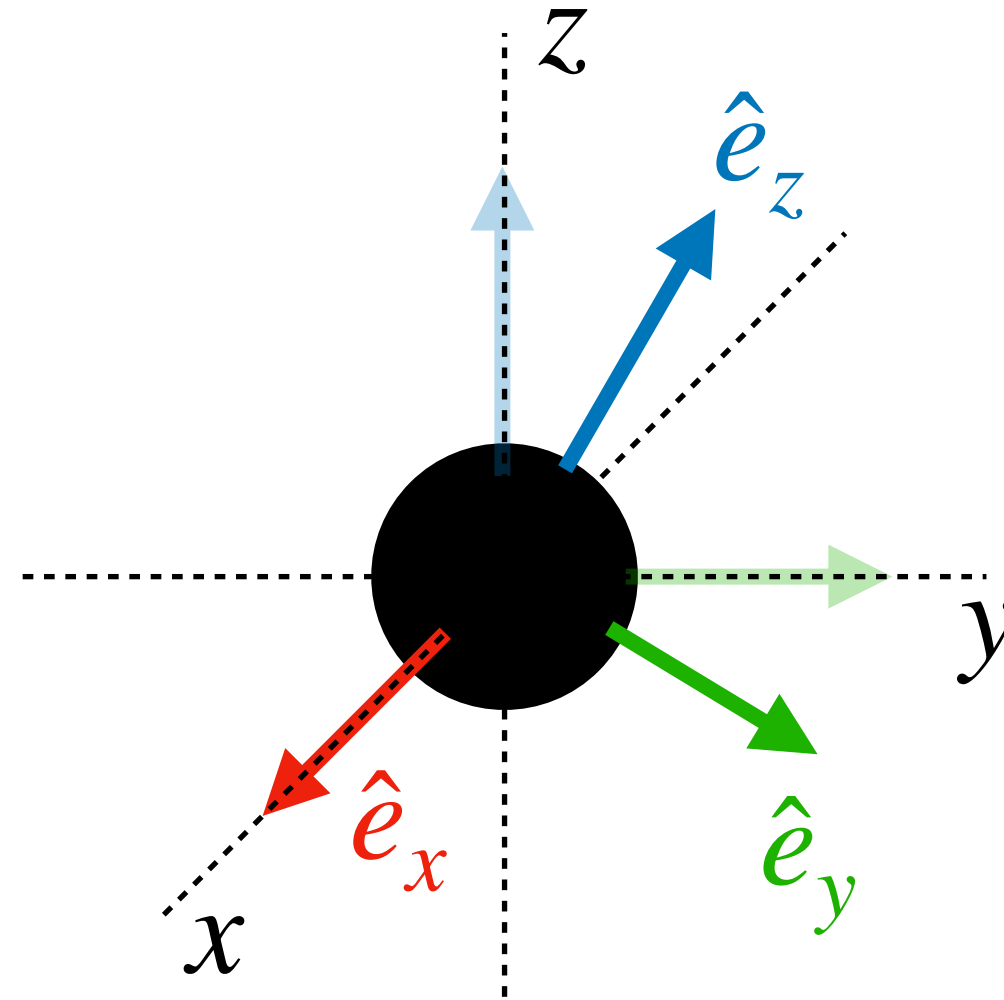
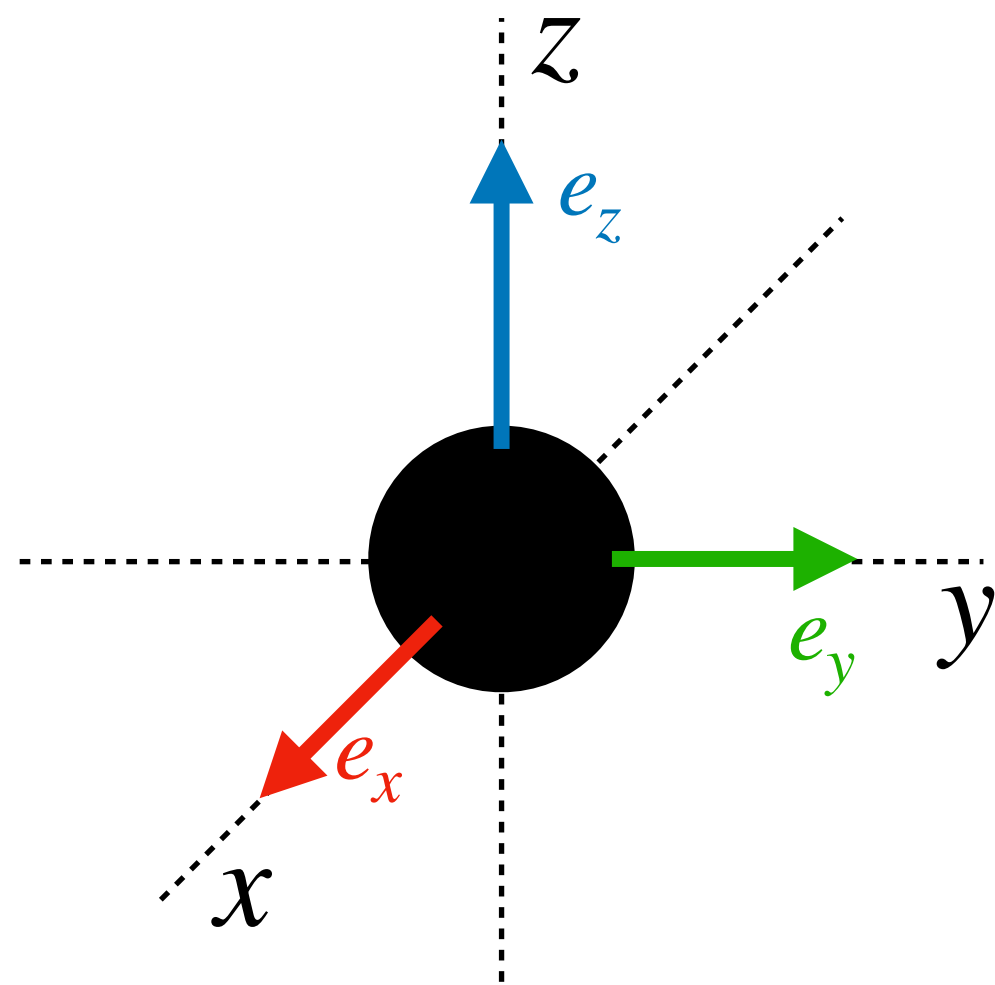
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with an [operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:

Group (mathematics)

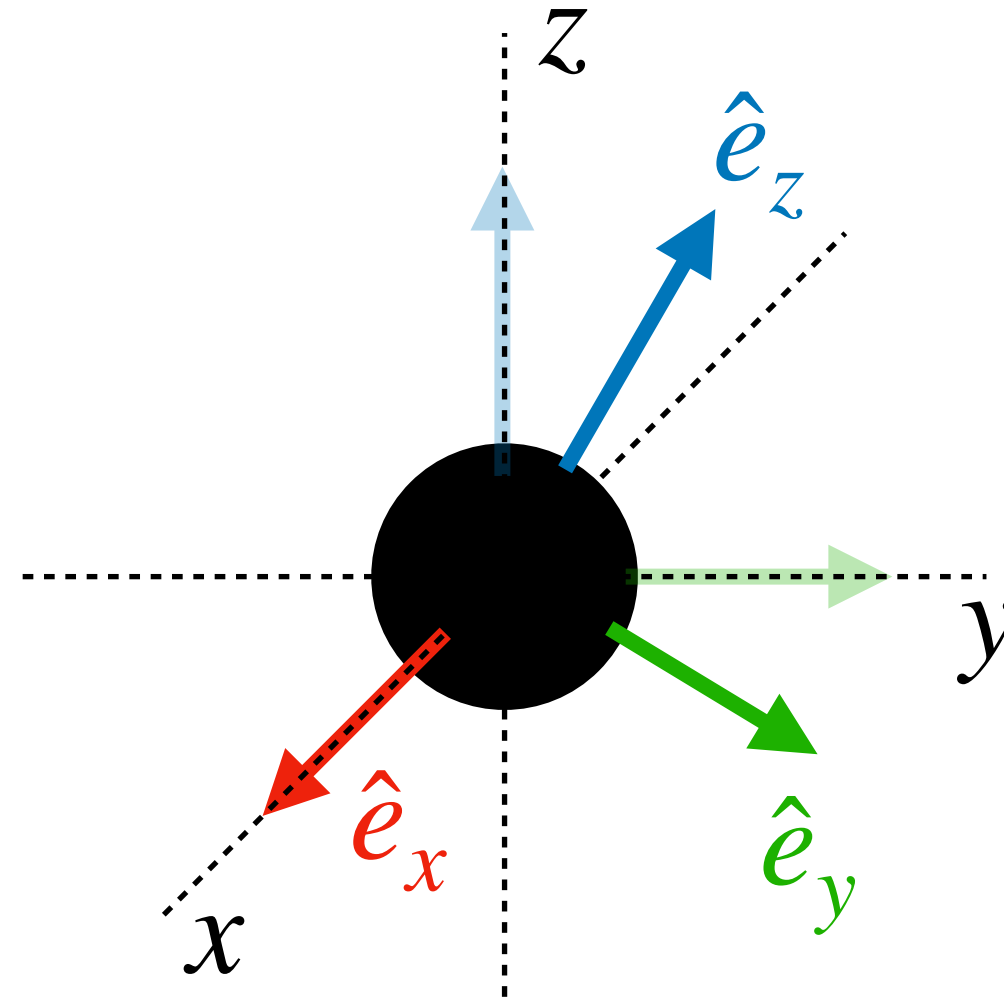
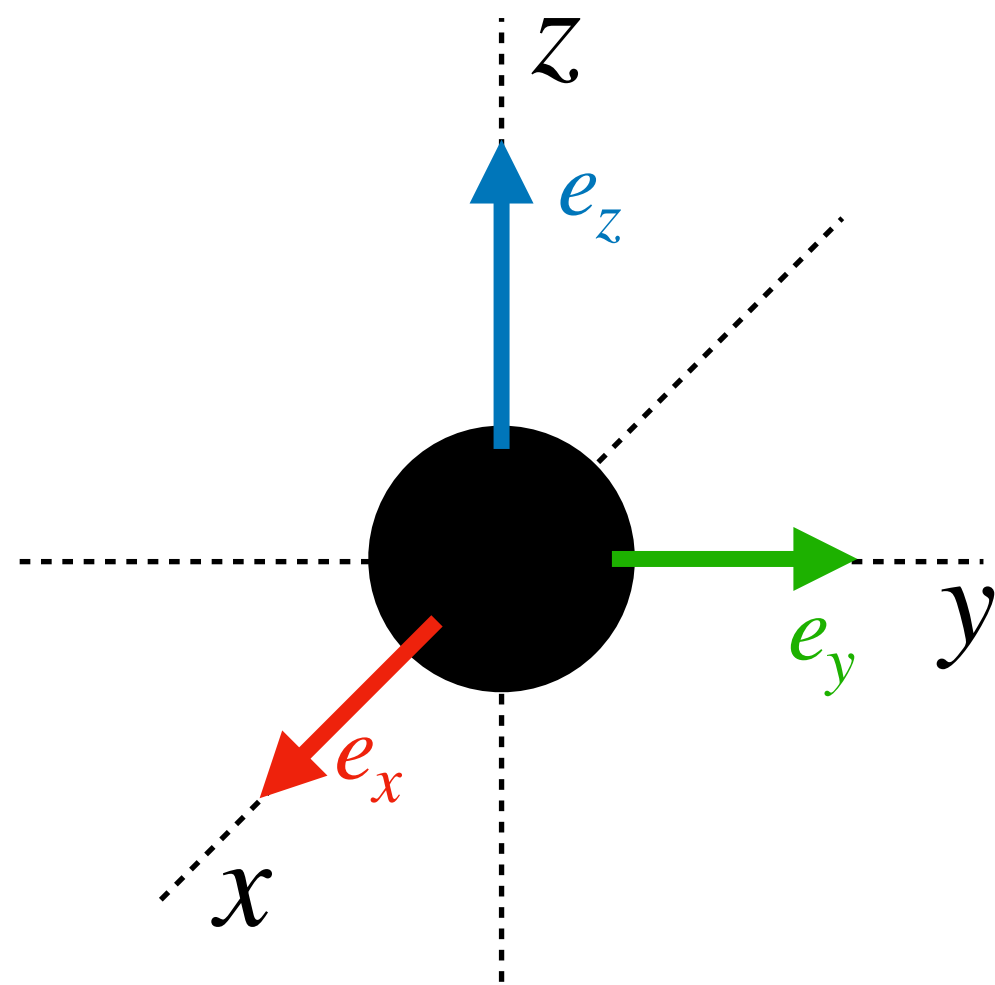
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:

Group (mathematics)

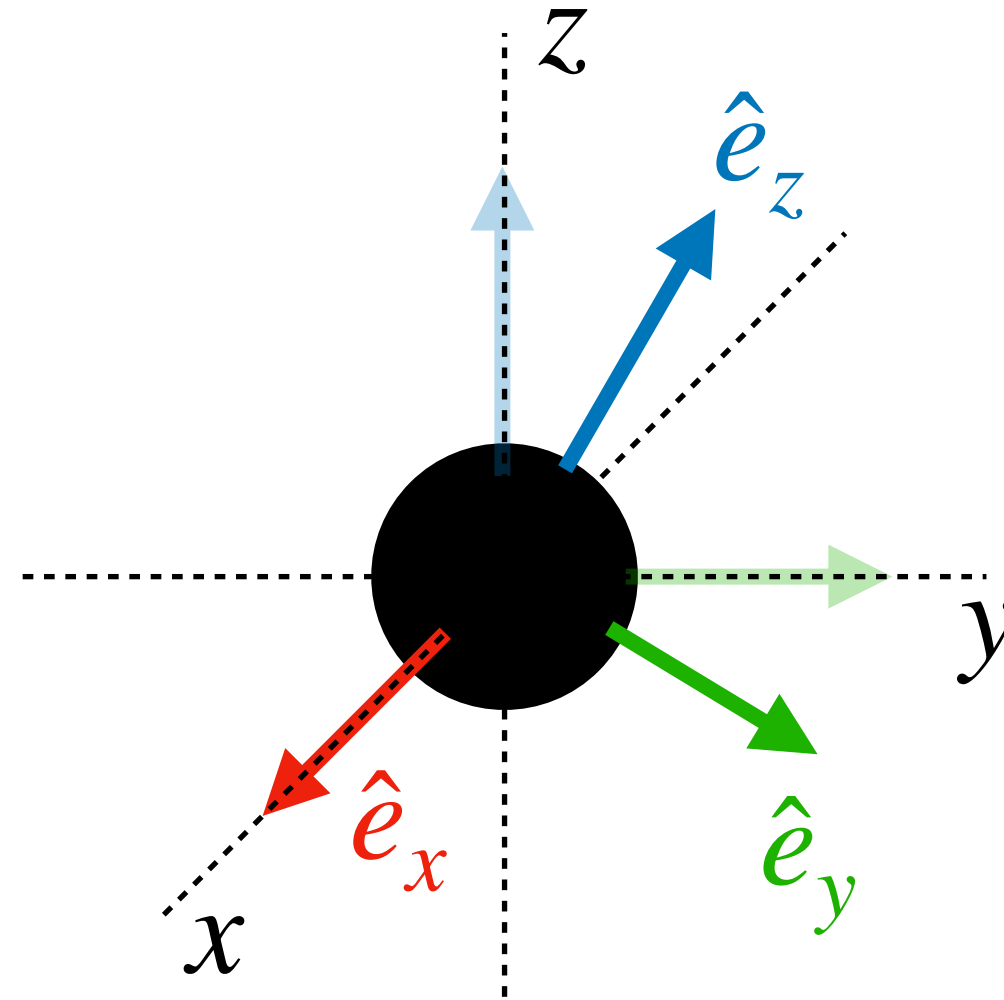
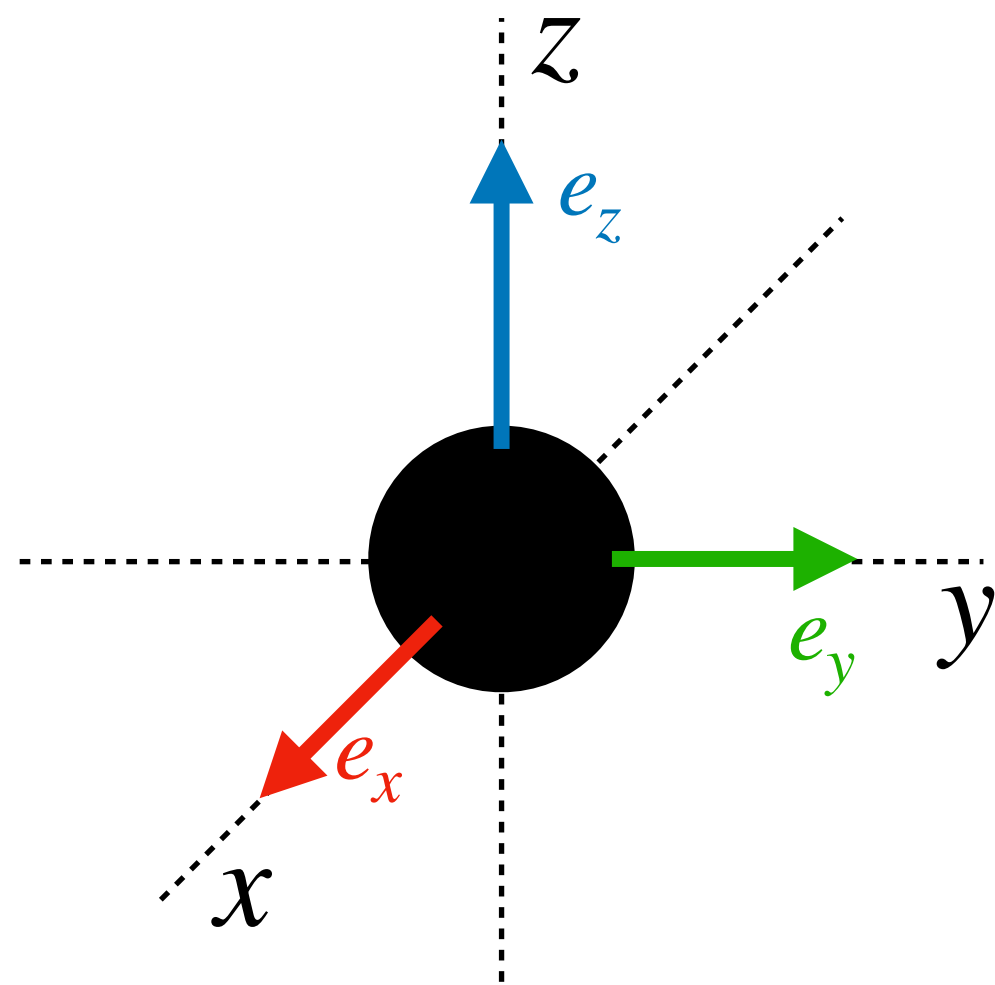
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:

Group (mathematics)

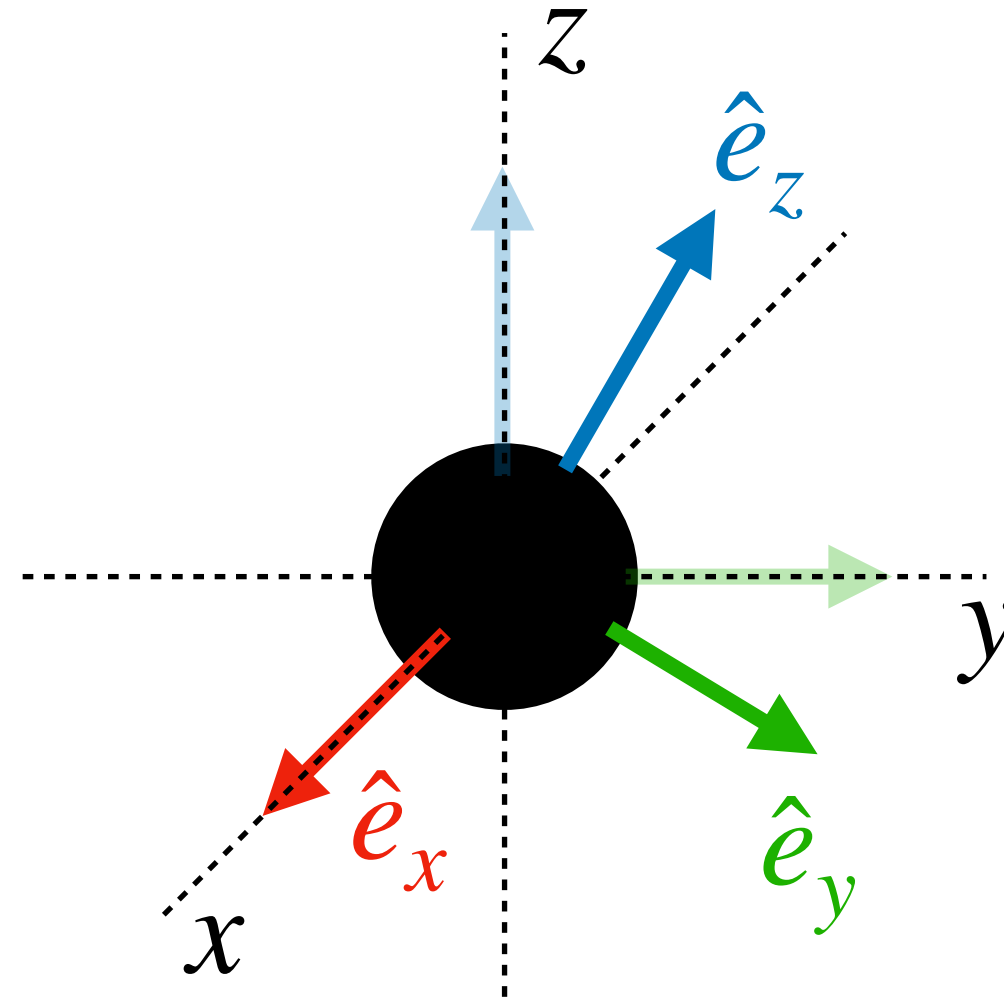
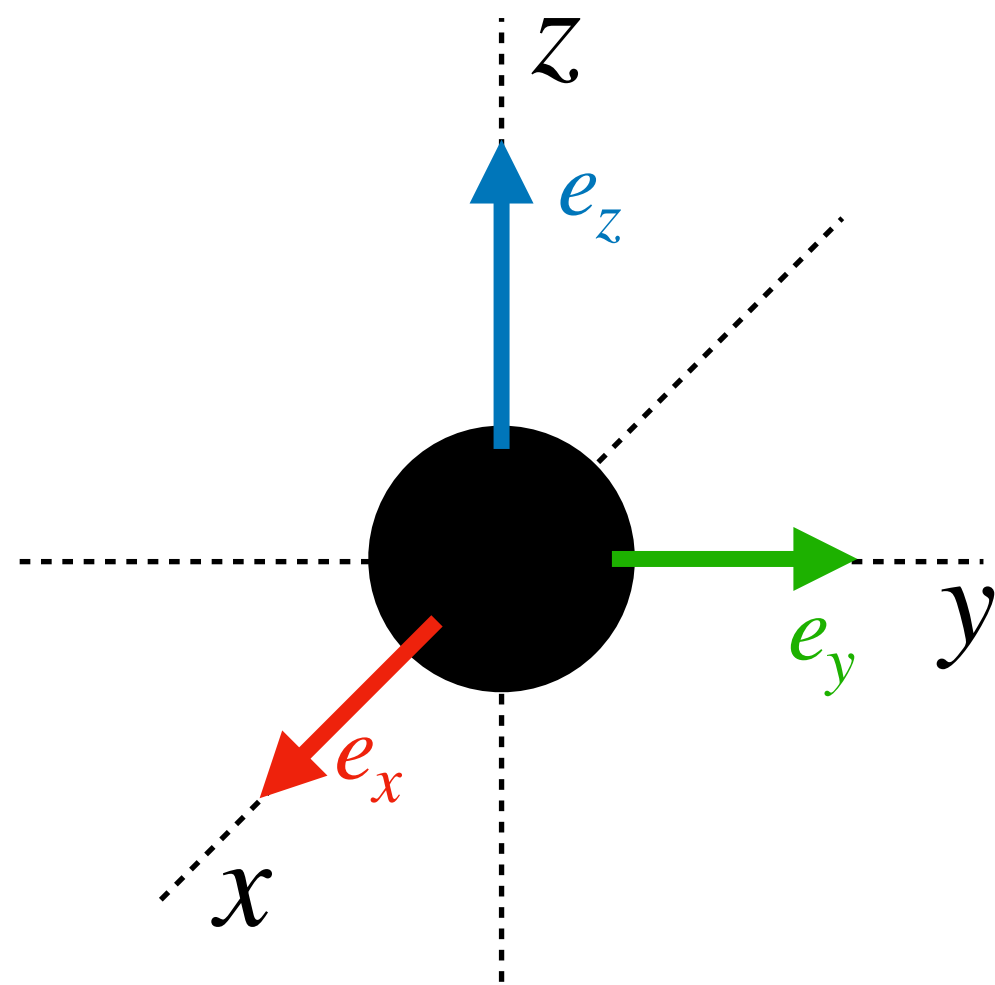
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:

Group (mathematics)

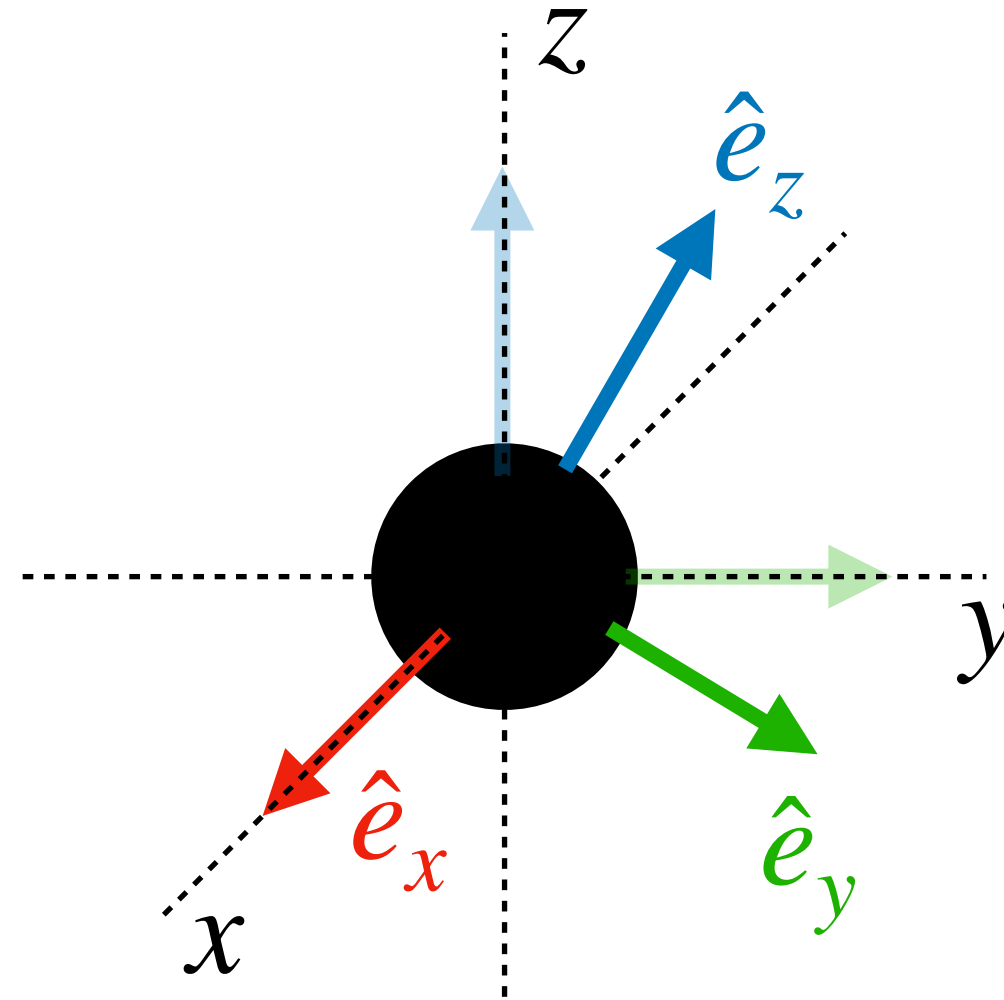
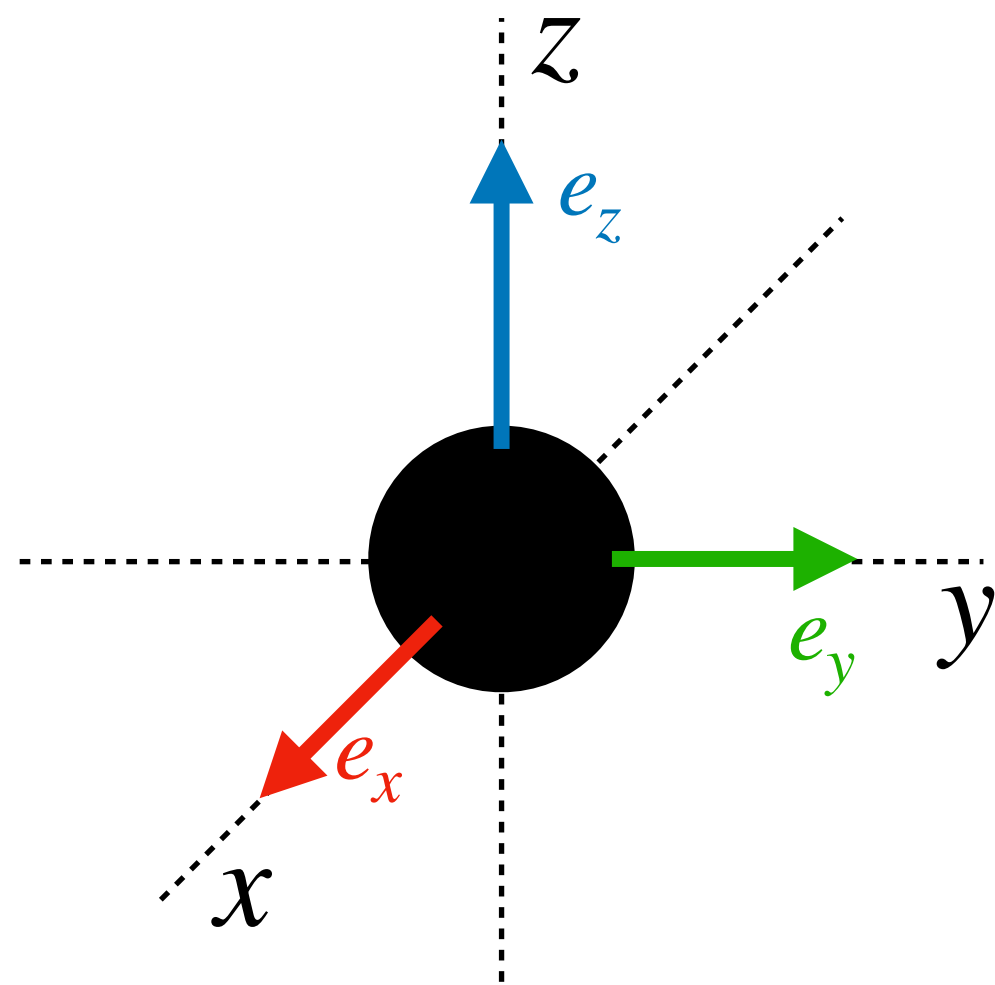
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:
 - Set: $\{R \in \mathbb{R}^{3 \times 3} \mid R^T R = \mathbb{I}_3, \det R = 1\}$

Group (mathematics)

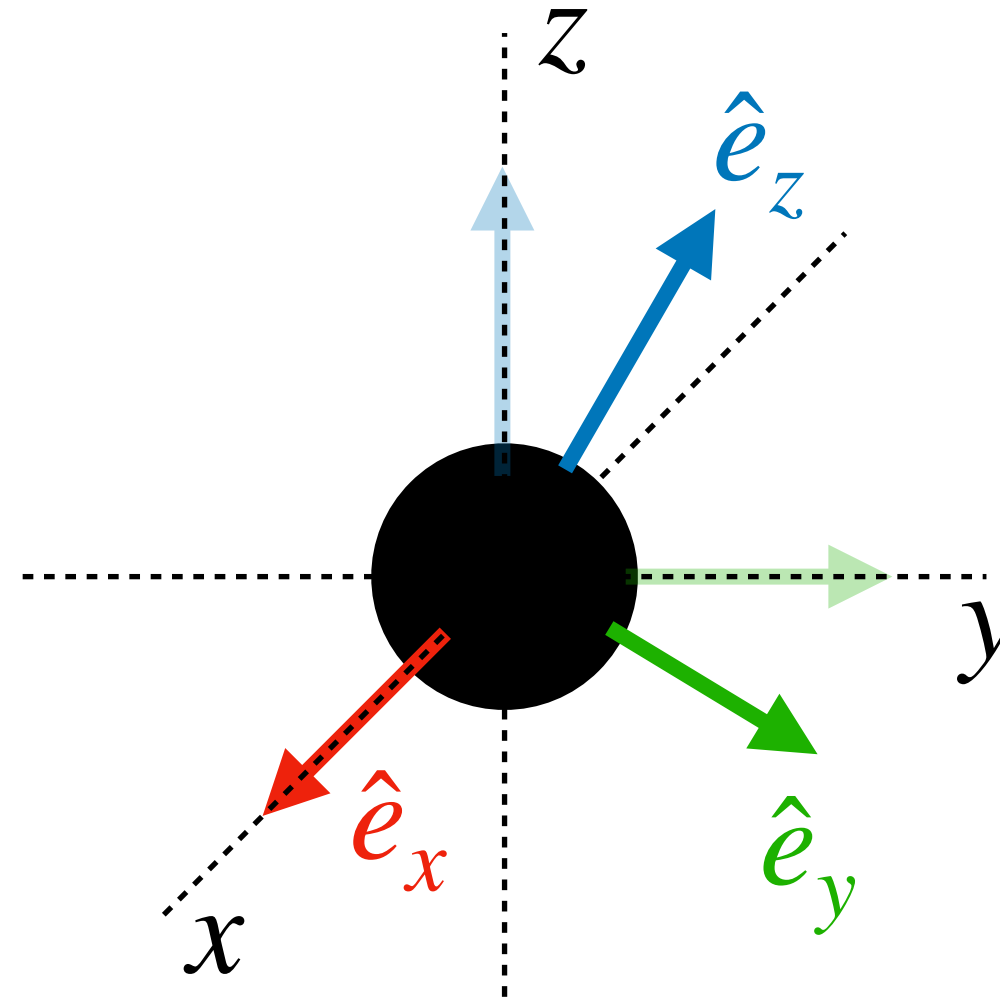
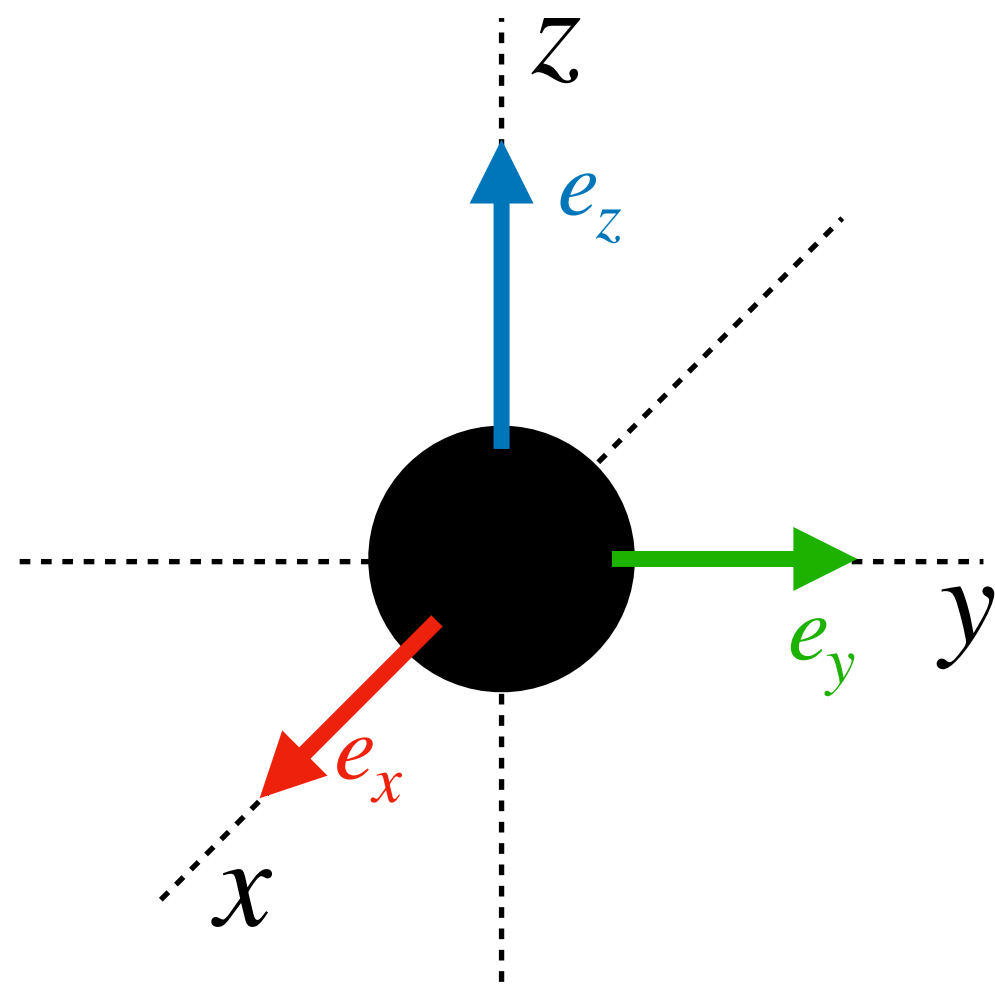
[Article](#) [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:
 - Set: $\{R \in \mathbb{R}^{3 \times 3} \mid R^T R = \mathbb{I}_3, \det R = 1\}$
 - Associative Operation: Matrix Multiplication $R_a = R_b R_c R_d$

Group (mathematics)

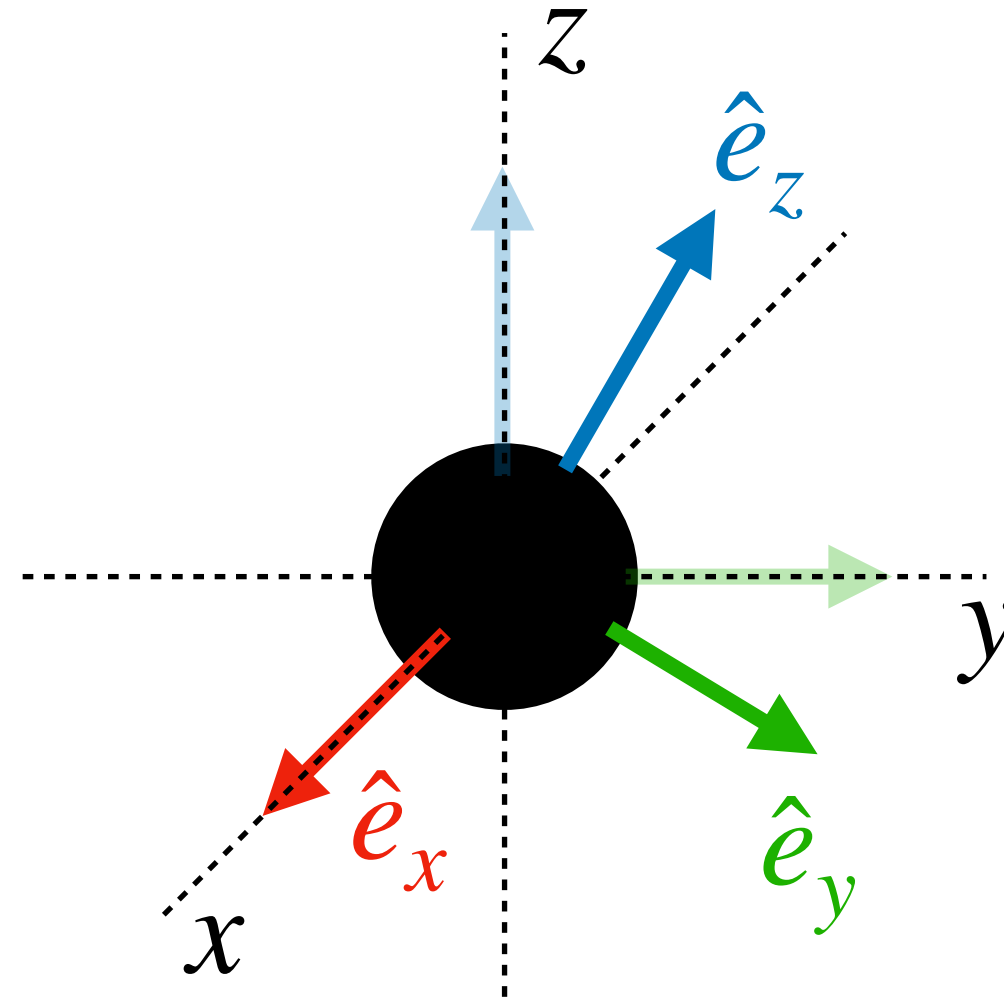
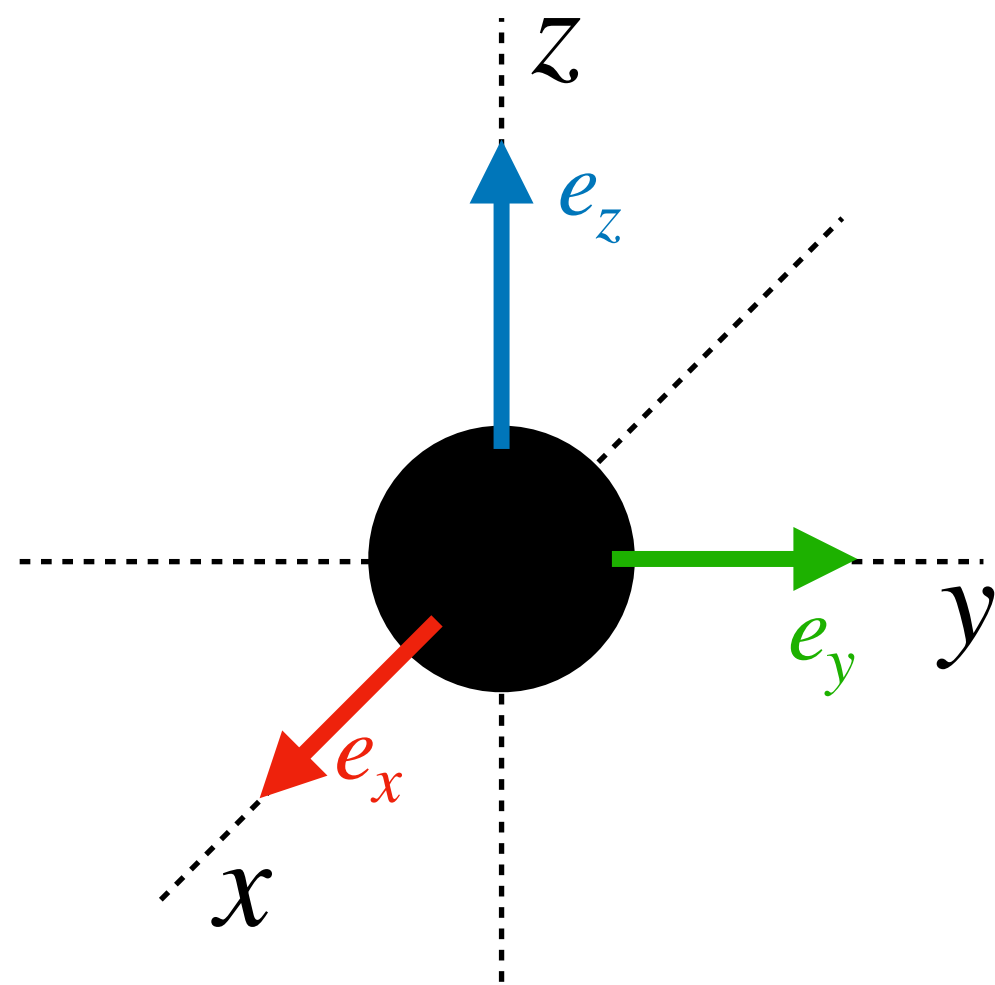
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:
 - Set: $\{R \in \mathbb{R}^{3 \times 3} \mid R^T R = \mathbb{I}_3, \det R = 1\}$
 - Associative Operation: Matrix Multiplication $R_a = R_b R_c R_d$
 - **Q:** Neutral Element: $RE = R$, $E =$

Group (mathematics)

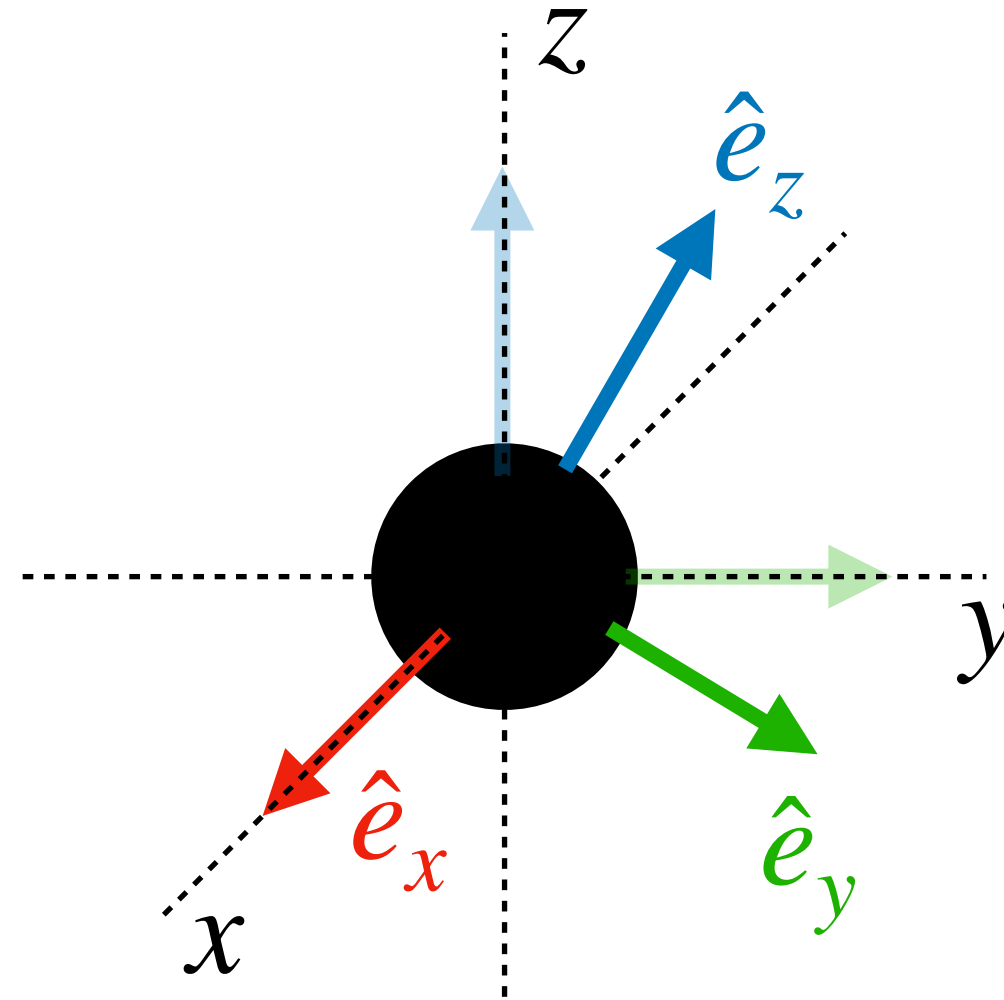
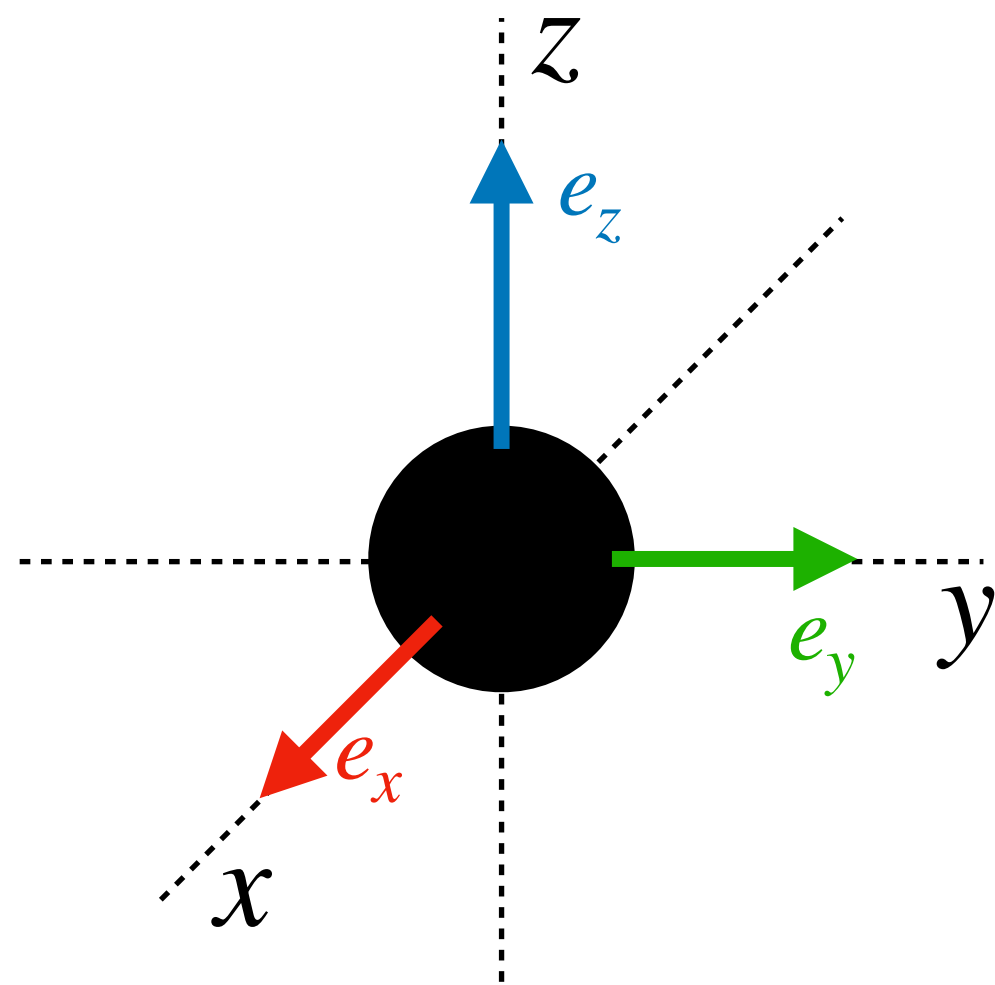
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:
 - Set: $\{R \in \mathbb{R}^{3 \times 3} \mid R^T R = \mathbb{I}_3, \det R = 1\}$
 - Associative Operation: Matrix Multiplication $R_a = R_b R_c R_d$
 - Q: Neutral Element: $RE = R$, $E = \mathbb{I}_3$ — “don't rotate”

Group (mathematics)

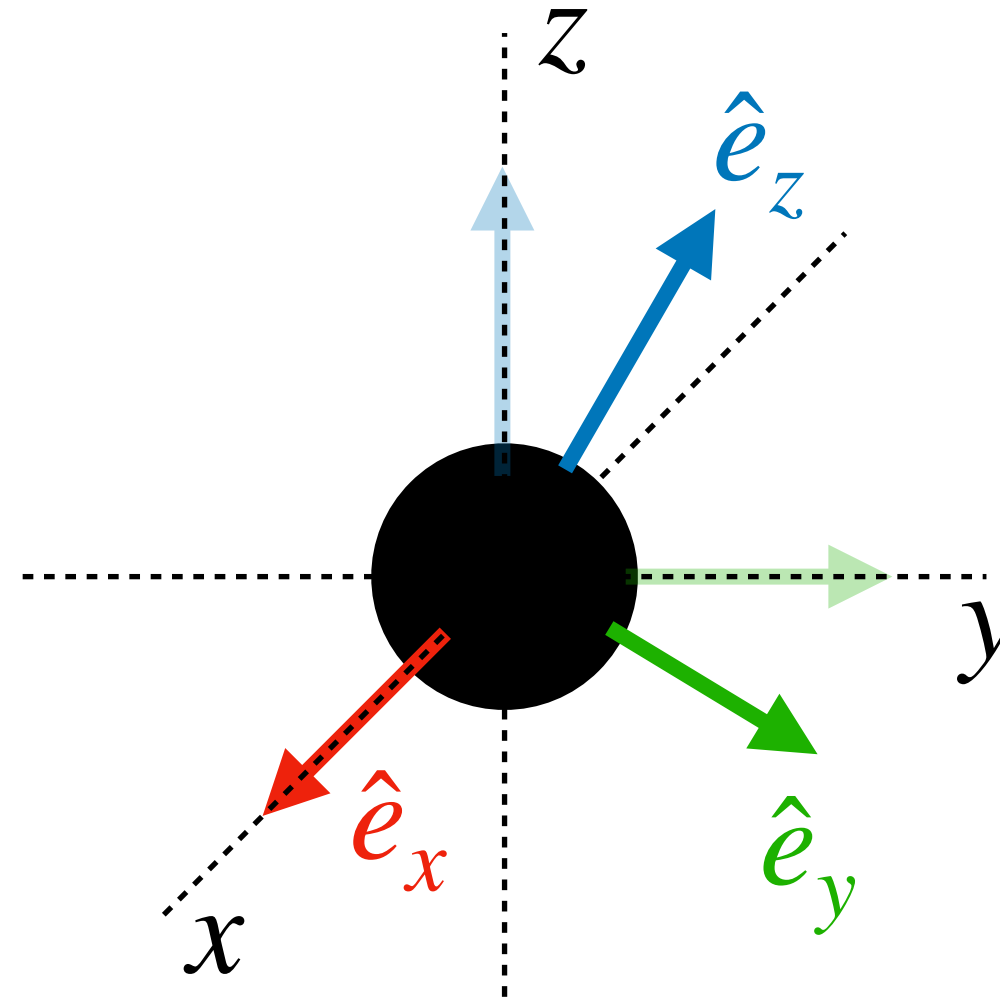
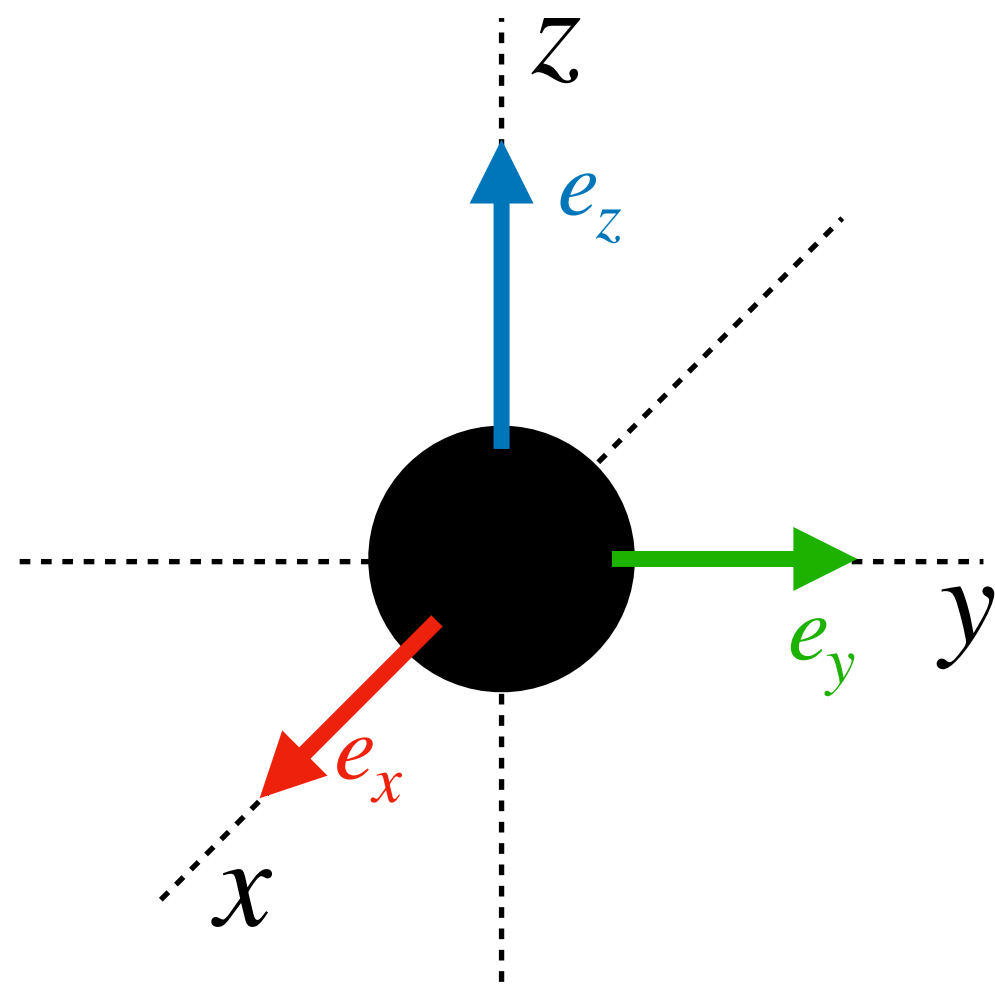
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:
 - Set: $\{R \in \mathbb{R}^{3 \times 3} \mid R^T R = \mathbb{I}_3, \det R = 1\}$
 - Associative Operation: Matrix Multiplication $R_a = R_b R_c R_d$
 - Q: Neutral Element: $RE = R$, $E = \mathbb{I}_3$ — “don't rotate”
 - Q: Inverse Element: $IR = \mathbb{I}_3$, $I =$

Group (mathematics)

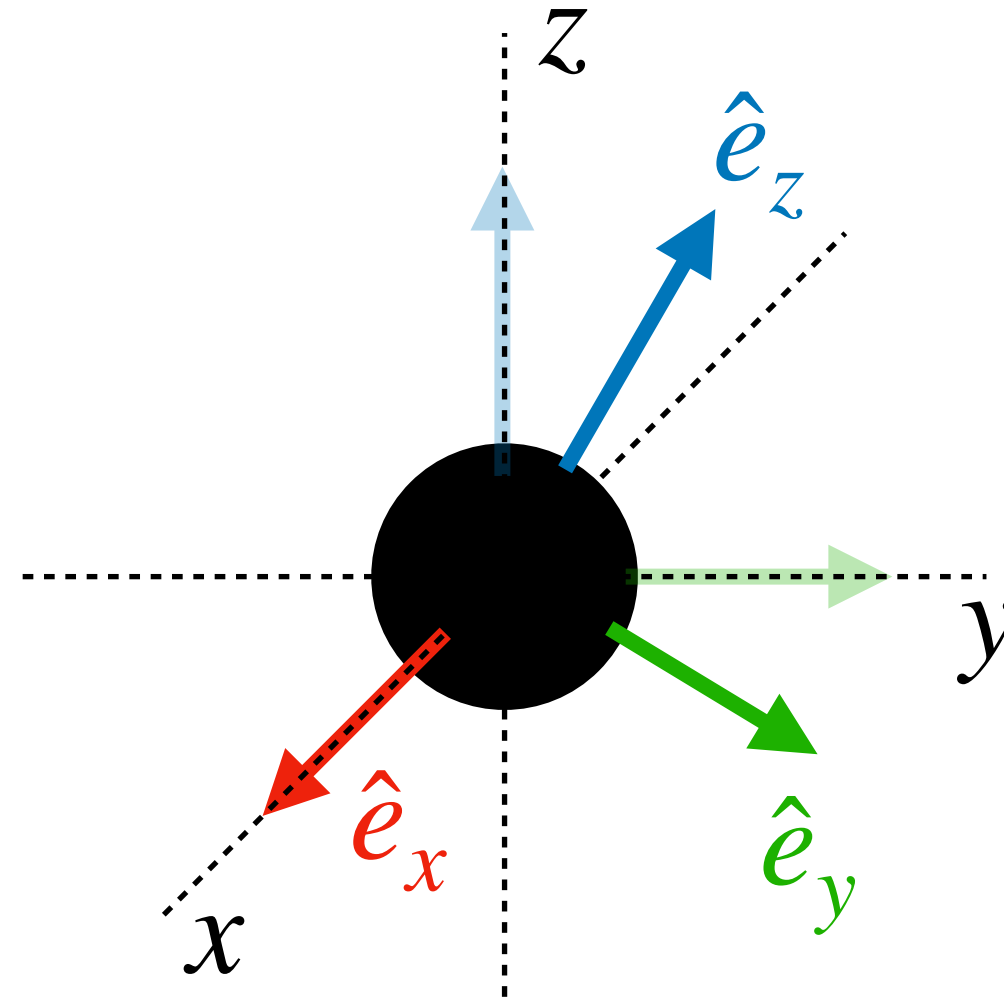
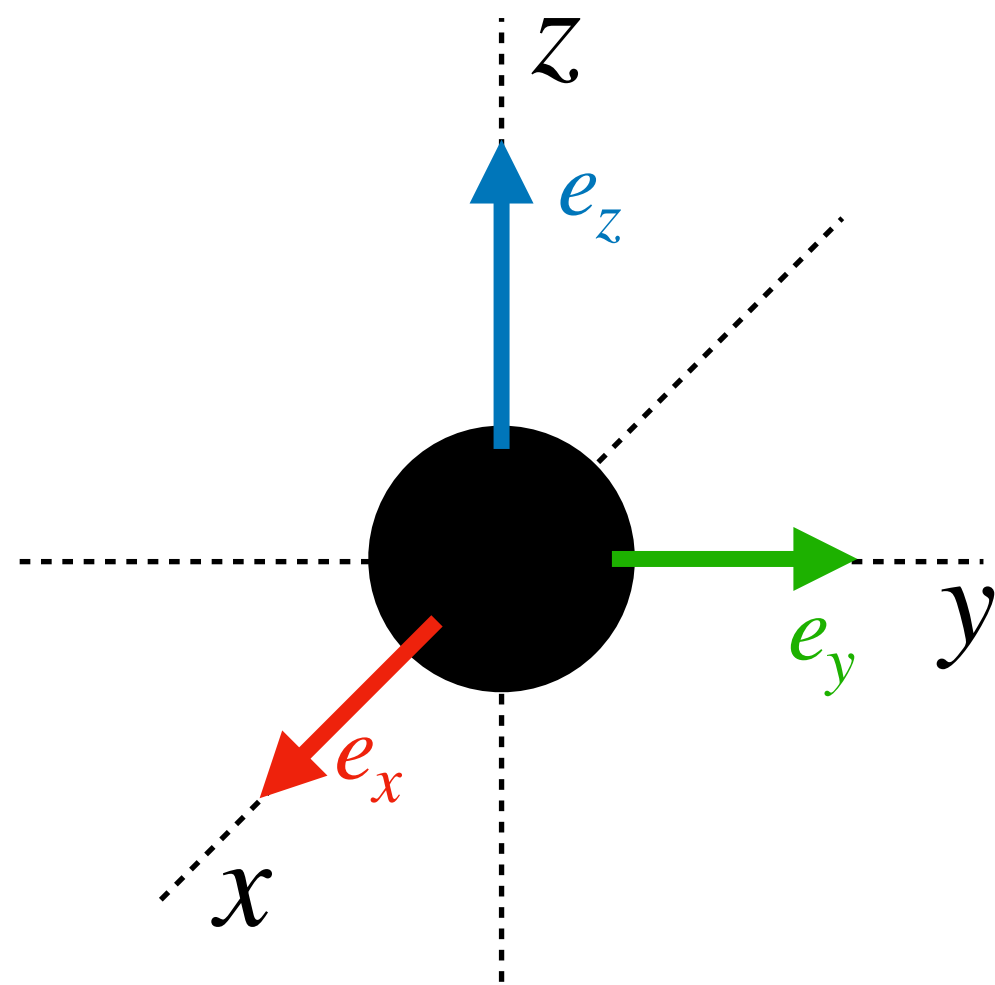
Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

Rotation Representation



$$R = \begin{bmatrix} | & | & | \\ \hat{e}_x & \hat{e}_y & \hat{e}_z \\ | & | & | \end{bmatrix}$$

- We can describe every rotation by a *rotation matrix*
- The set of all possible rotations is given by the *special orthogonal group* $SO(3)$:
 - Set: $\{R \in \mathbb{R}^{3 \times 3} \mid R^T R = \mathbb{I}_3, \det R = 1\}$
 - Associative Operation: Matrix Multiplication $R_a = R_b R_c R_d$
 - Q: Neutral Element: $RE = R$, $E = \mathbb{I}_3$ — “don't rotate”
 - Q: Inverse Element: $IR = \mathbb{I}_3$, $I = R^{-1}$ — “rotate back”

Group (mathematics)

Article [Talk](#)

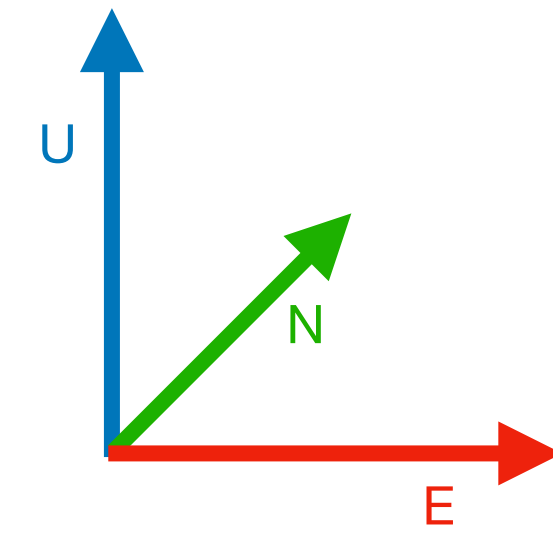
From Wikipedia, the free encyclopedia

This article is about basic notions of groups in mathematics. For a more advanced treatment

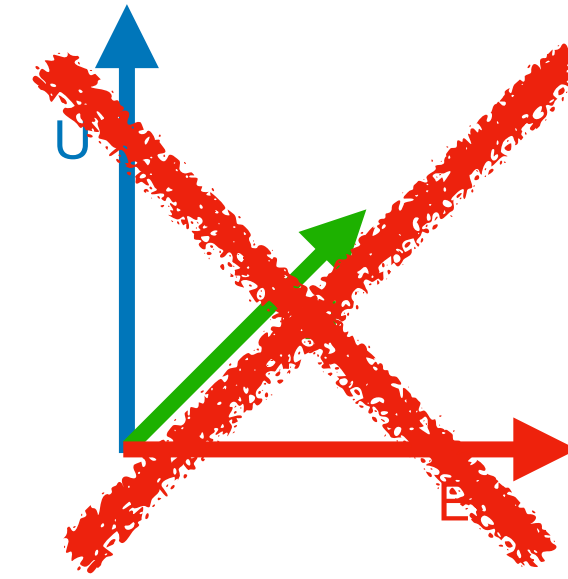
In [mathematics](#), a **group** is [a set](#) with [an operation](#) that combines any two elements of the set to produce a third element within the same set and the following conditions must hold: the operation is [associative](#), it has an [identity element](#), and every element of the set has an [inverse element](#).

World & Body Frame

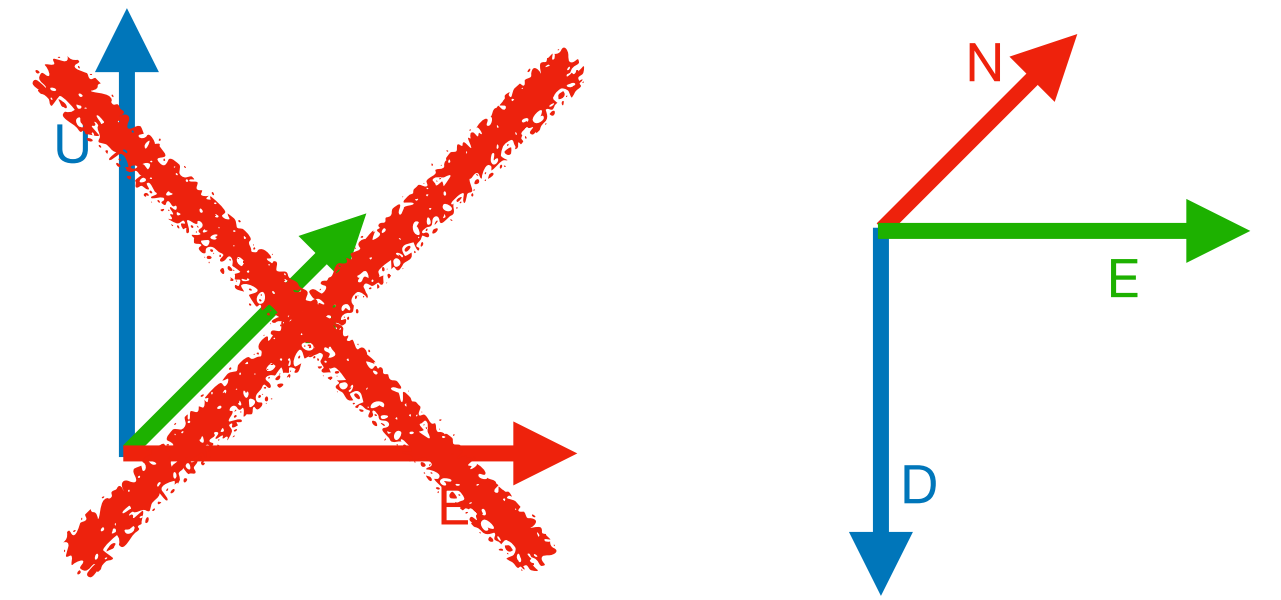
World & Body Frame



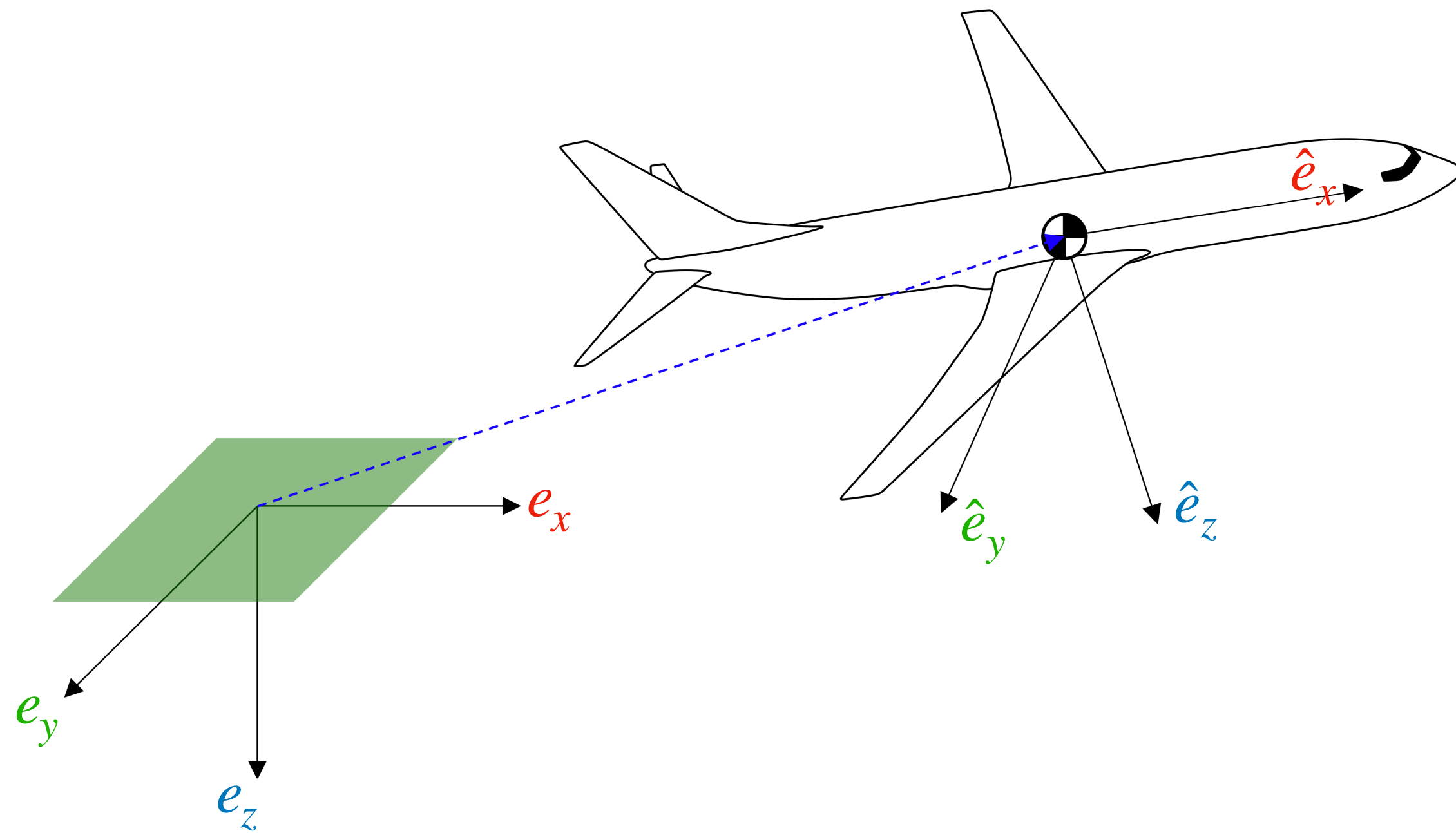
World & Body Frame



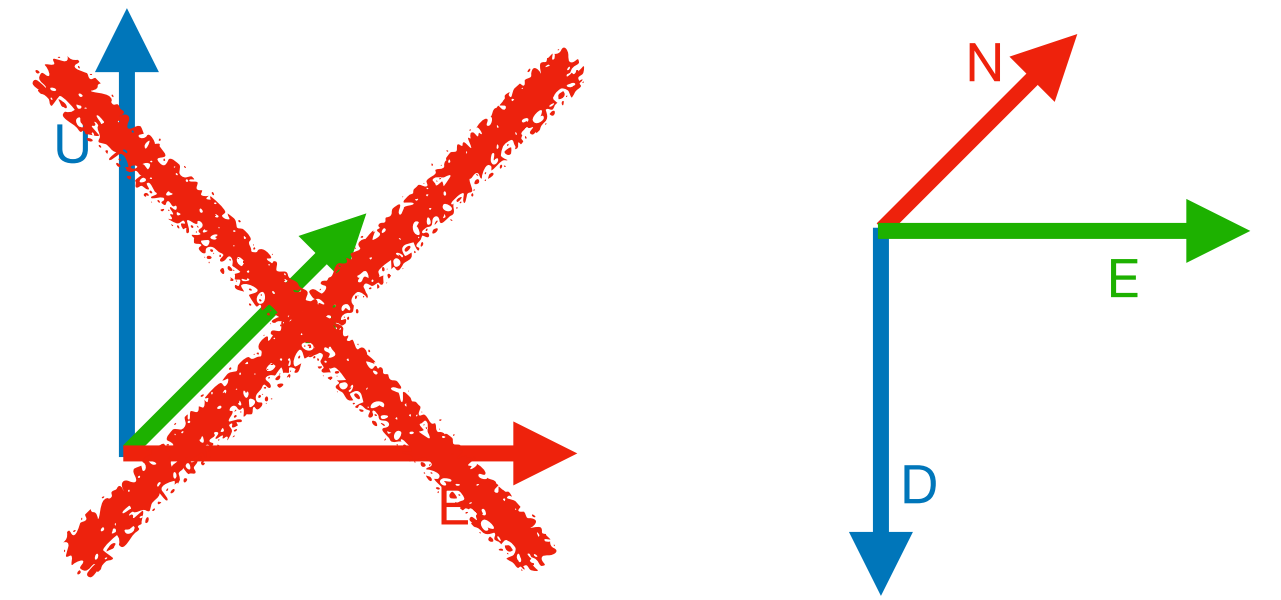
World & Body Frame



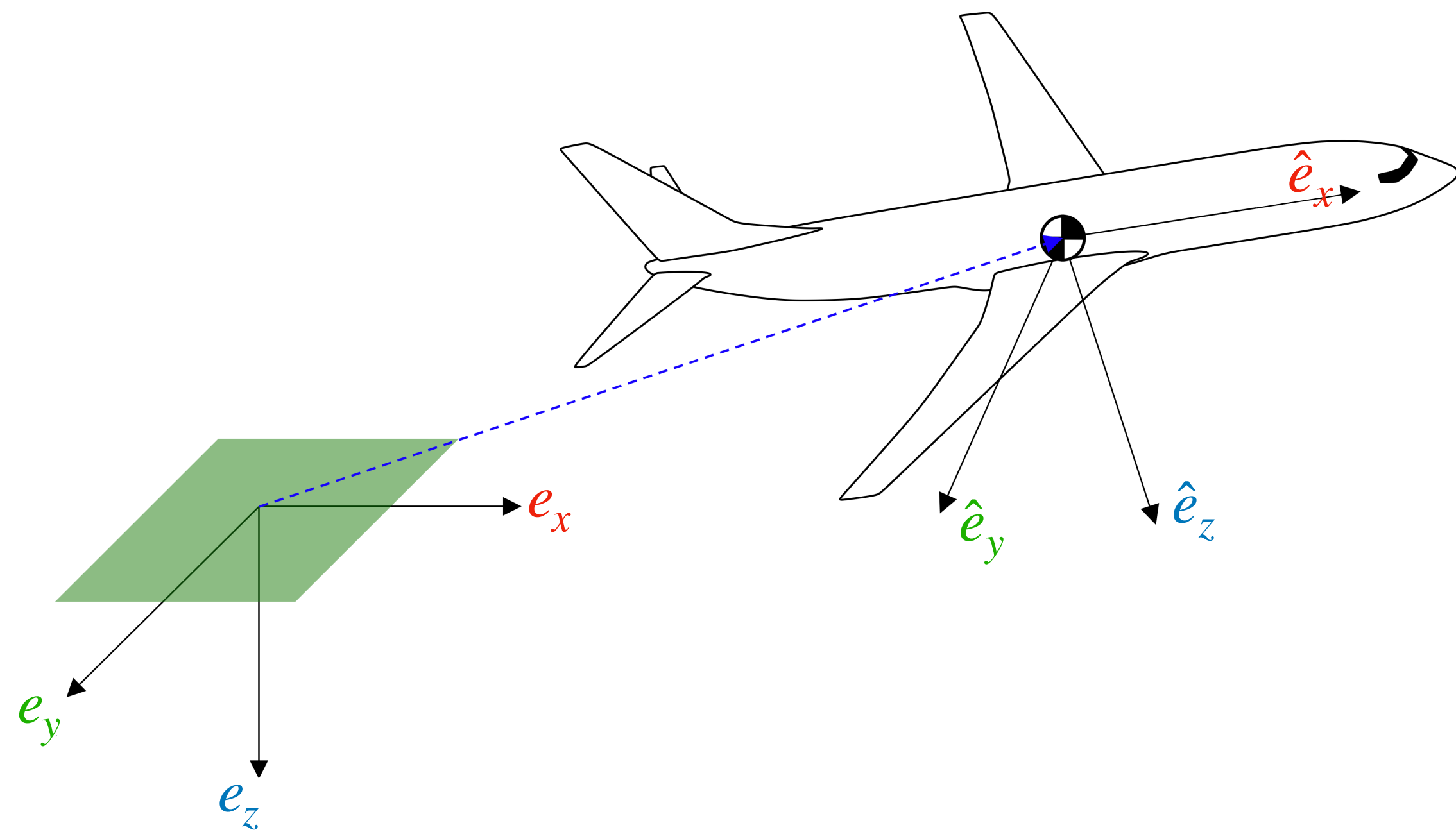
World & Body Frame



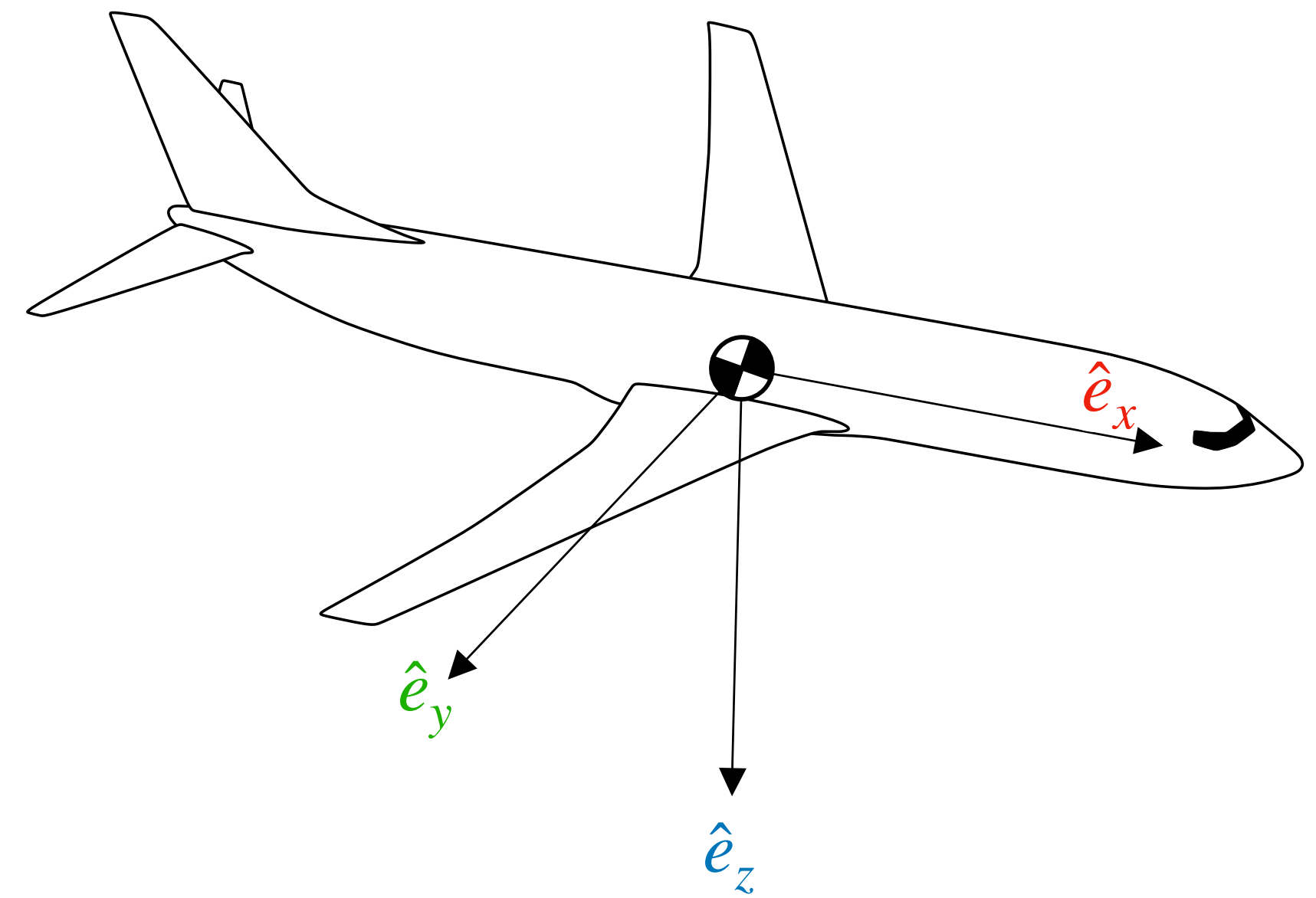
NED World Frame



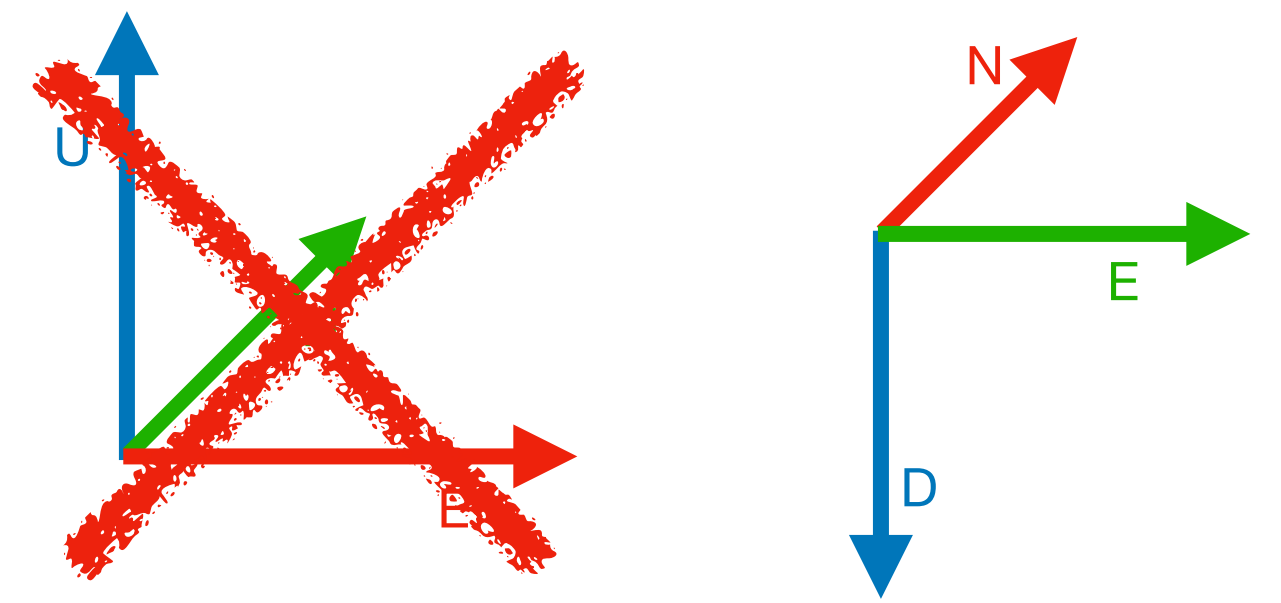
World & Body Frame



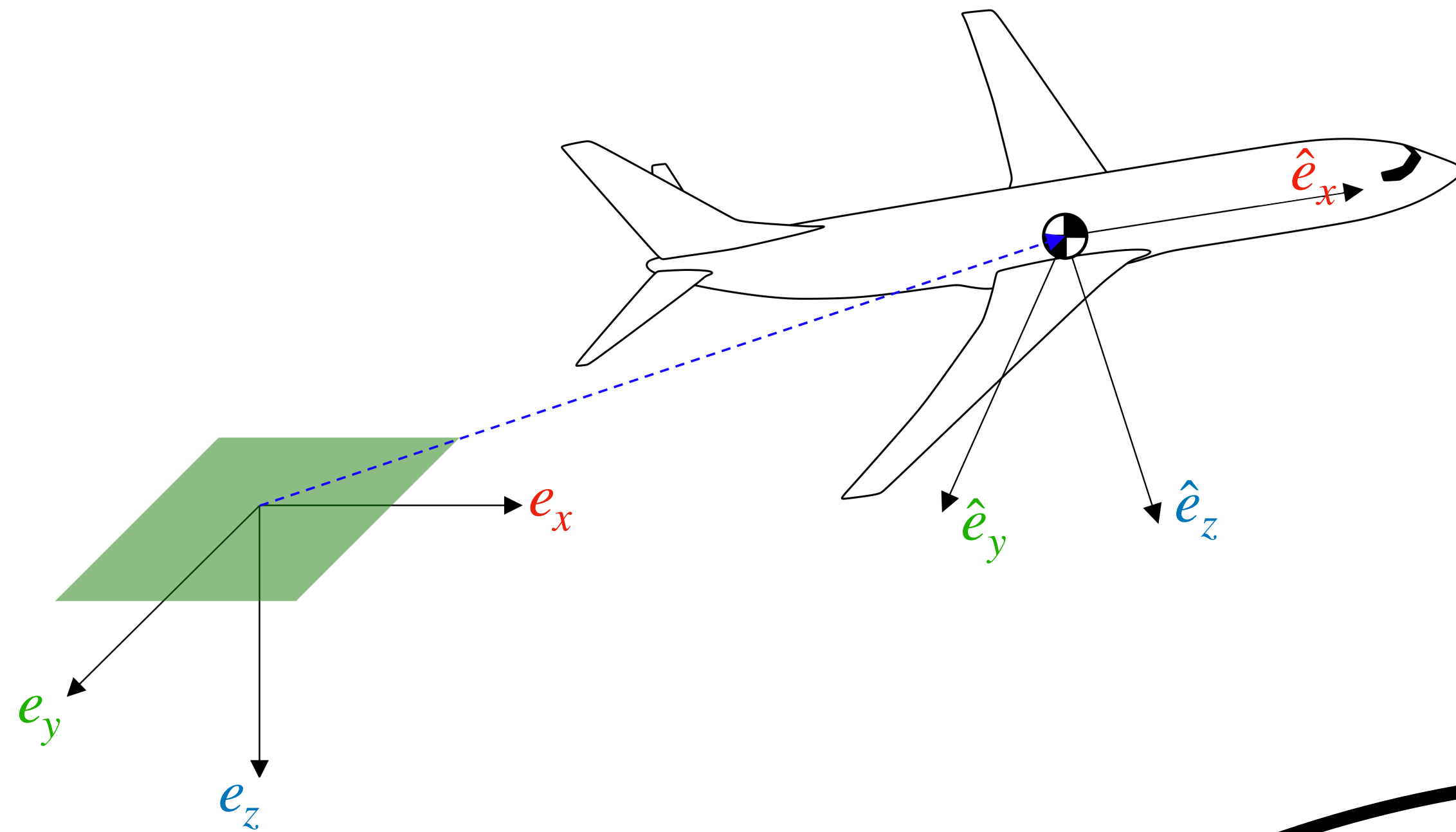
NED World Frame



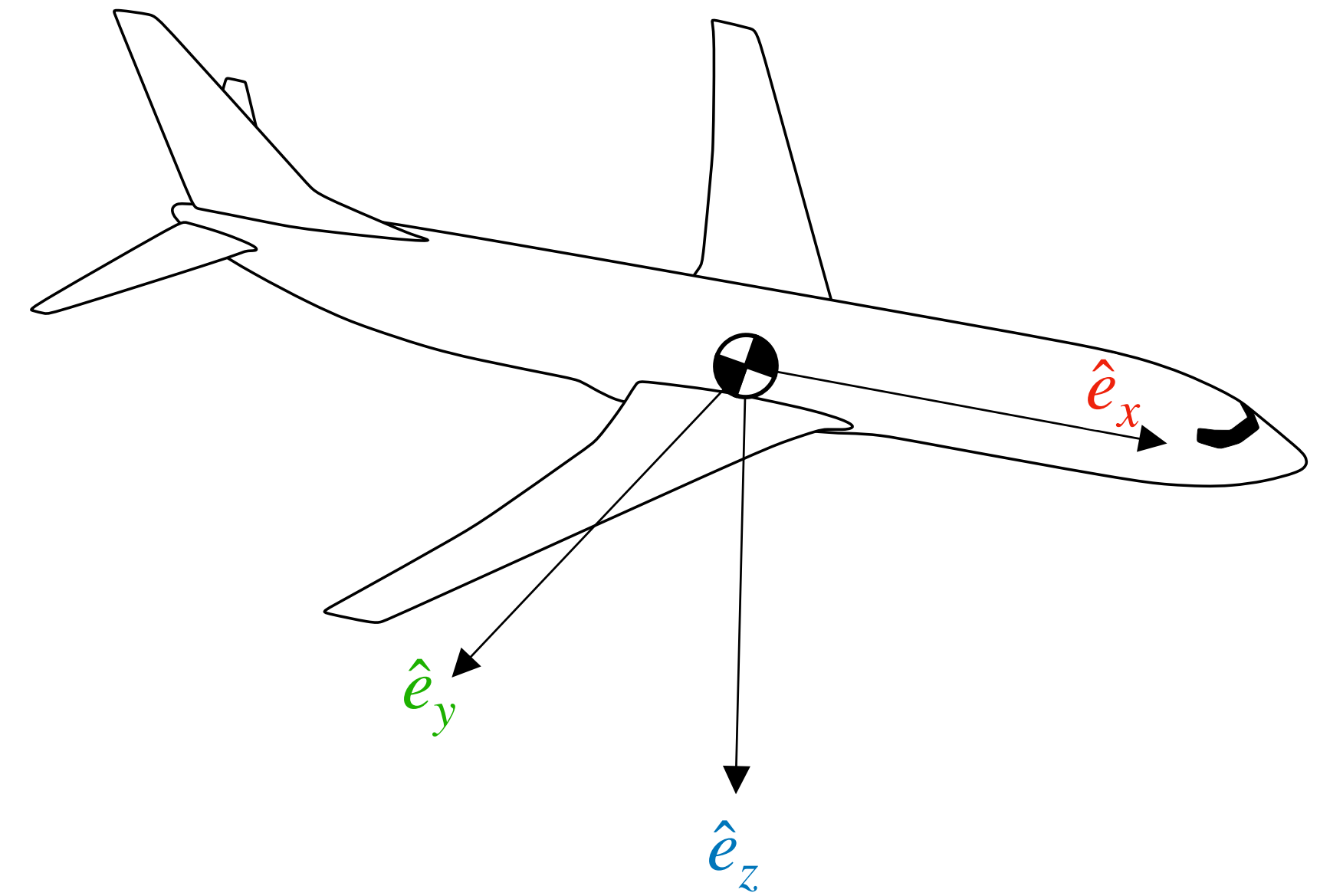
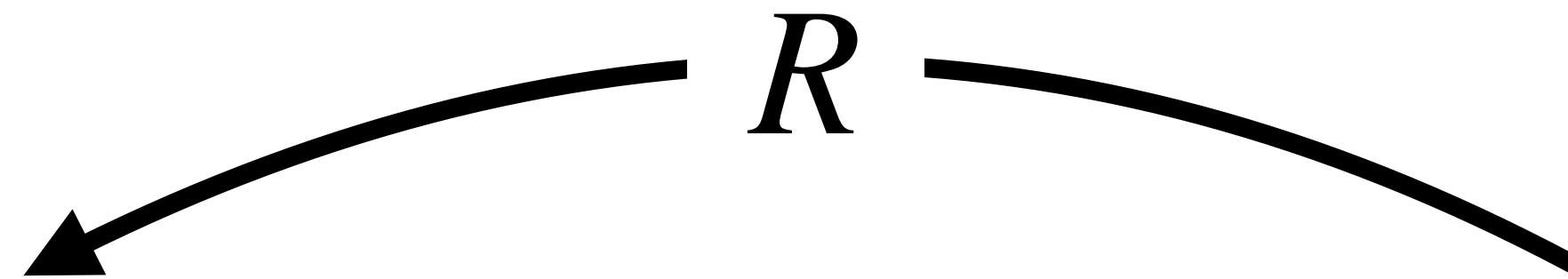
NED Body Frame



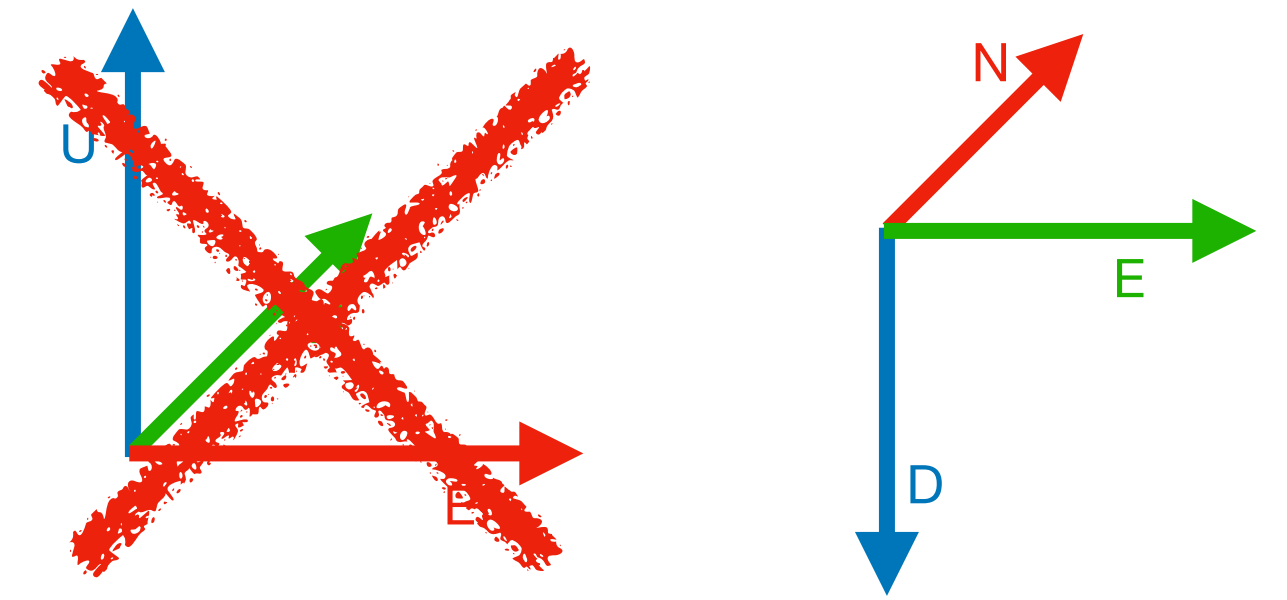
World & Body Frame



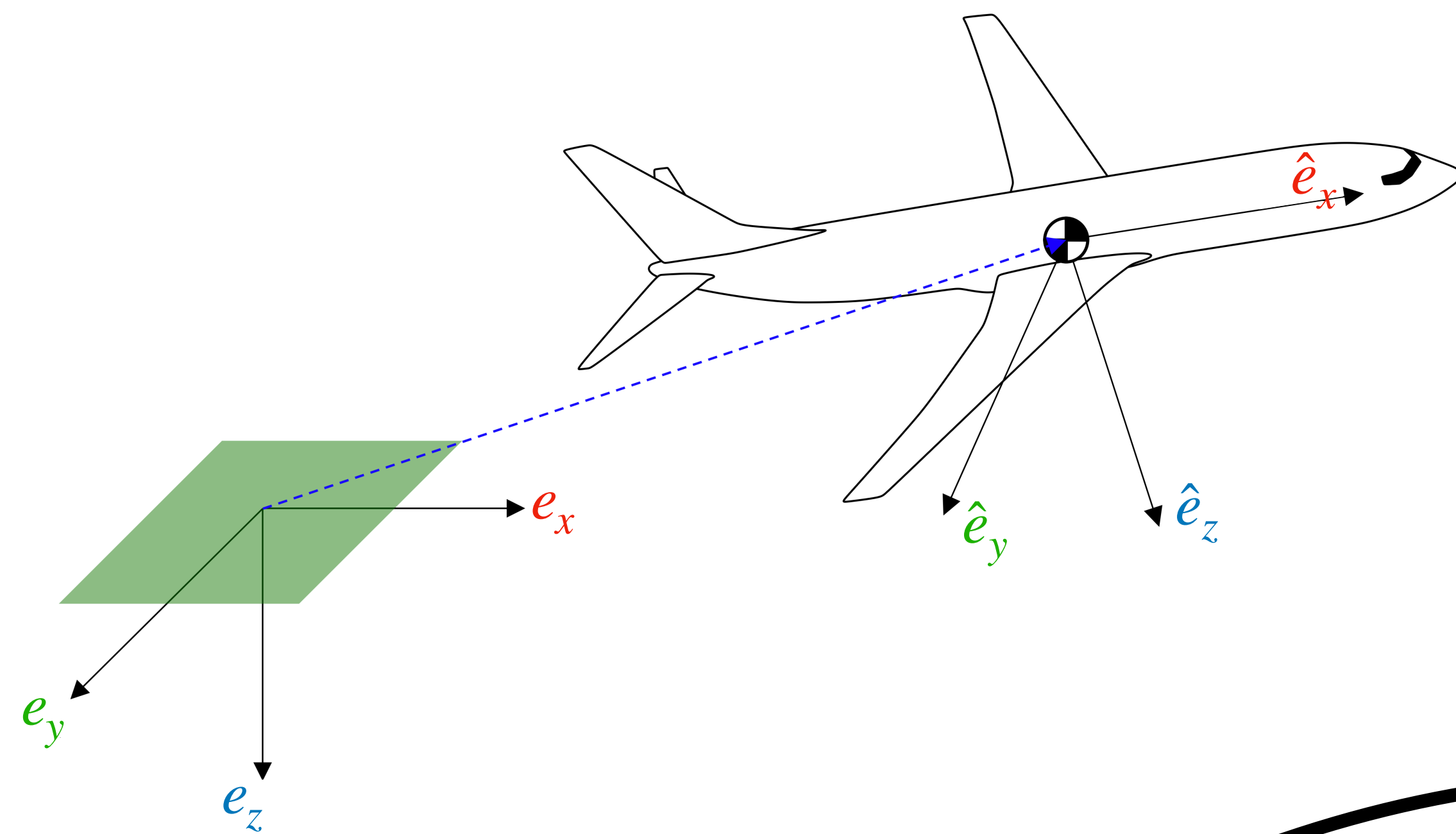
NED World Frame



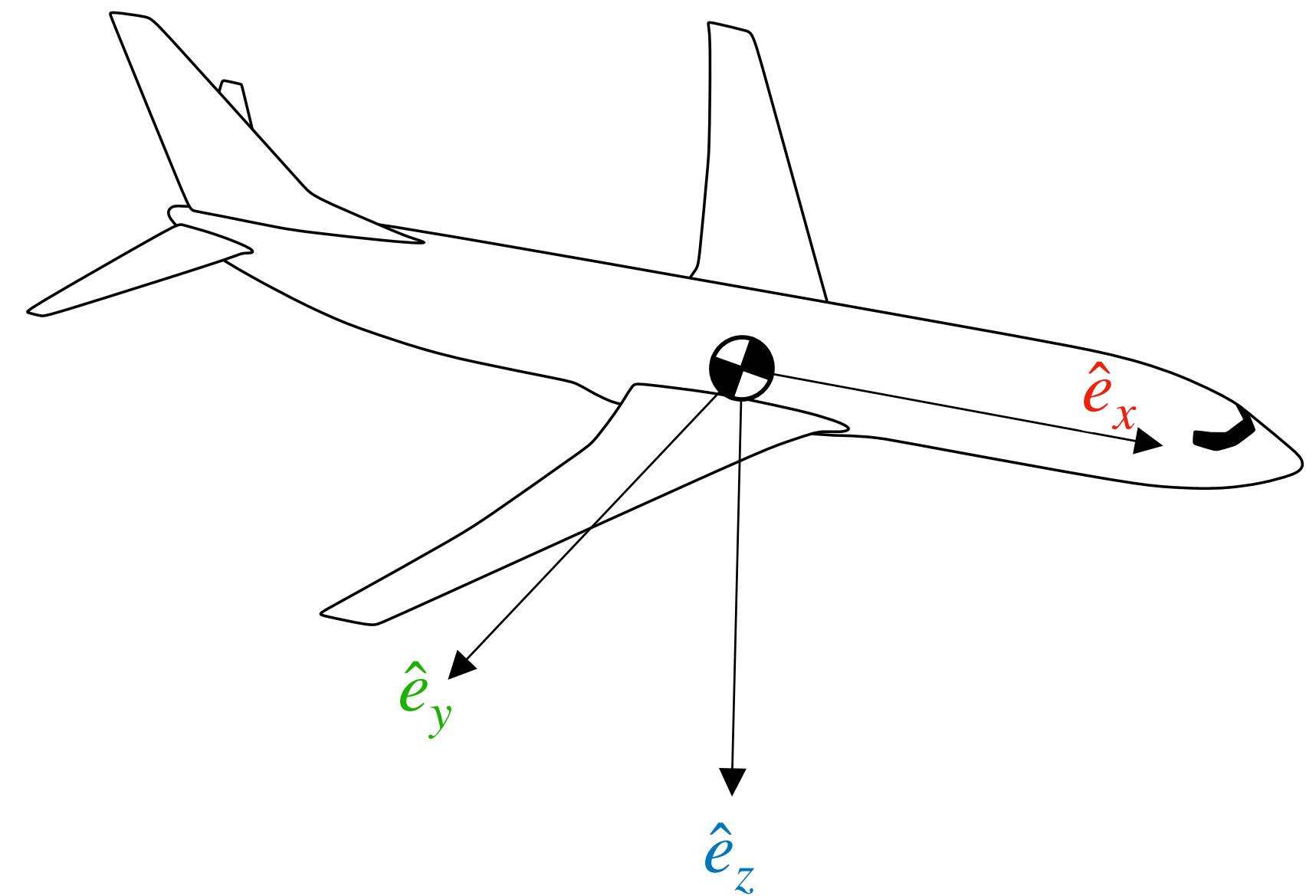
NED Body Frame



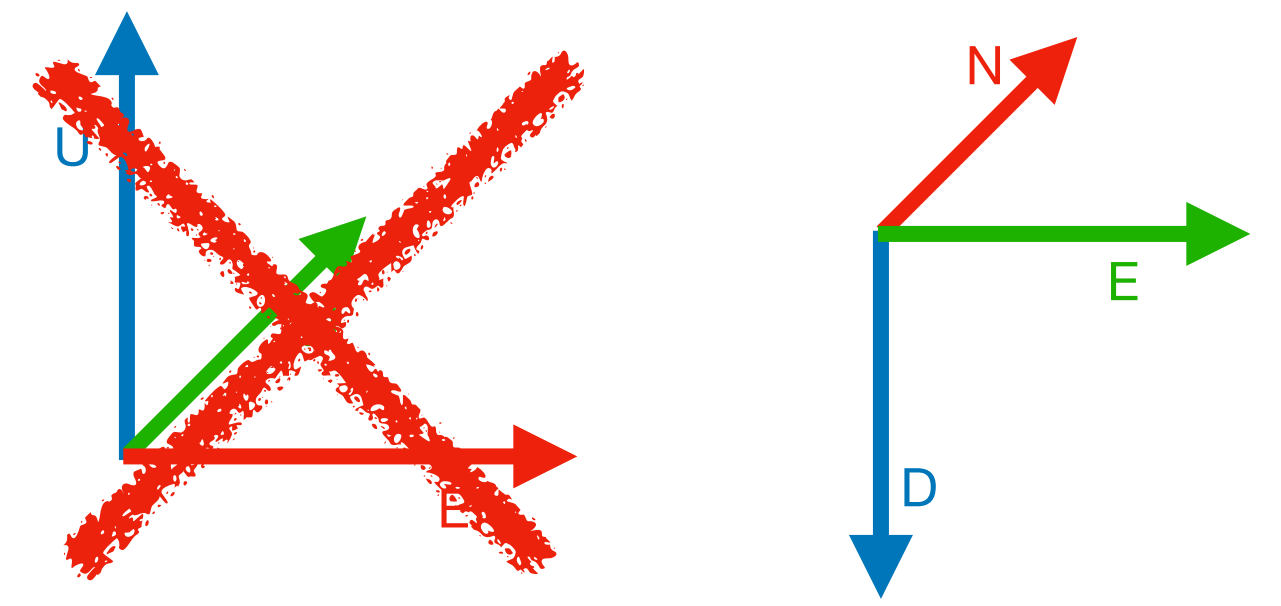
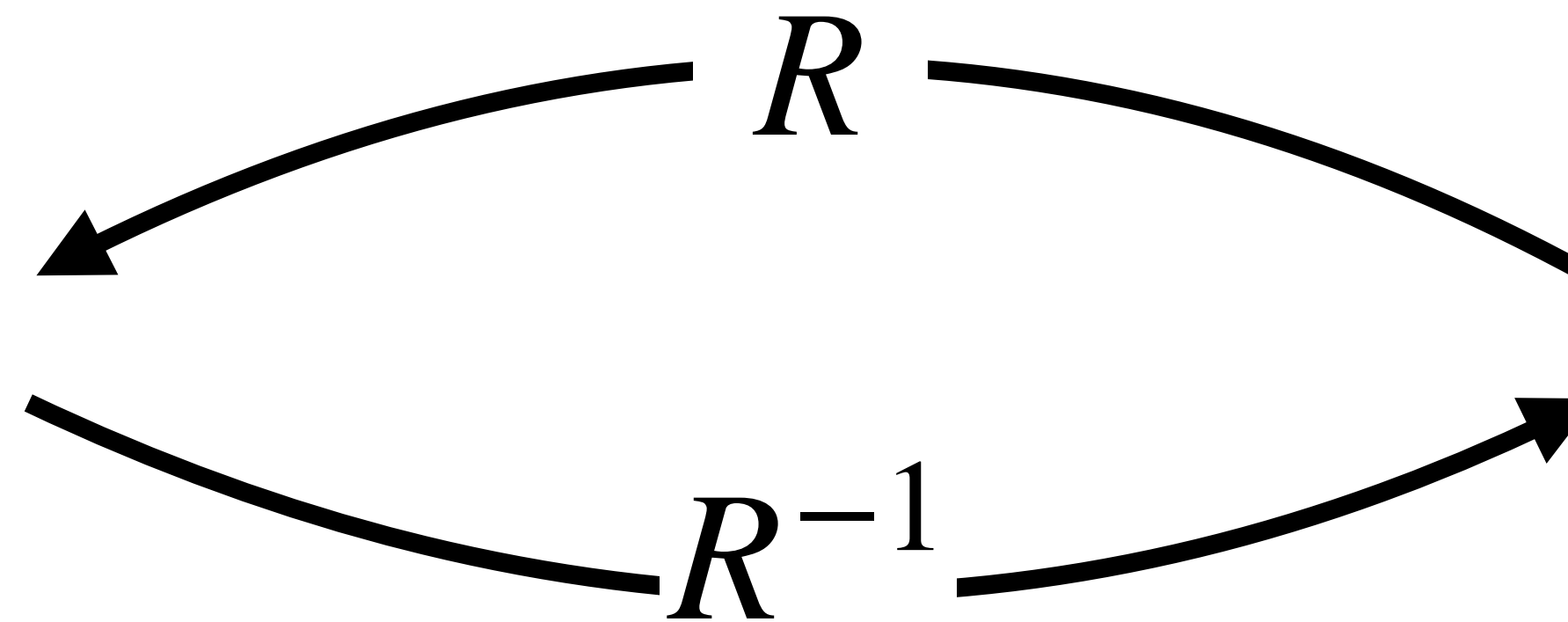
World & Body Frame



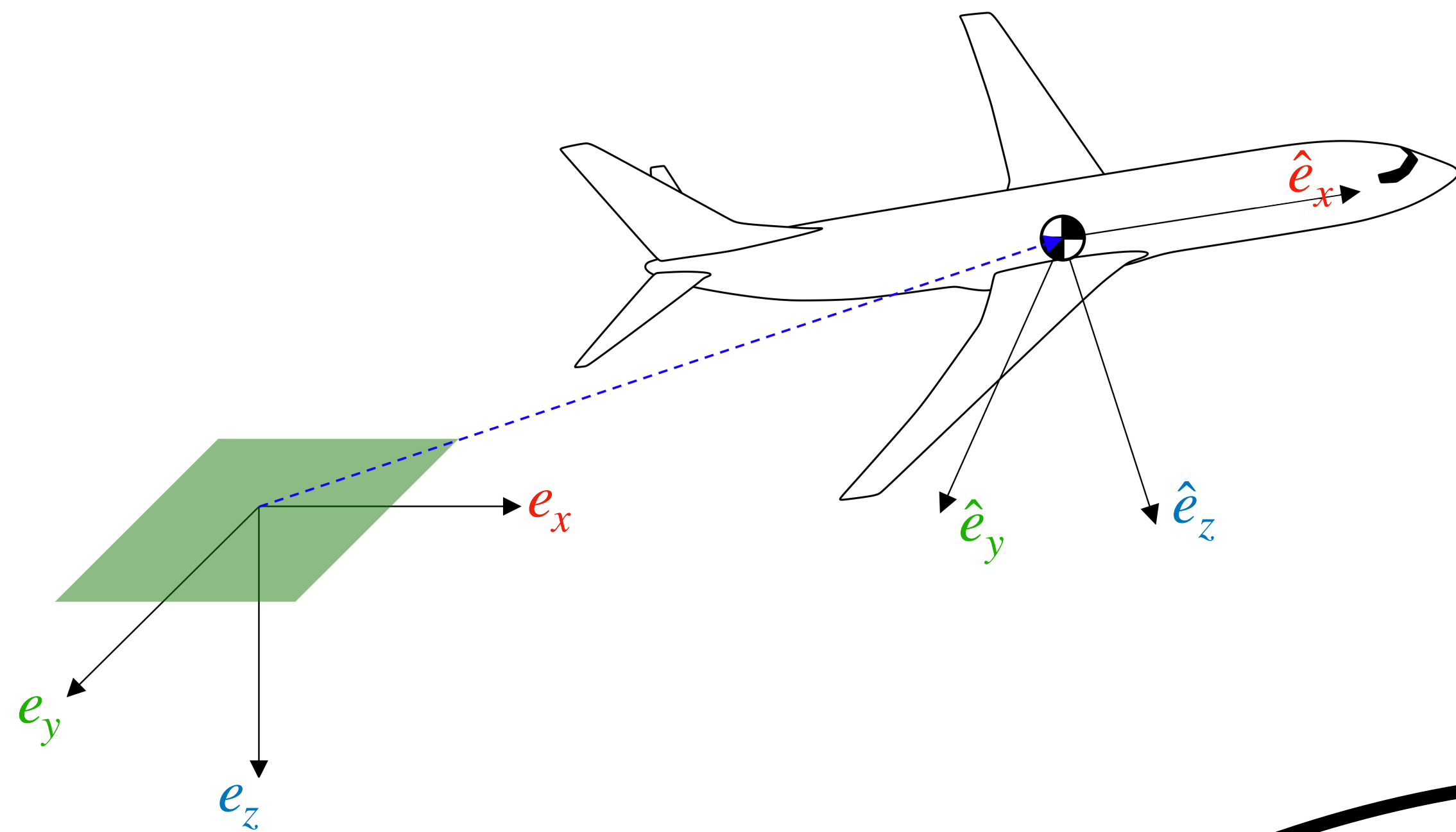
NED World Frame



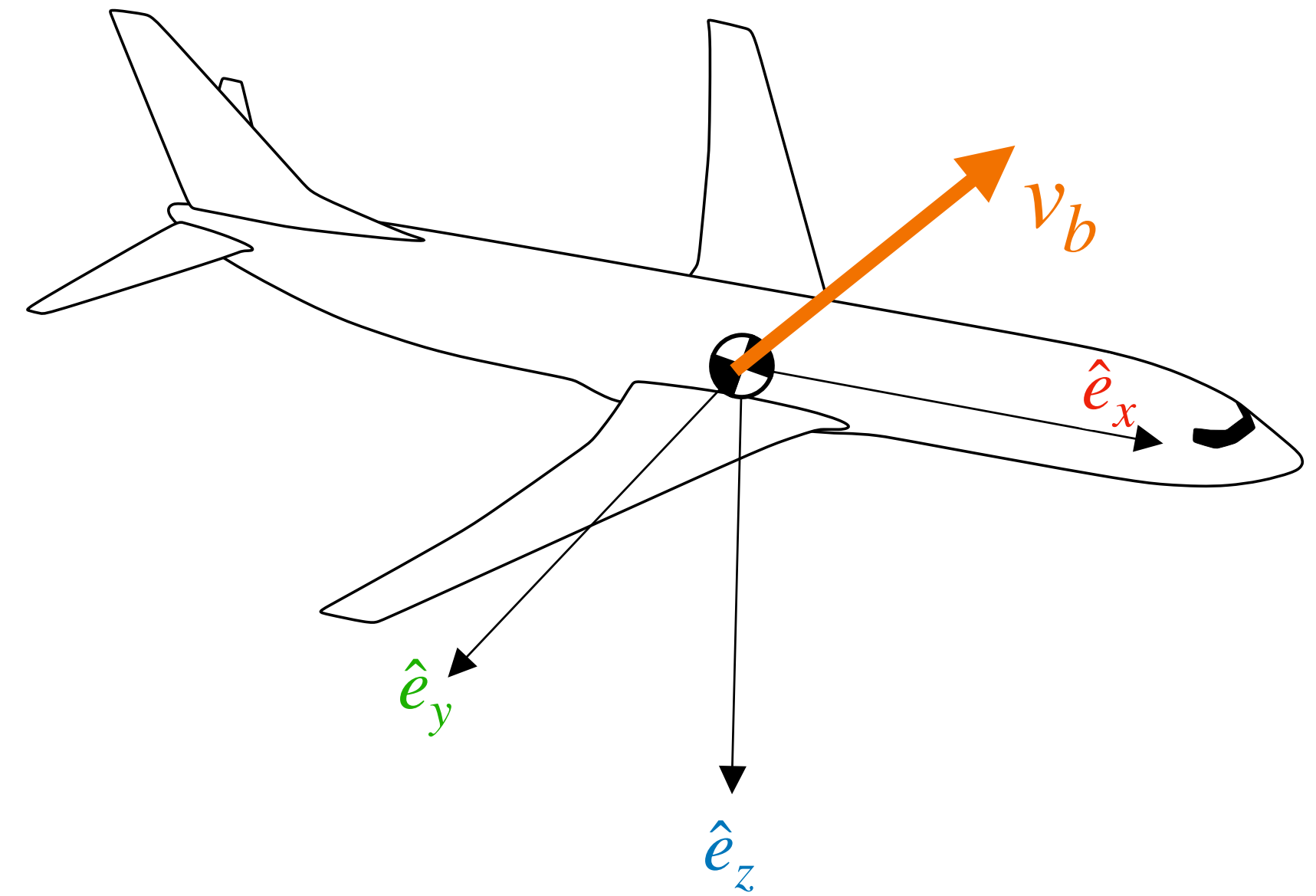
NED Body Frame



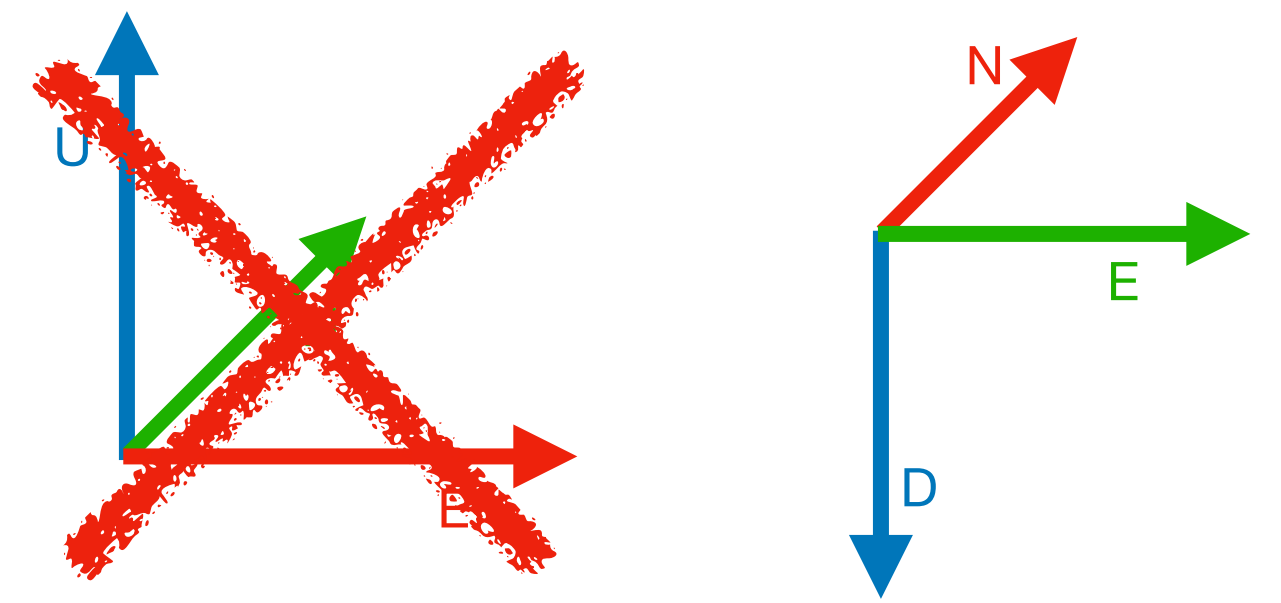
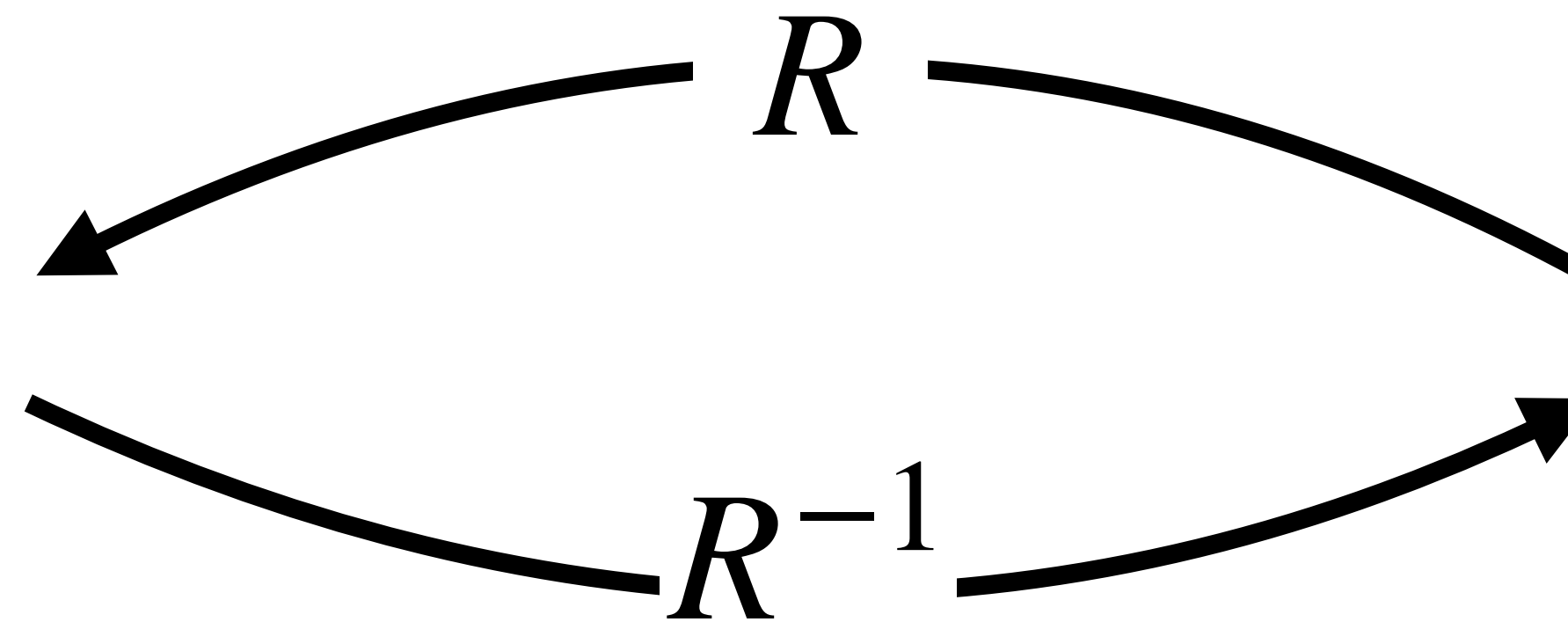
World & Body Frame



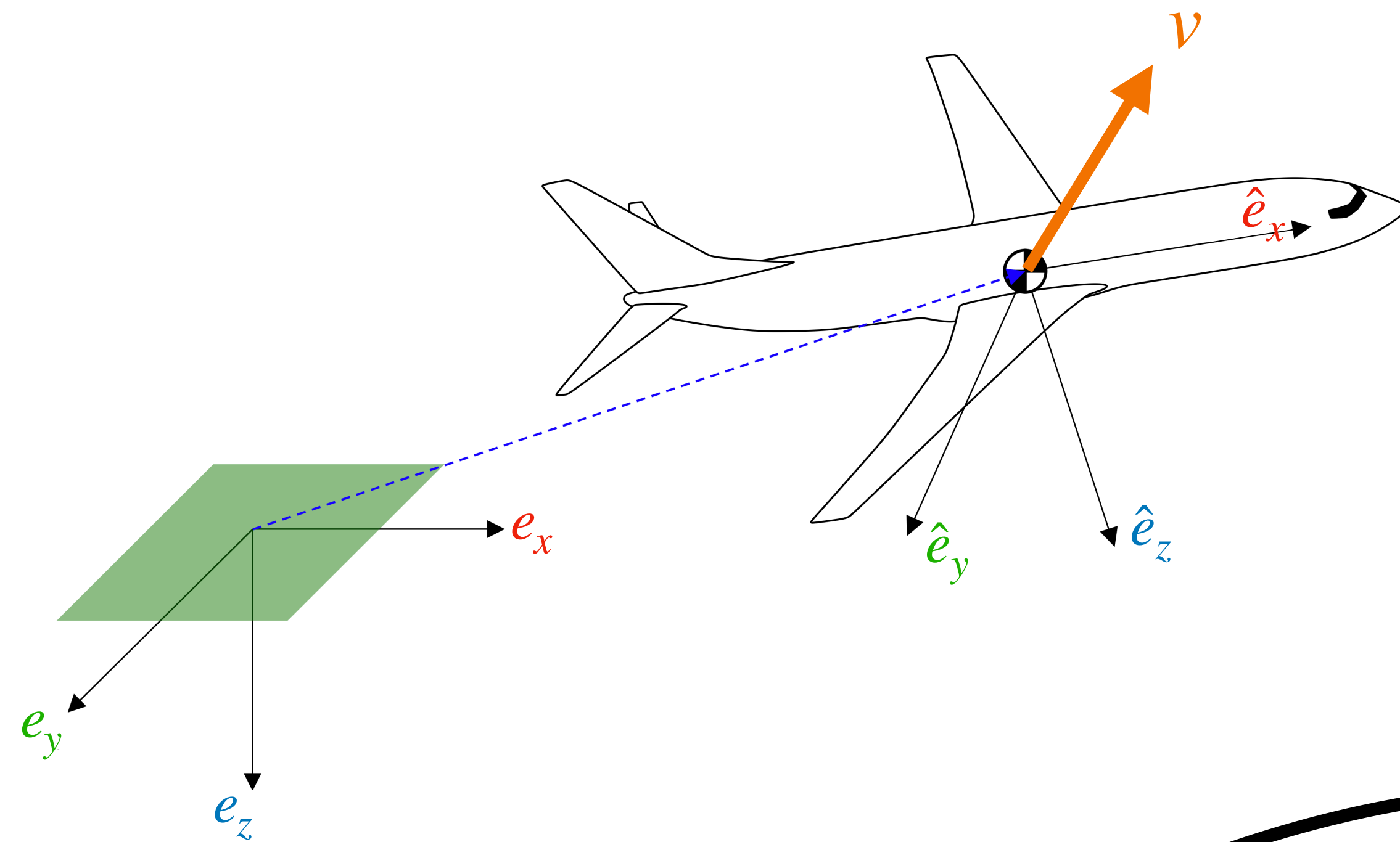
NED World Frame



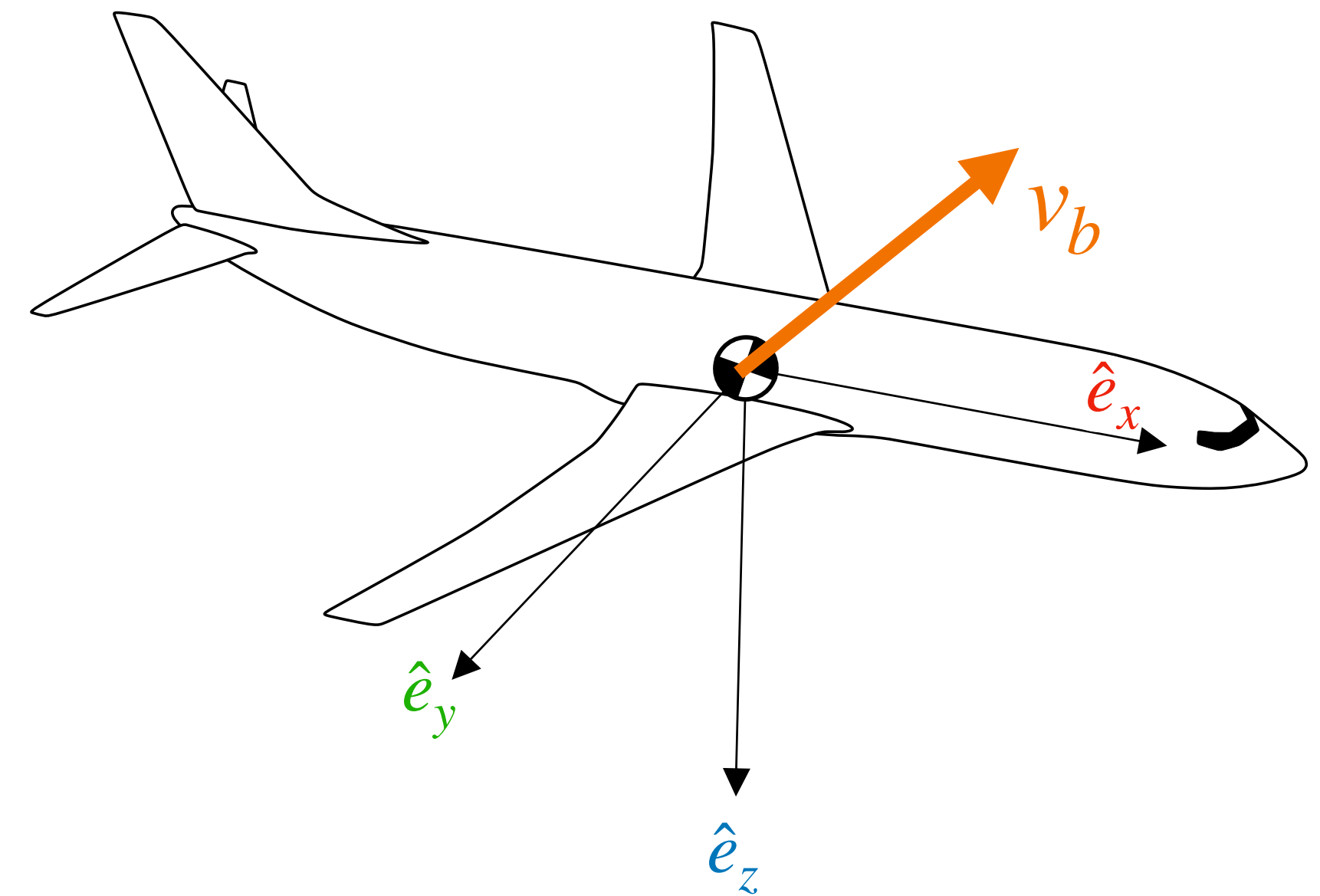
NED Body Frame



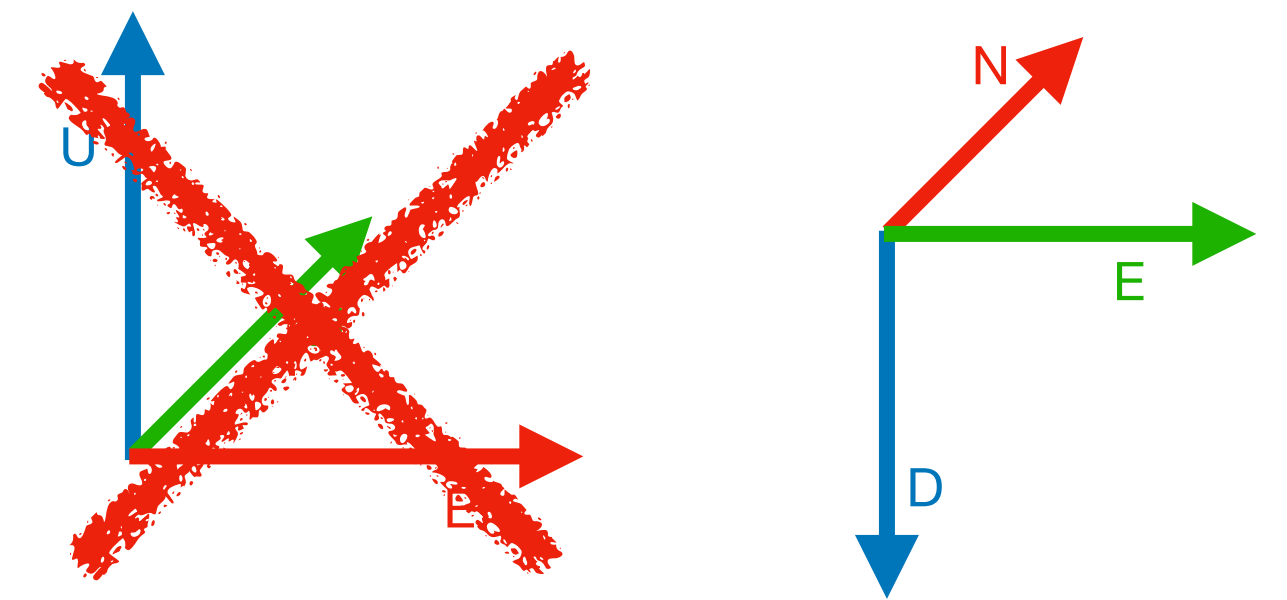
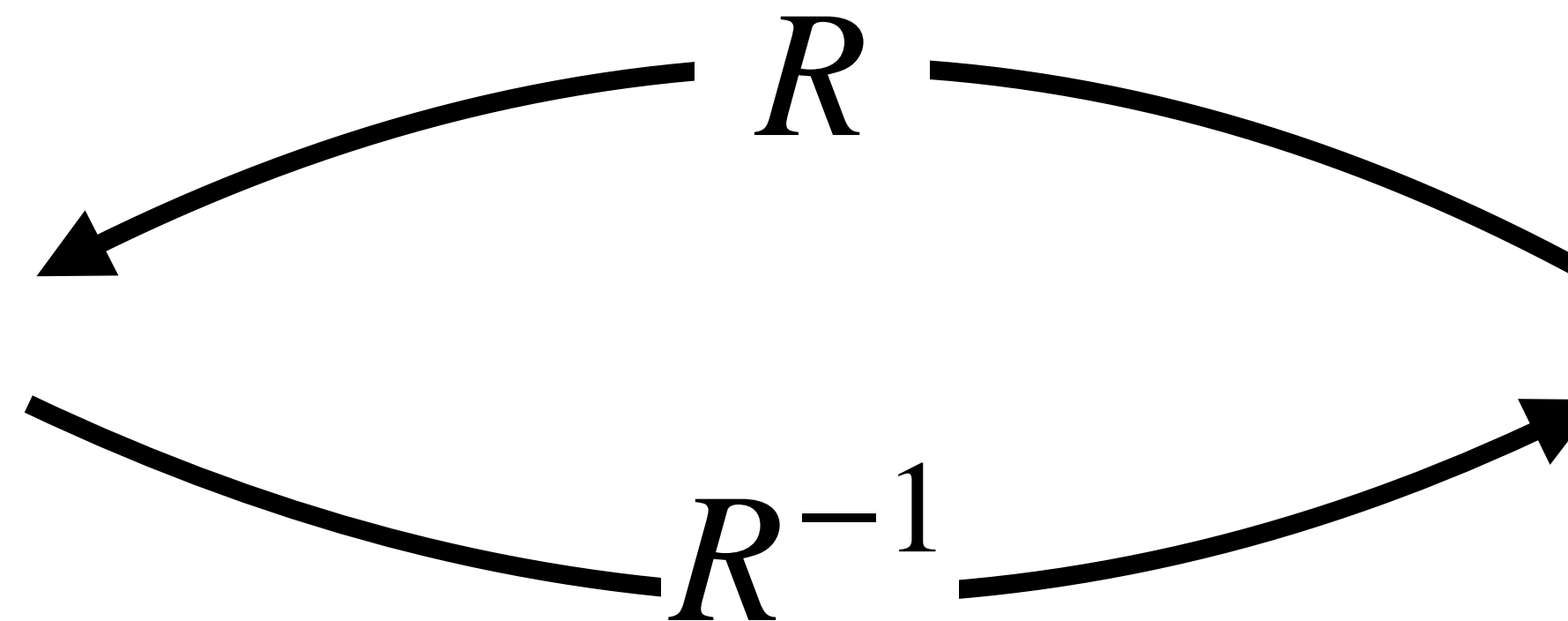
World & Body Frame



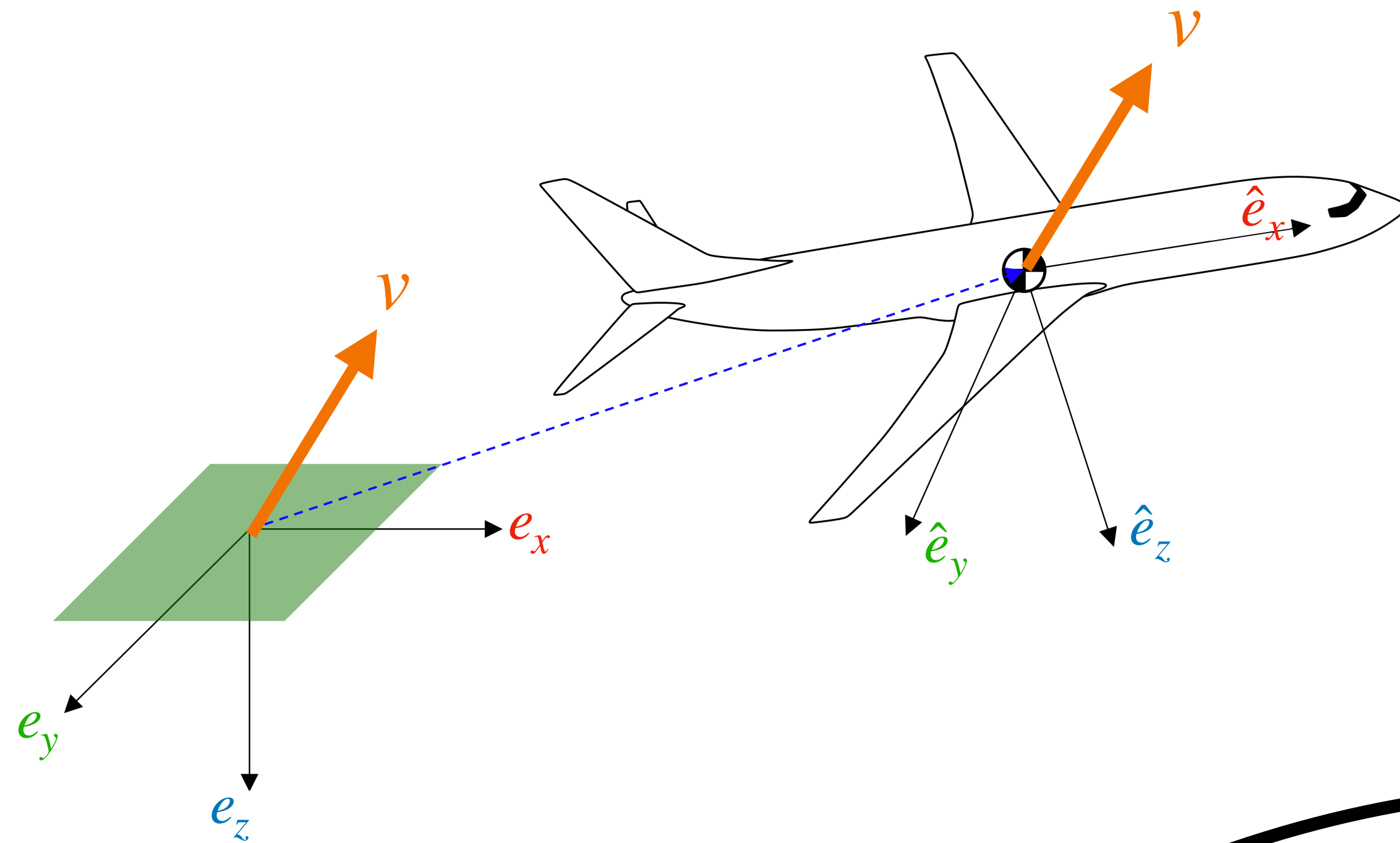
NED World Frame



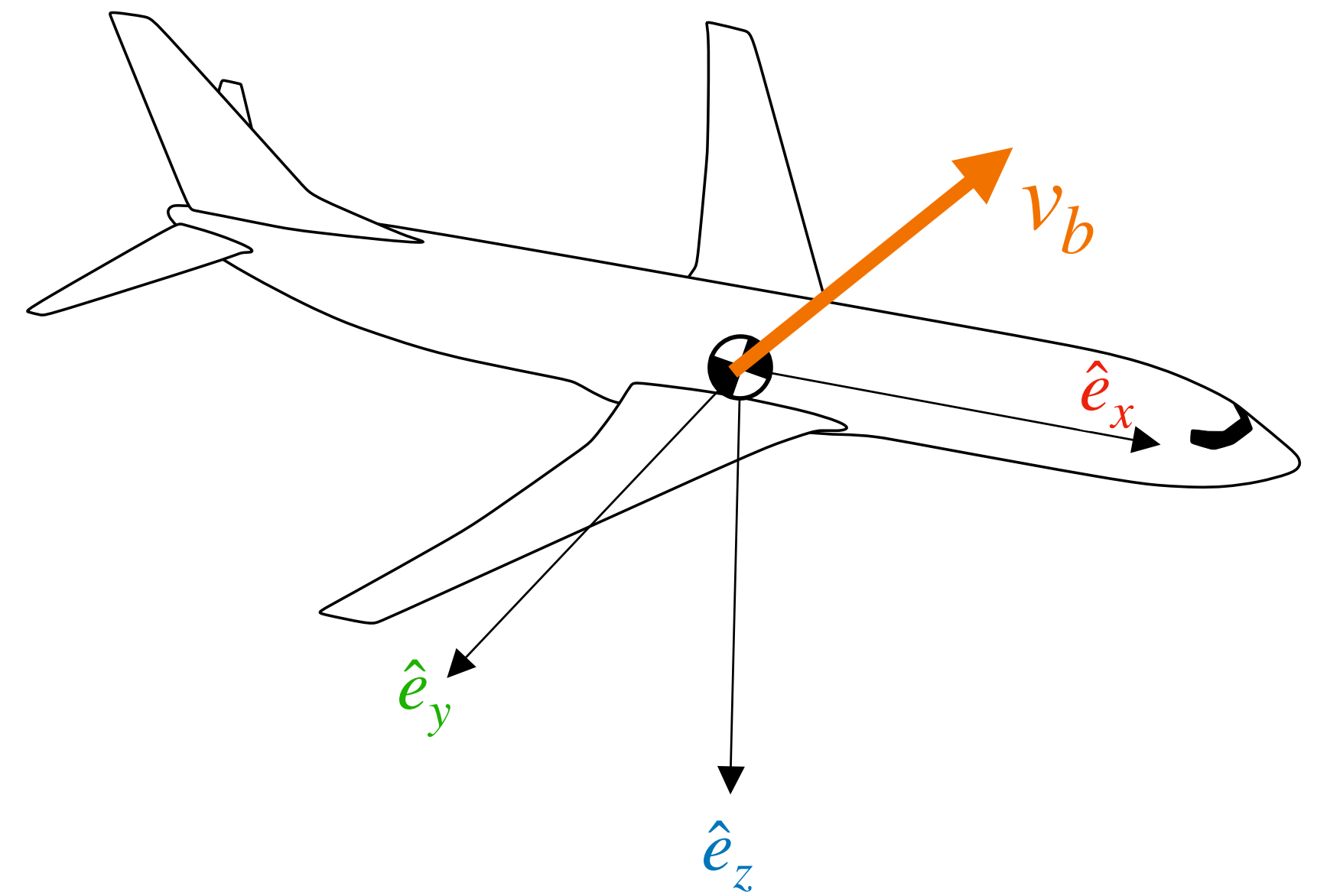
NED Body Frame



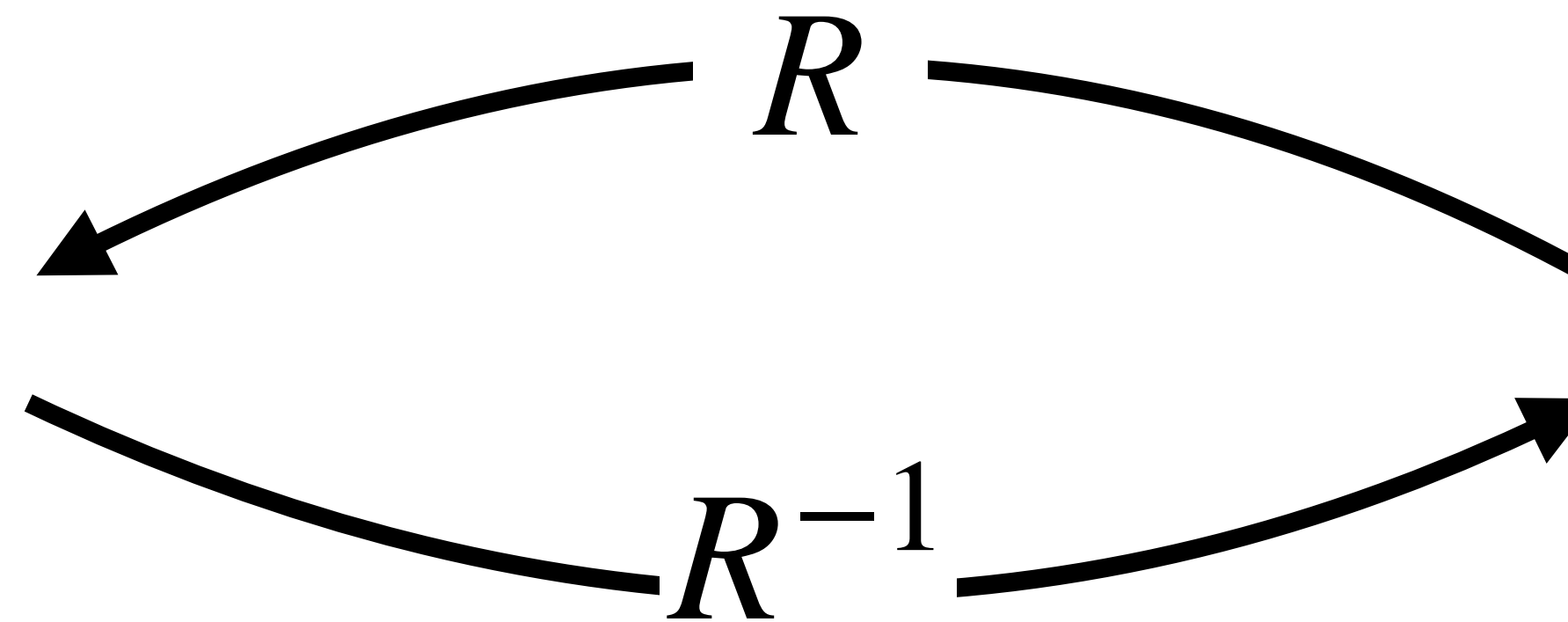
World & Body Frame



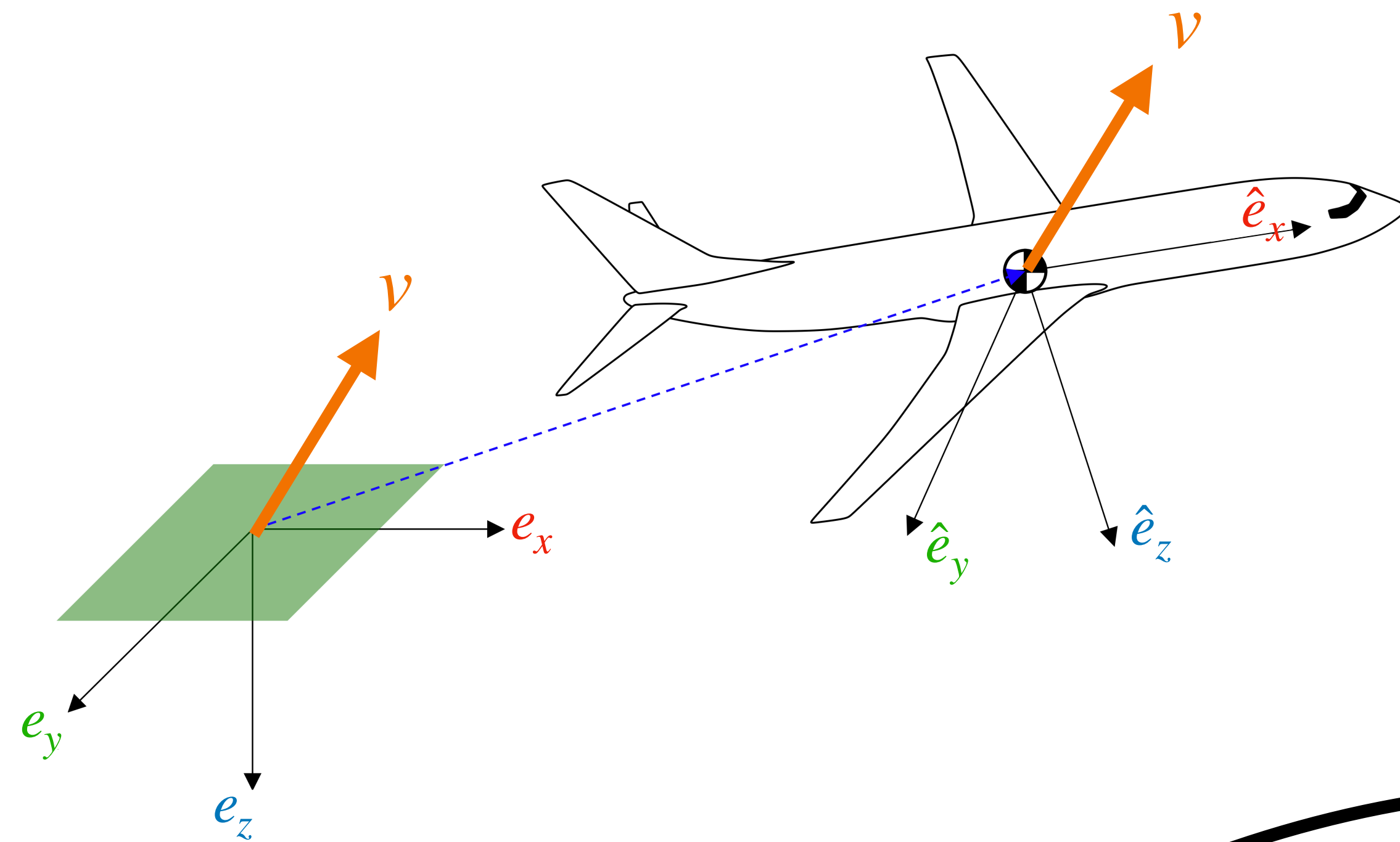
NED World Frame



NED Body Frame

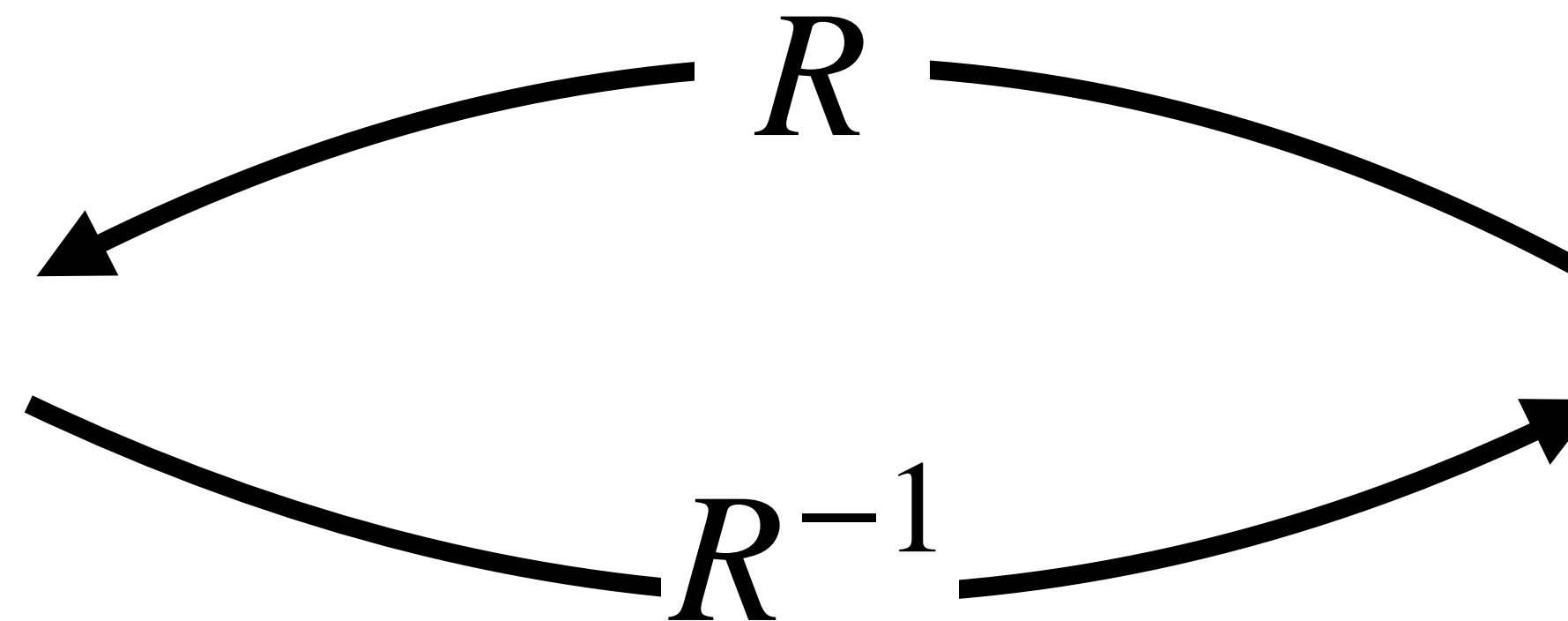


World & Body Frame

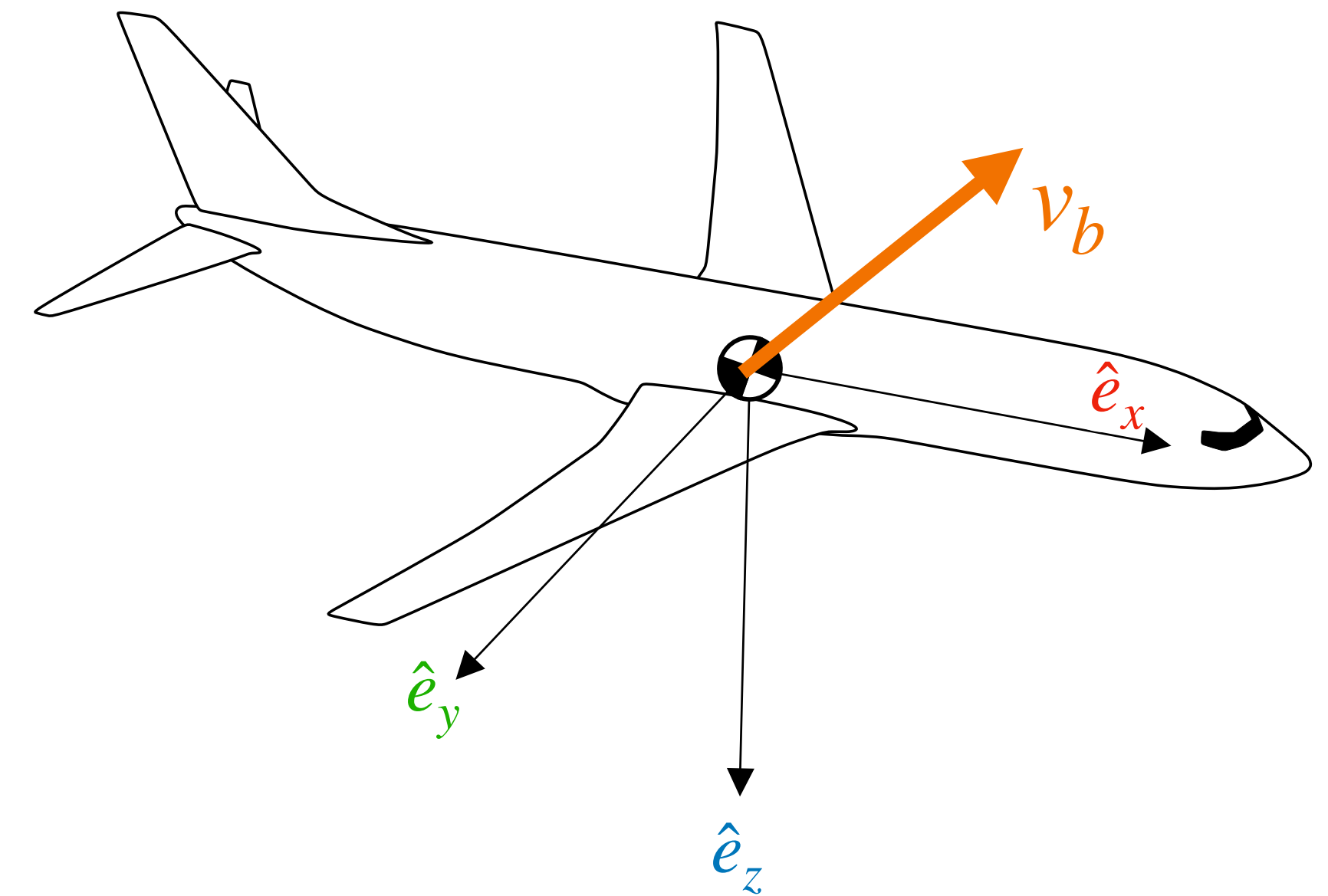
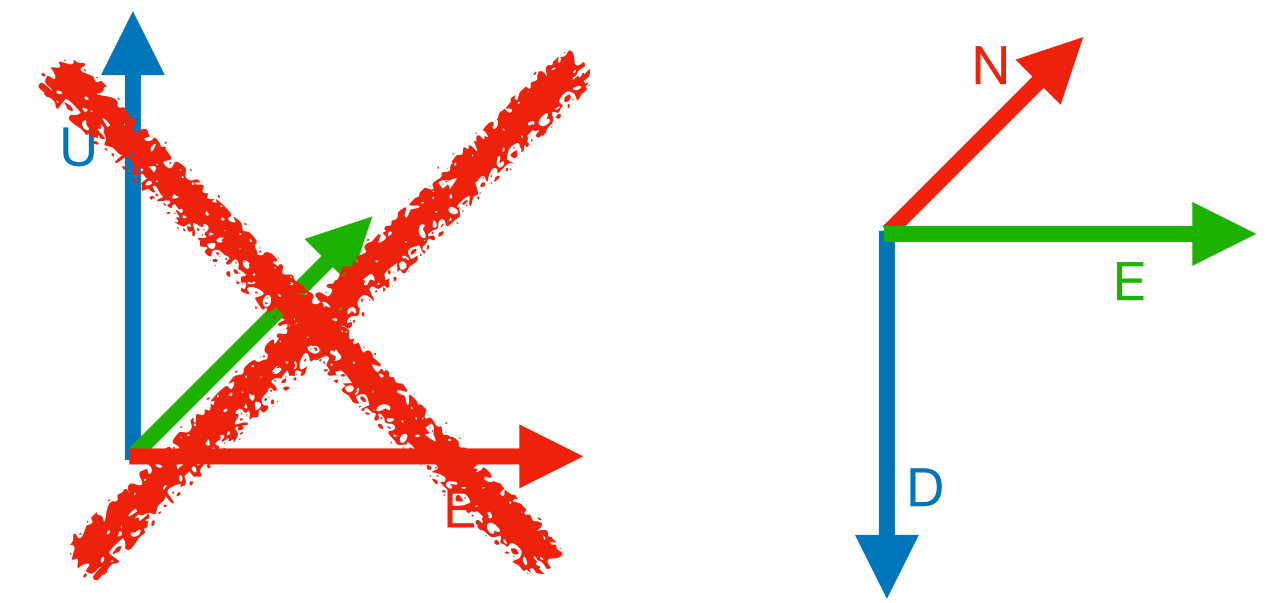


NED World Frame

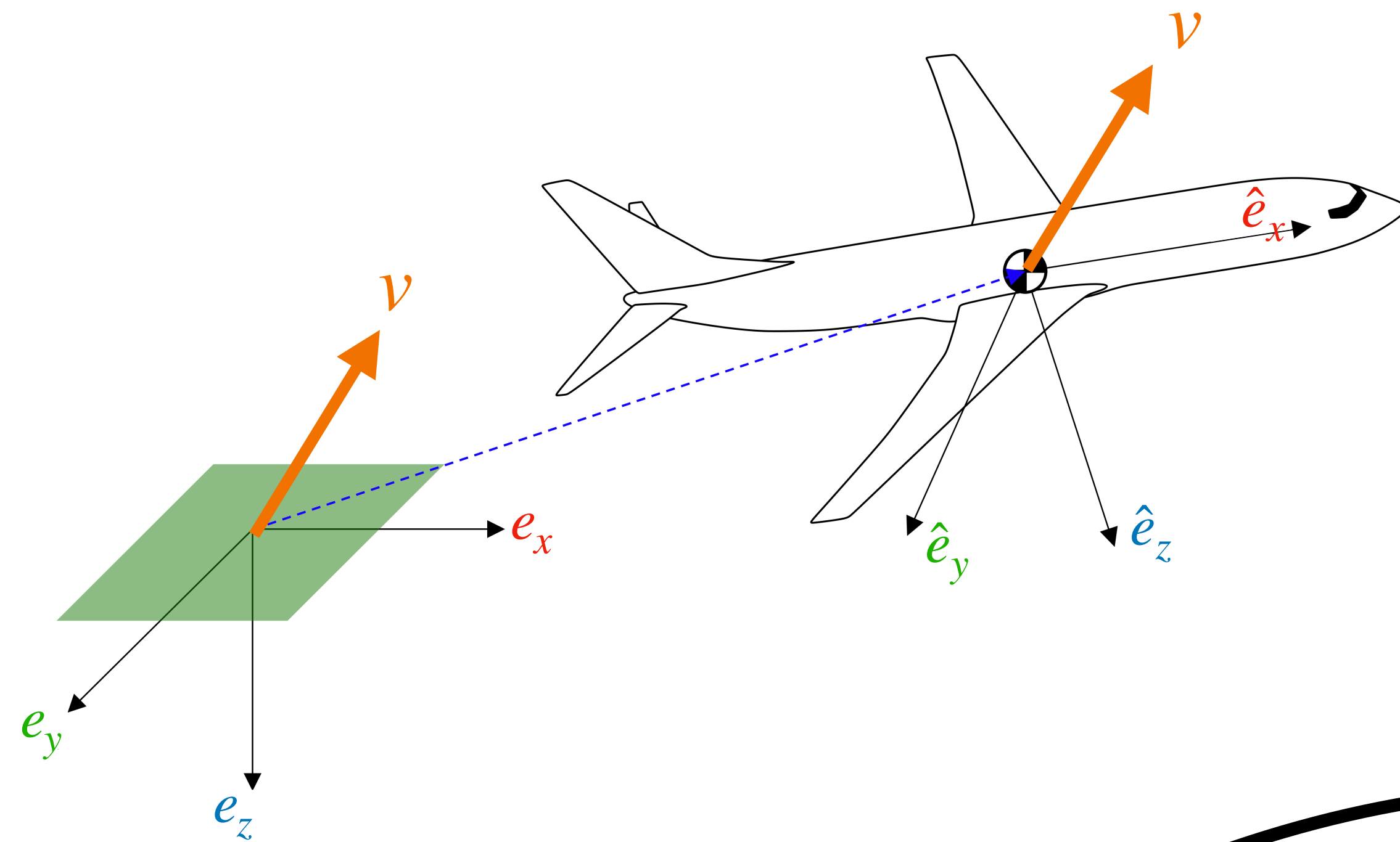
$$v = Rv_b$$



NED Body Frame

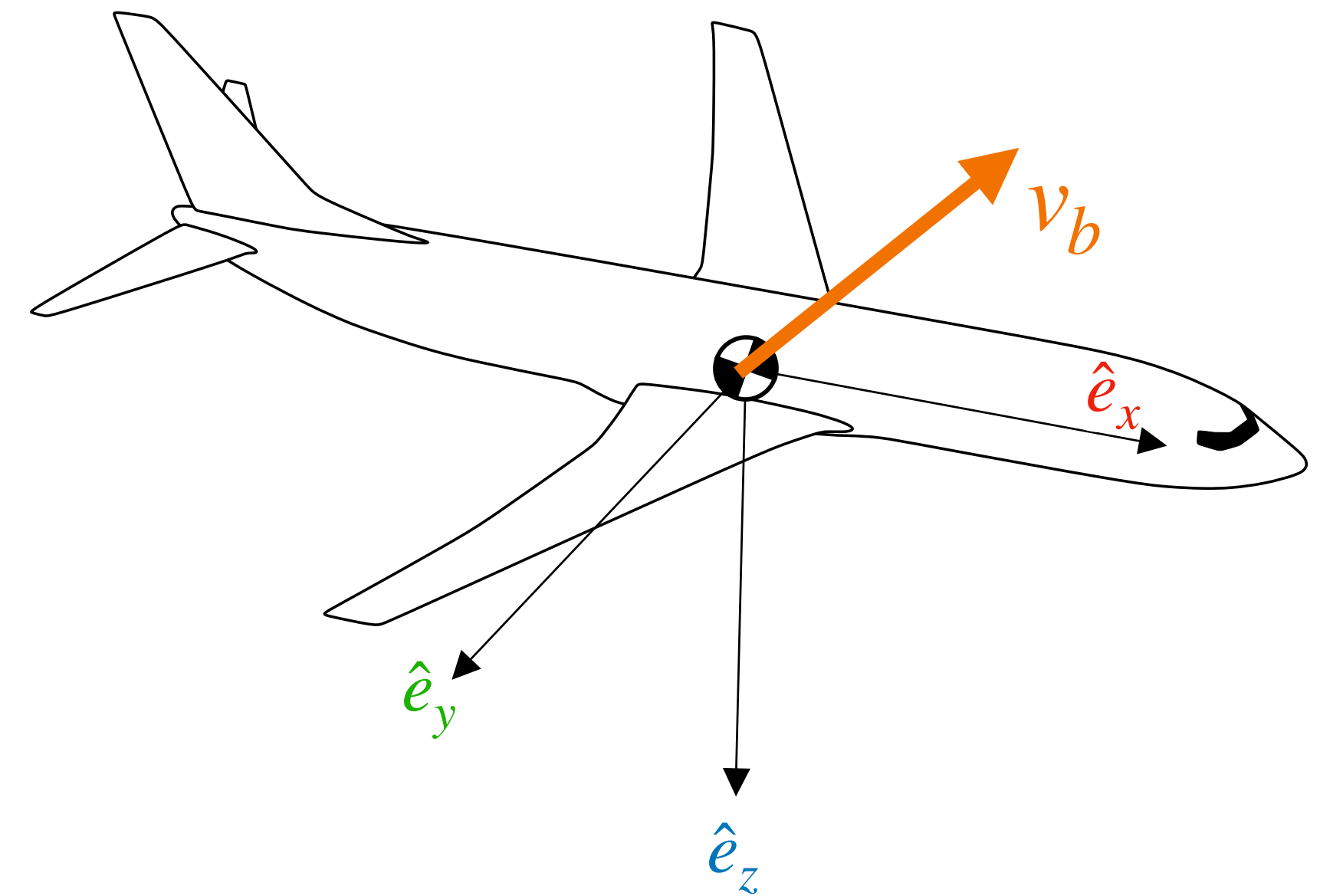


World & Body Frame



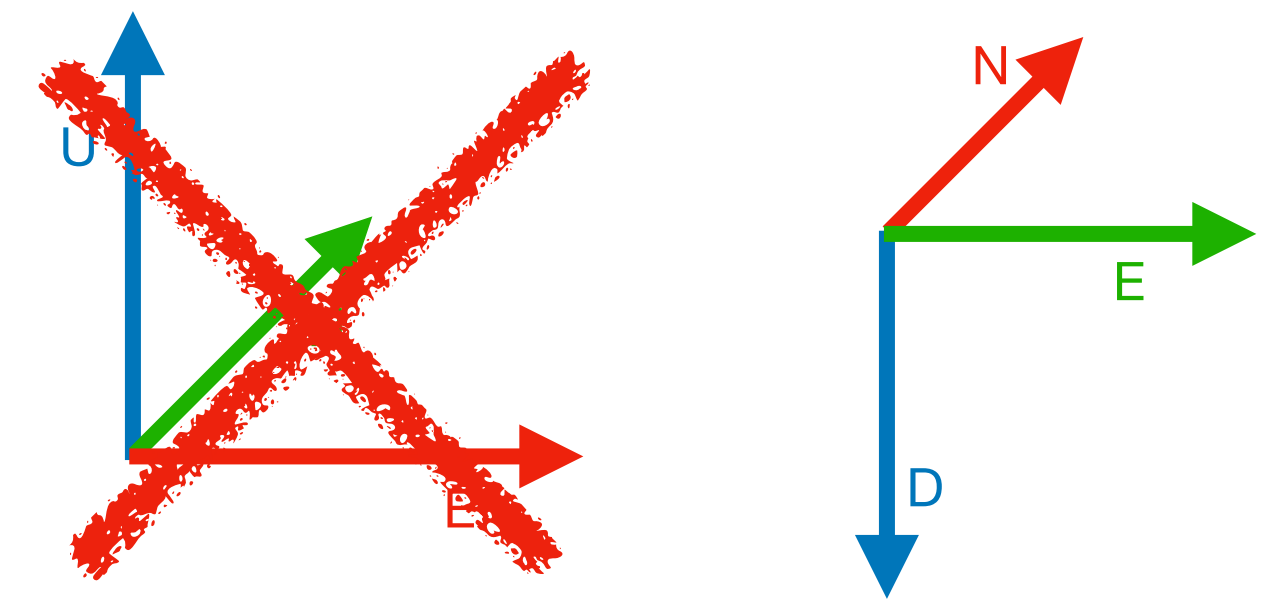
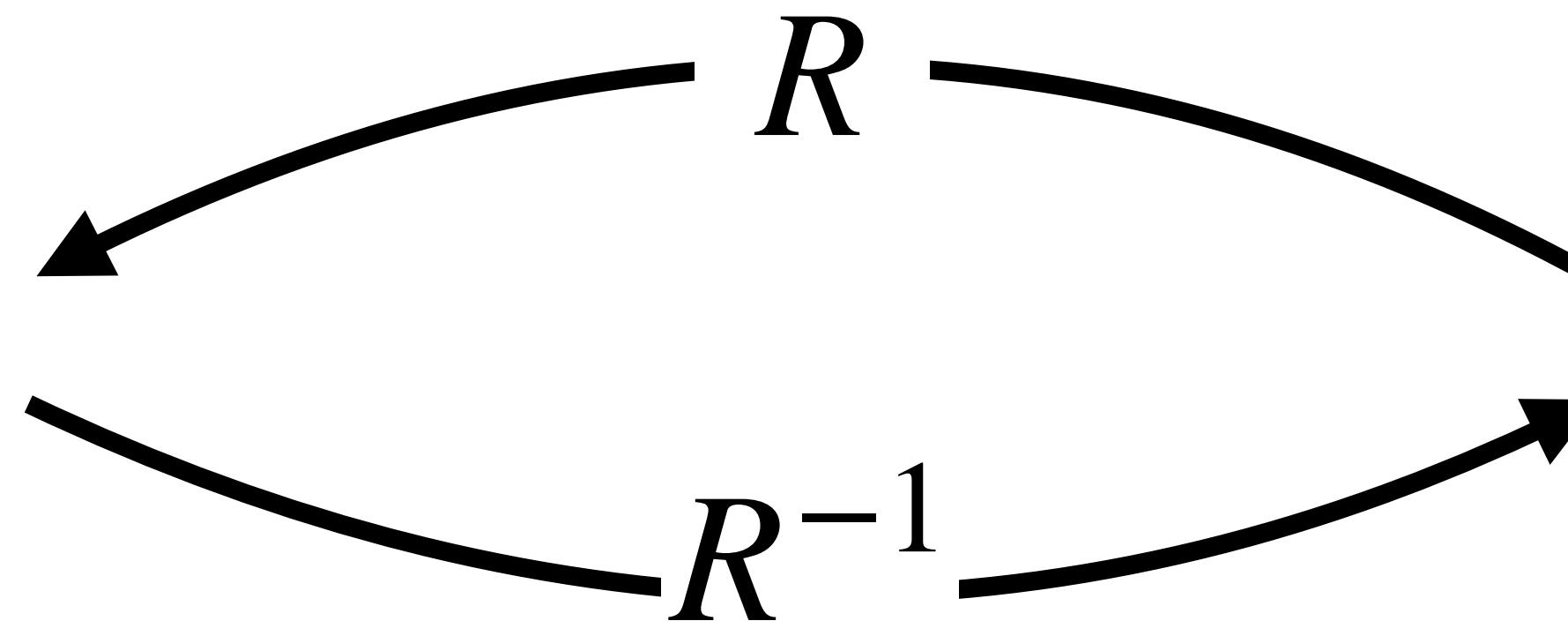
NED World Frame

$$v = Rv_b$$



NED Body Frame

$$v_b = R^{-1}v$$



State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

Orientation $R \in \mathbb{R}^9$

Rot. Vel. [rad/s] $\omega \in \mathbb{R}^3$

State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

States

Position [m] $p \in \mathbb{R}^3$

Velocity [m/s] $v \in \mathbb{R}^3$

Orientation $R \in \mathbb{R}^9$

Rot. Vel. [rad/s] $\omega \in \mathbb{R}^3$



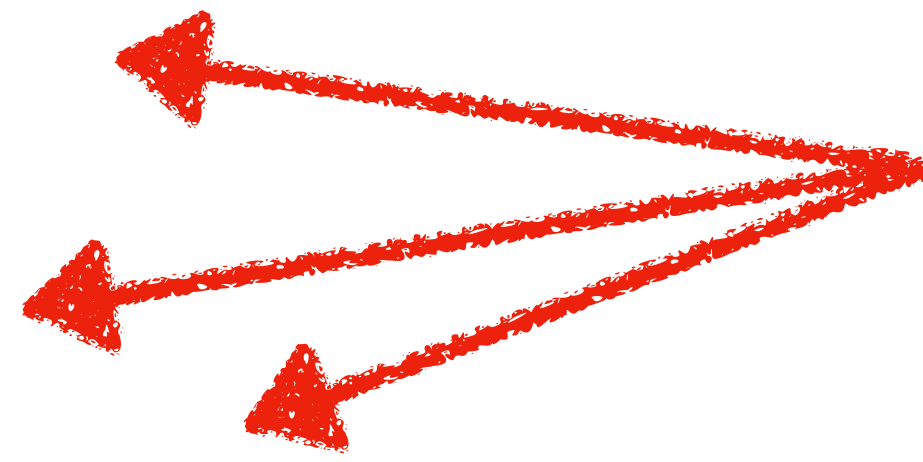
State-Space Model



$$x = \begin{bmatrix} p \\ v \\ R \\ \omega \end{bmatrix} \in \mathbb{R}^{18}$$

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

States

Position [m]	$p \in \mathbb{R}^3$		in world frame
Velocity [m/s]	$v \in \mathbb{R}^3$		
Orientation	$R \in \mathbb{R}^9$		
Rot. Vel. [rad/s]	$\omega \in \mathbb{R}^3$		in body frame

Rotational Dynamics

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Rotational Dynamics

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Rotational Dynamics

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$$\dot{R} = R [\omega]_{\times}$$

Rotational Dynamics

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\boldsymbol{\omega}]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

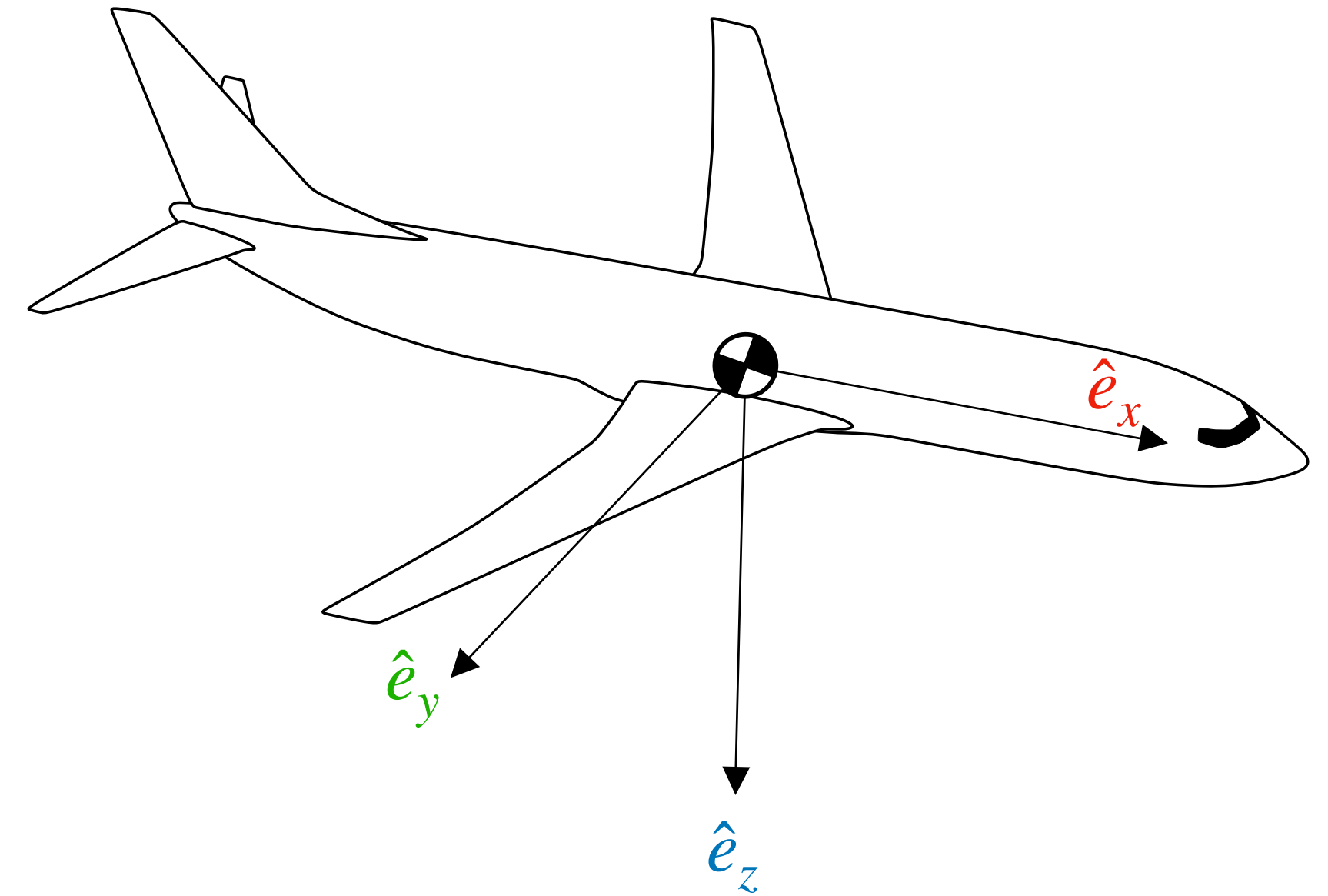
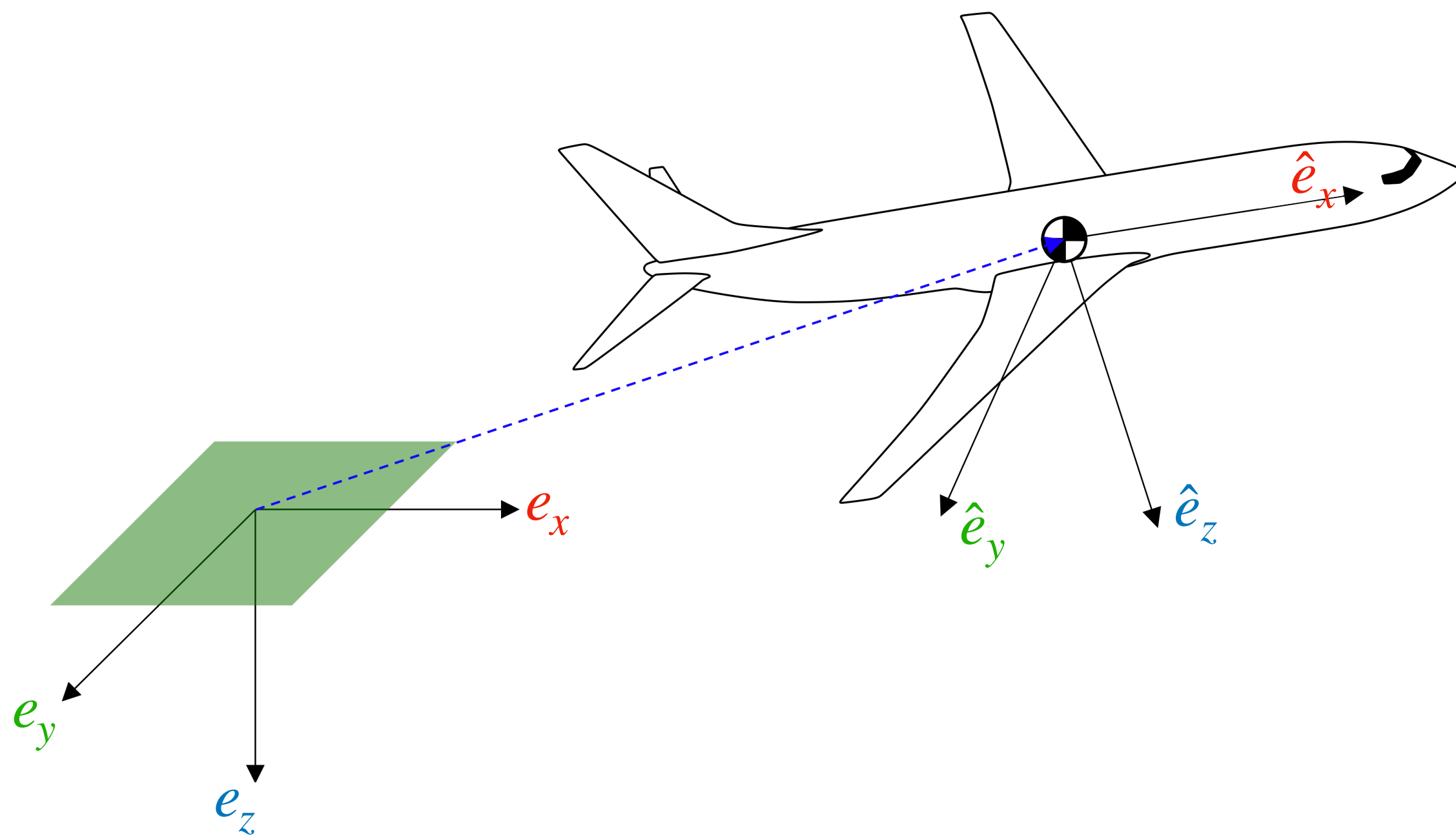
$$\dot{R} = R [\boldsymbol{\omega}]_{\times}$$

Rotational Dynamics

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$$\dot{R} = R [\omega]_{\times}$$

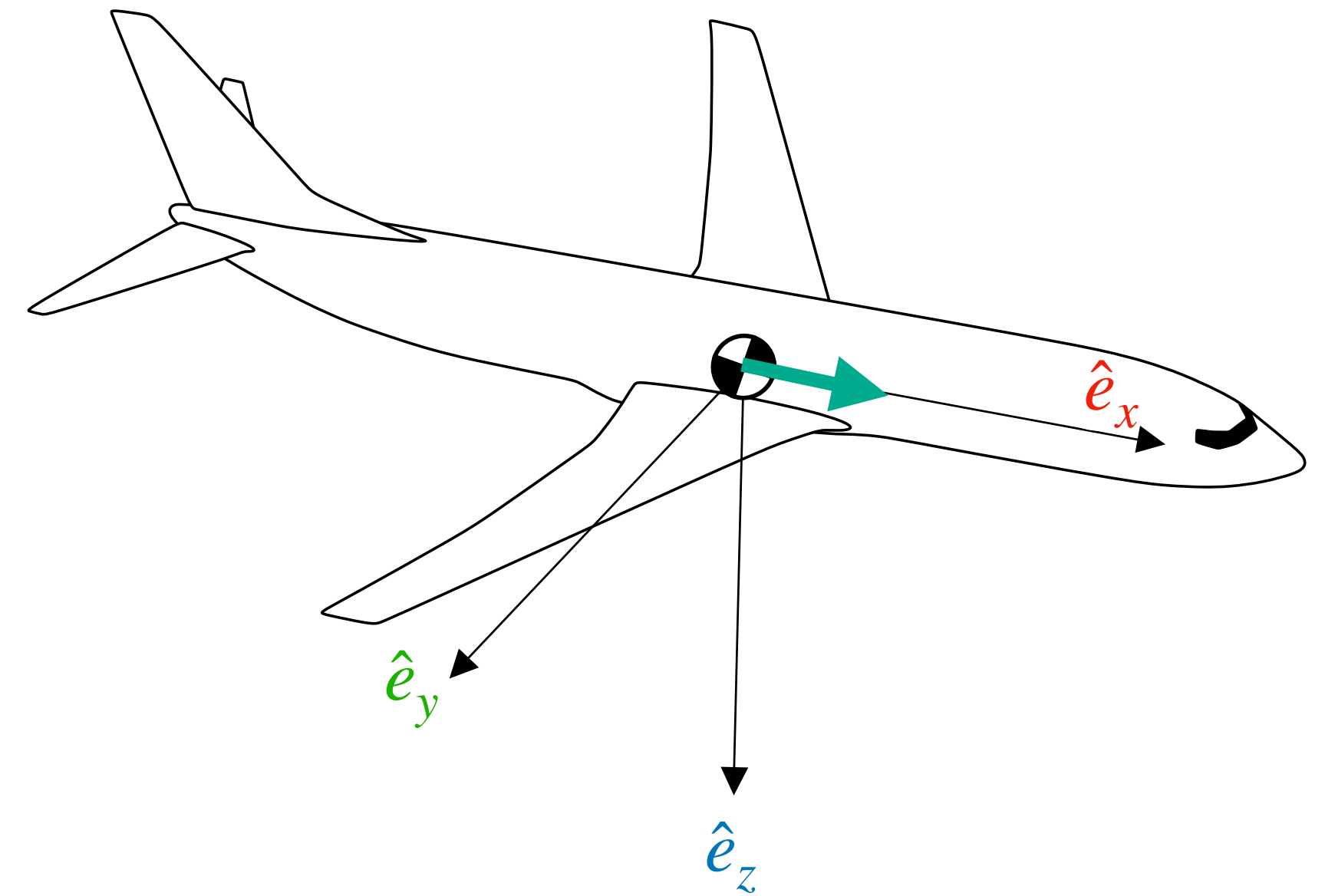
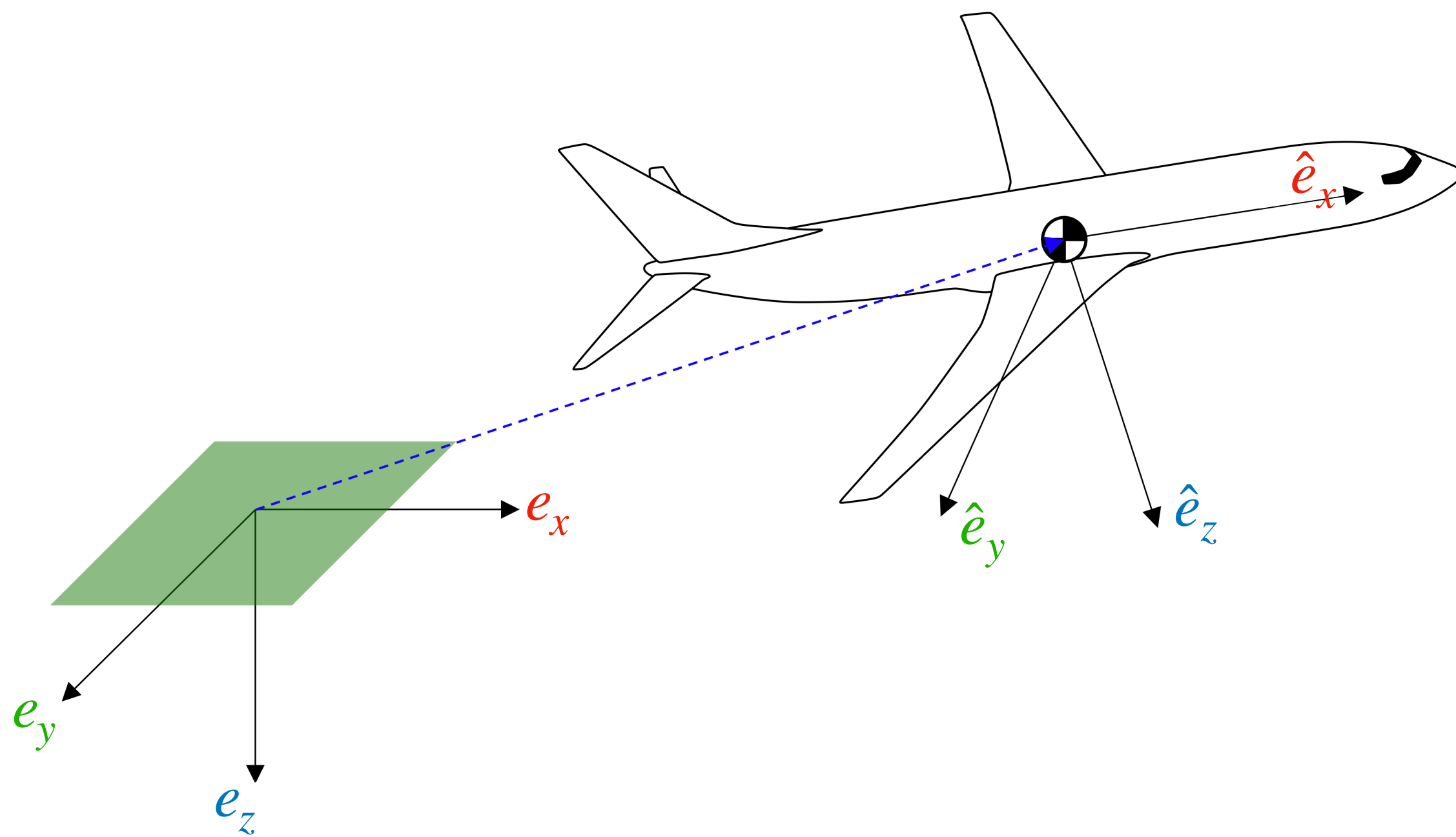


Rotational Dynamics

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$$\dot{R} = R [\omega]_{\times}$$

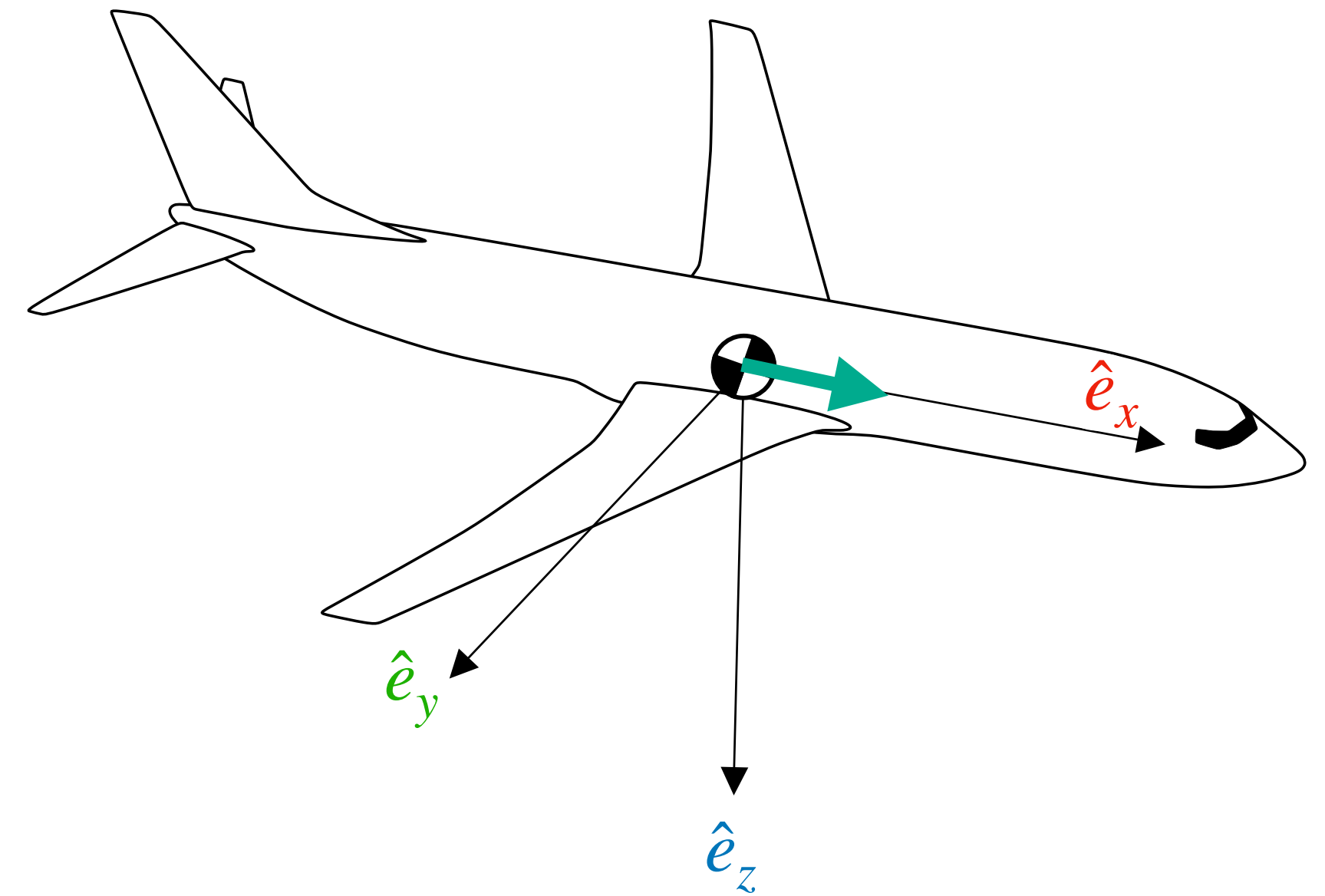
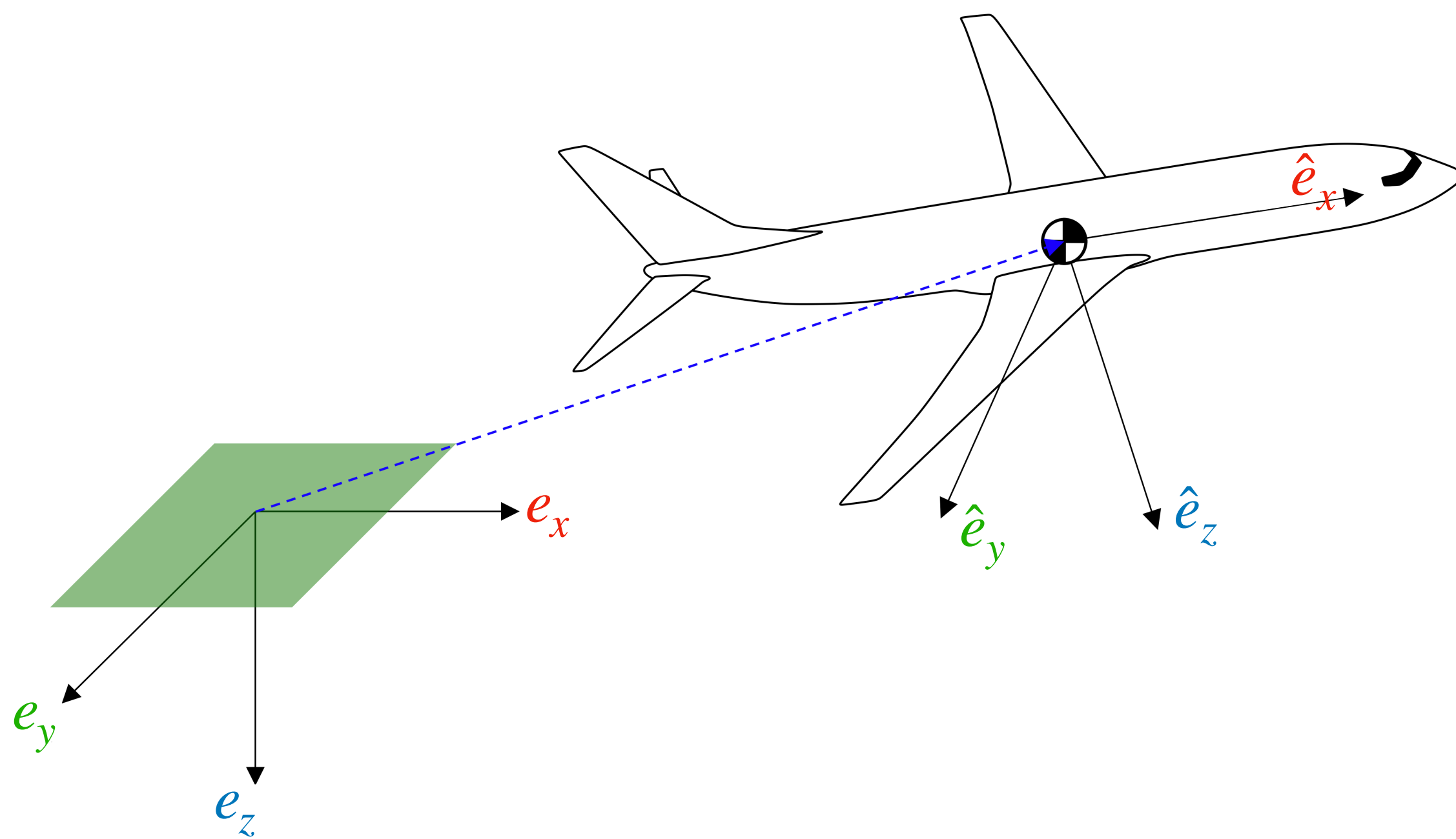


Rotational Dynamics

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

$$\dot{R} = R [\omega]_{\times}$$

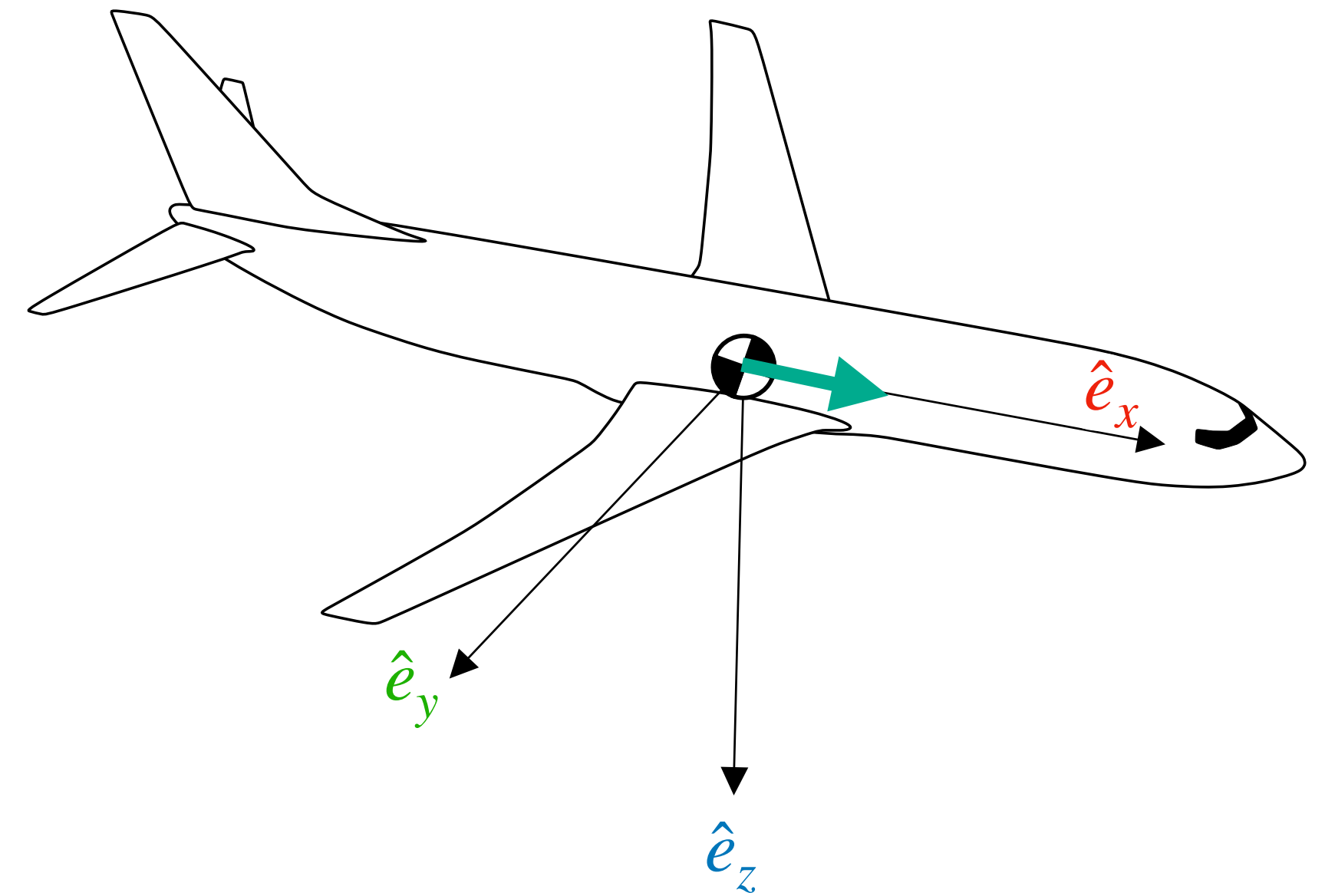
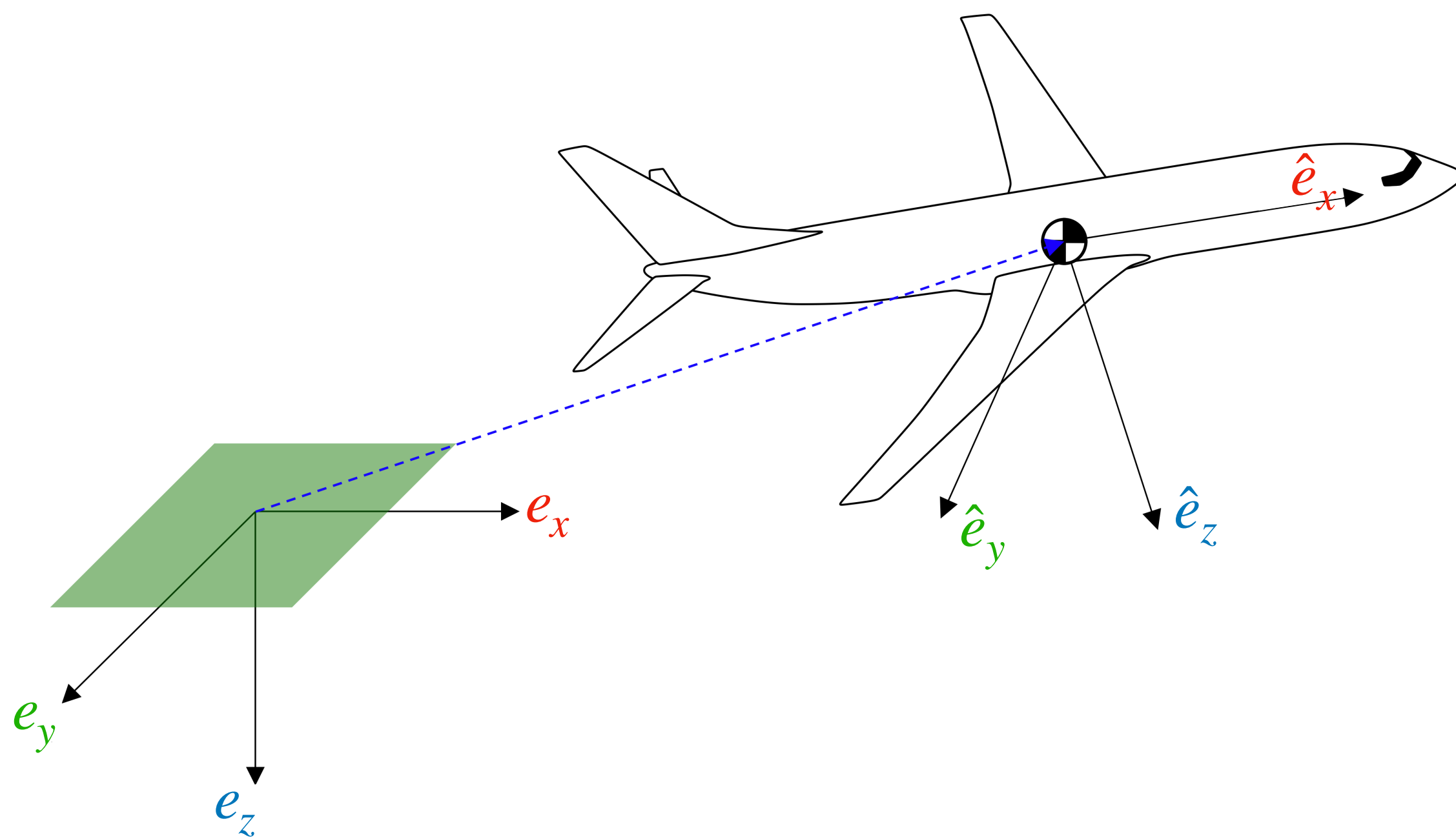


Rotational Dynamics

$$\begin{bmatrix} | & | & | \\ \dot{\hat{e}}_x & \dot{\hat{e}}_y & \dot{\hat{e}}_z \\ | & | & | \end{bmatrix} = R \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

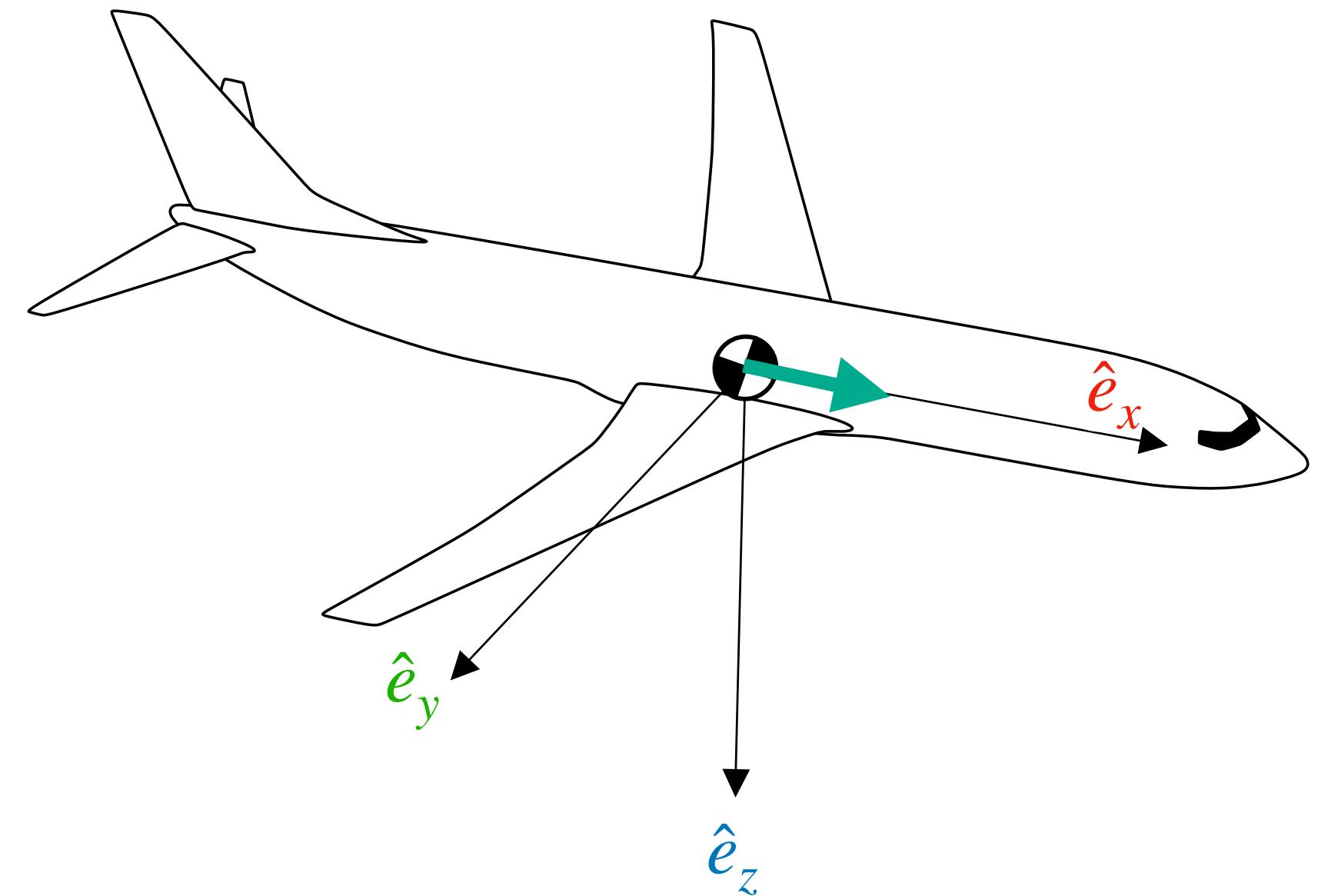
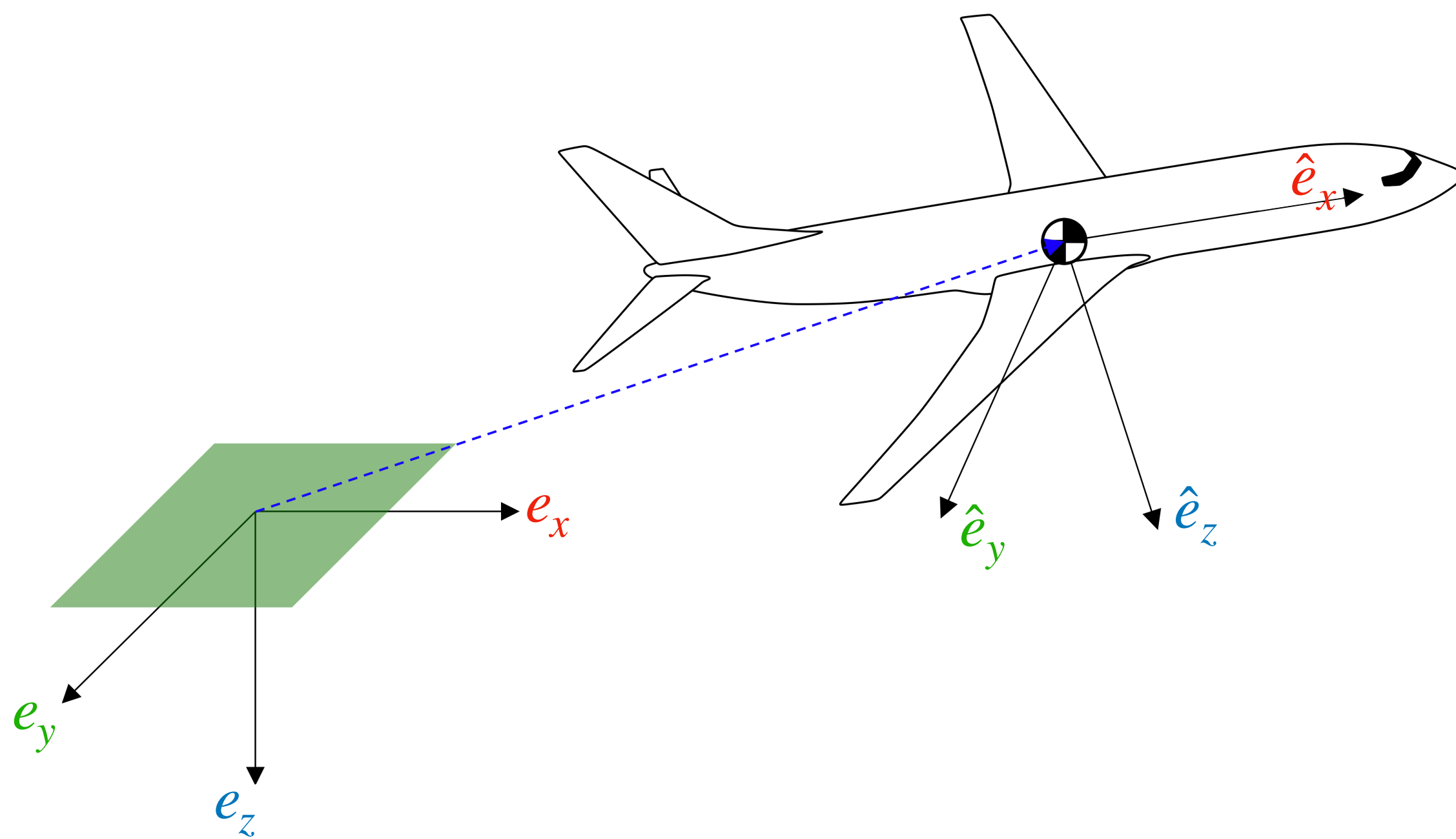


Rotational Dynamics

$$\begin{bmatrix} | & | & | \\ \dot{\hat{e}}_x & \dot{\hat{e}}_y & \dot{\hat{e}}_z \\ | & | & | \end{bmatrix} = R \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

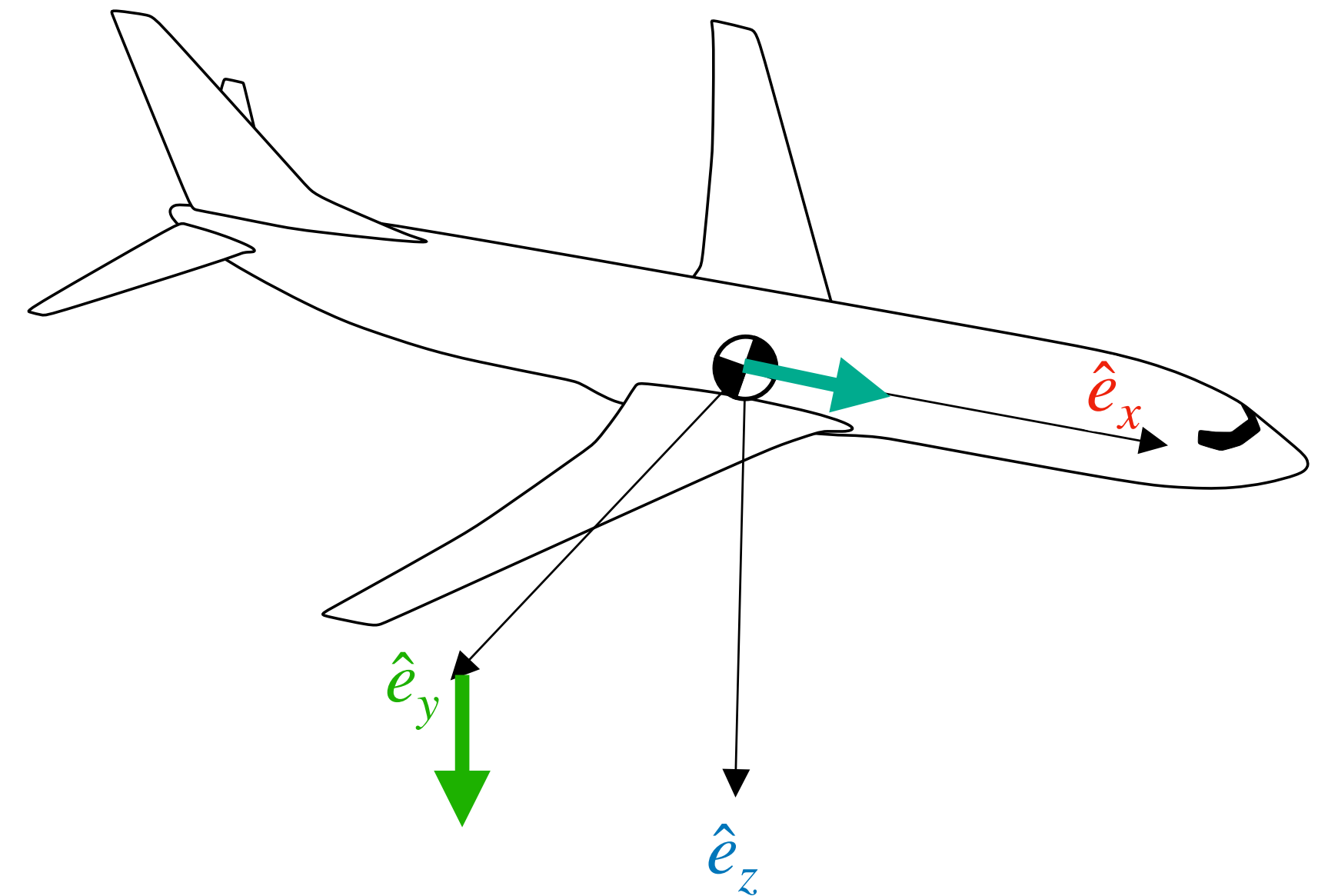
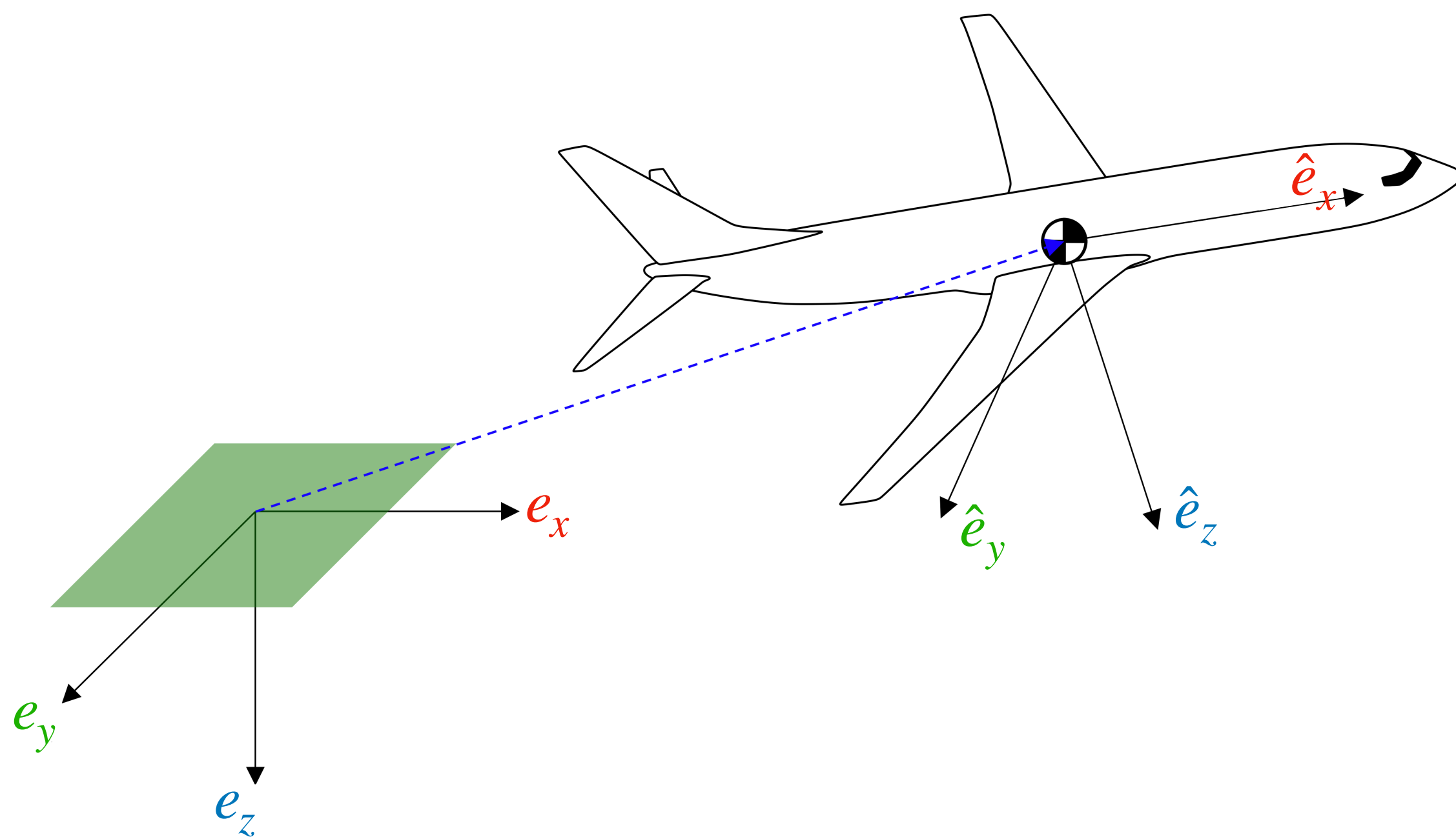


Rotational Dynamics

$$\begin{bmatrix} | & | & | \\ \dot{\hat{e}}_x & \dot{\hat{e}}_y & \dot{\hat{e}}_z \\ | & | & | \end{bmatrix} = R \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

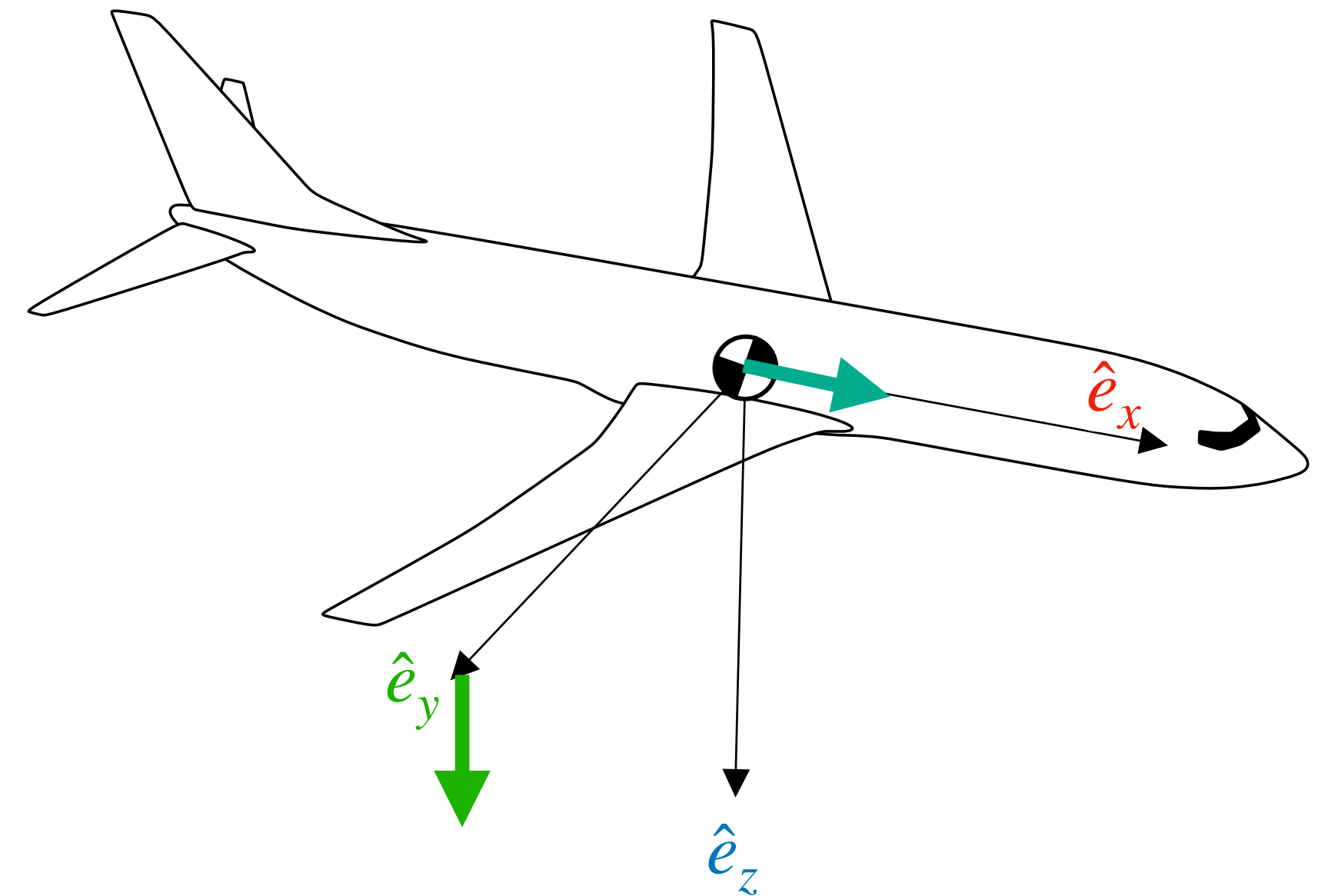
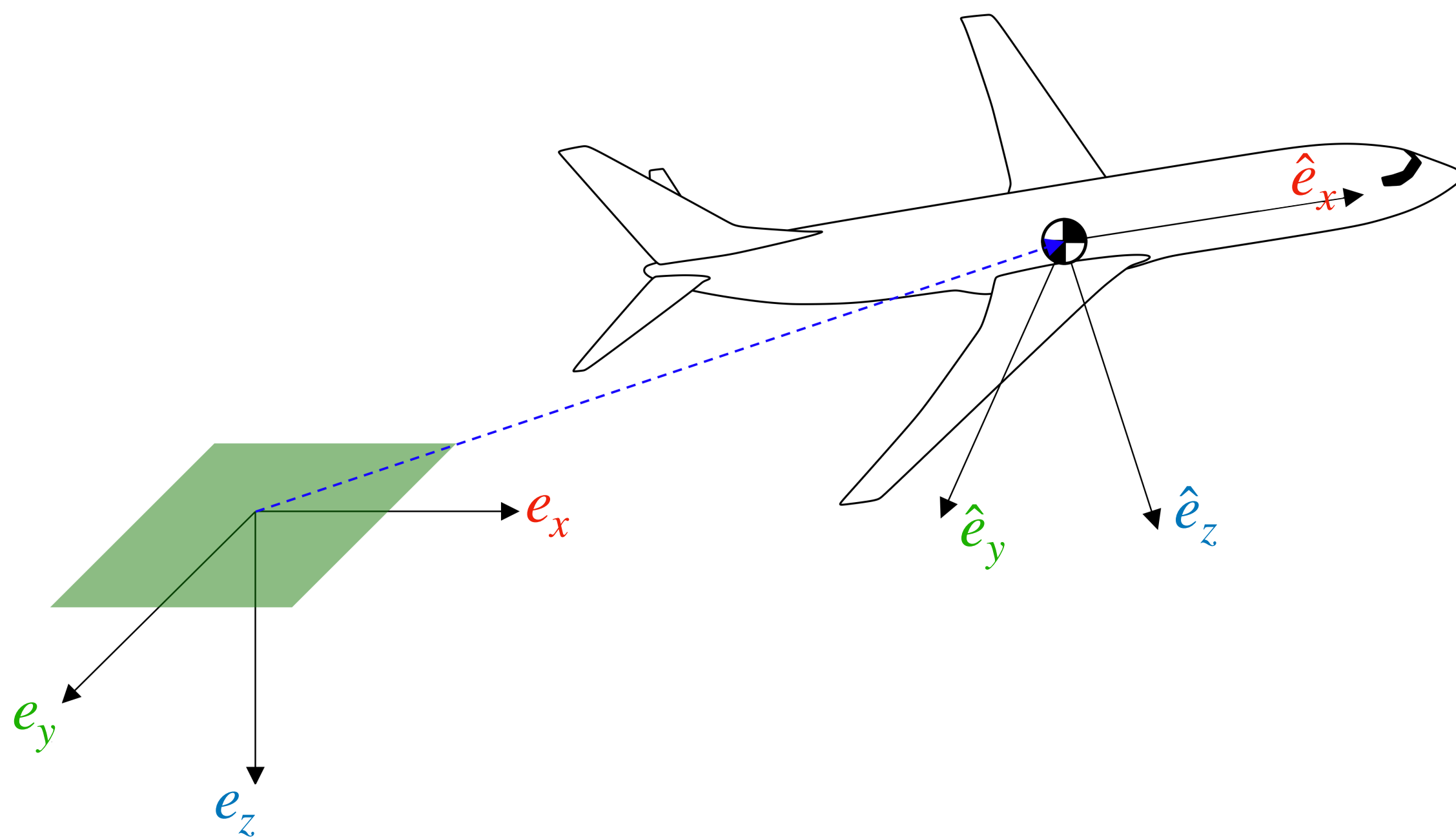


Rotational Dynamics

$$\begin{bmatrix} | & | & | \\ \dot{\hat{e}}_x & \dot{\hat{e}}_y & \dot{\hat{e}}_z \\ | & | & | \end{bmatrix} = R \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

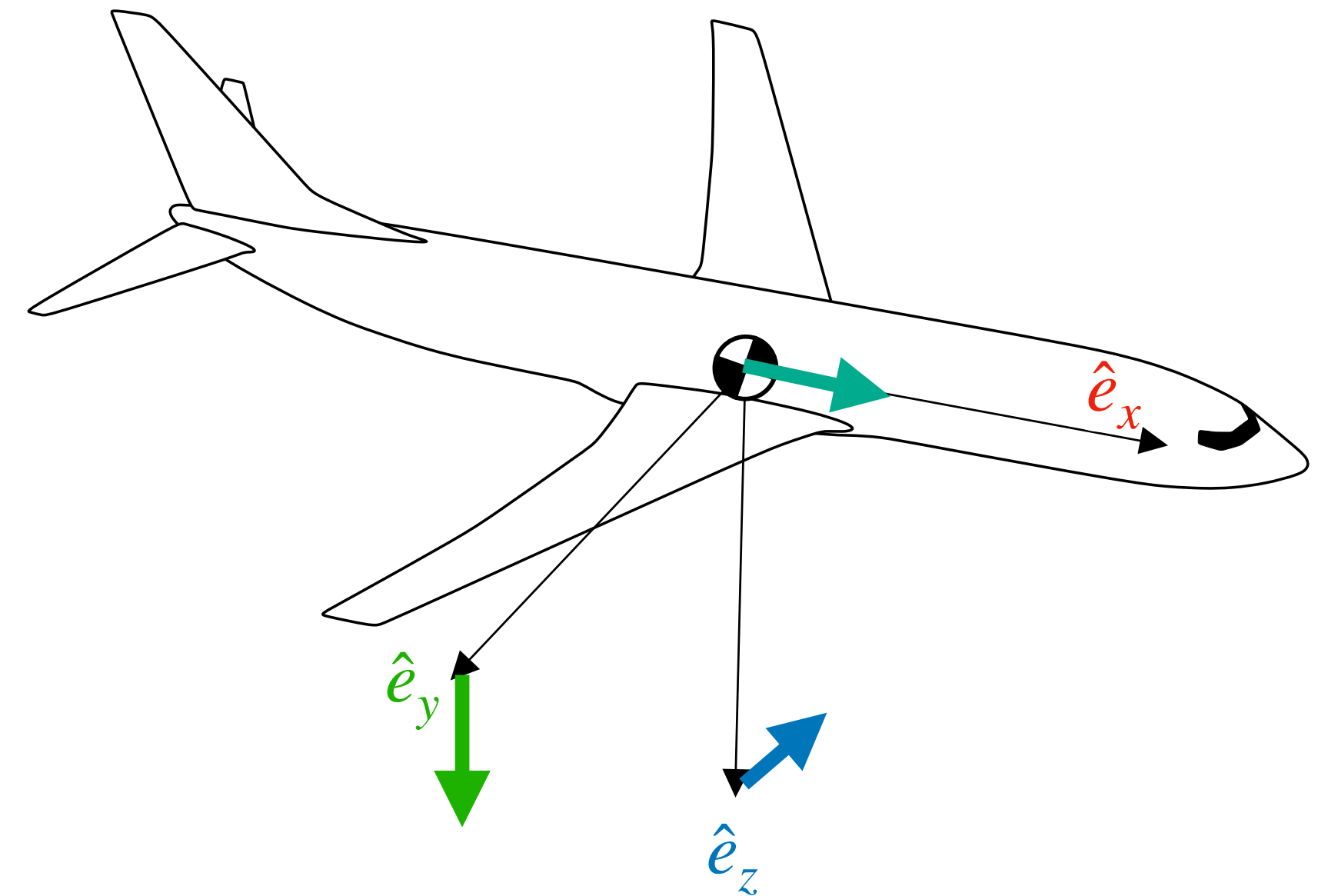
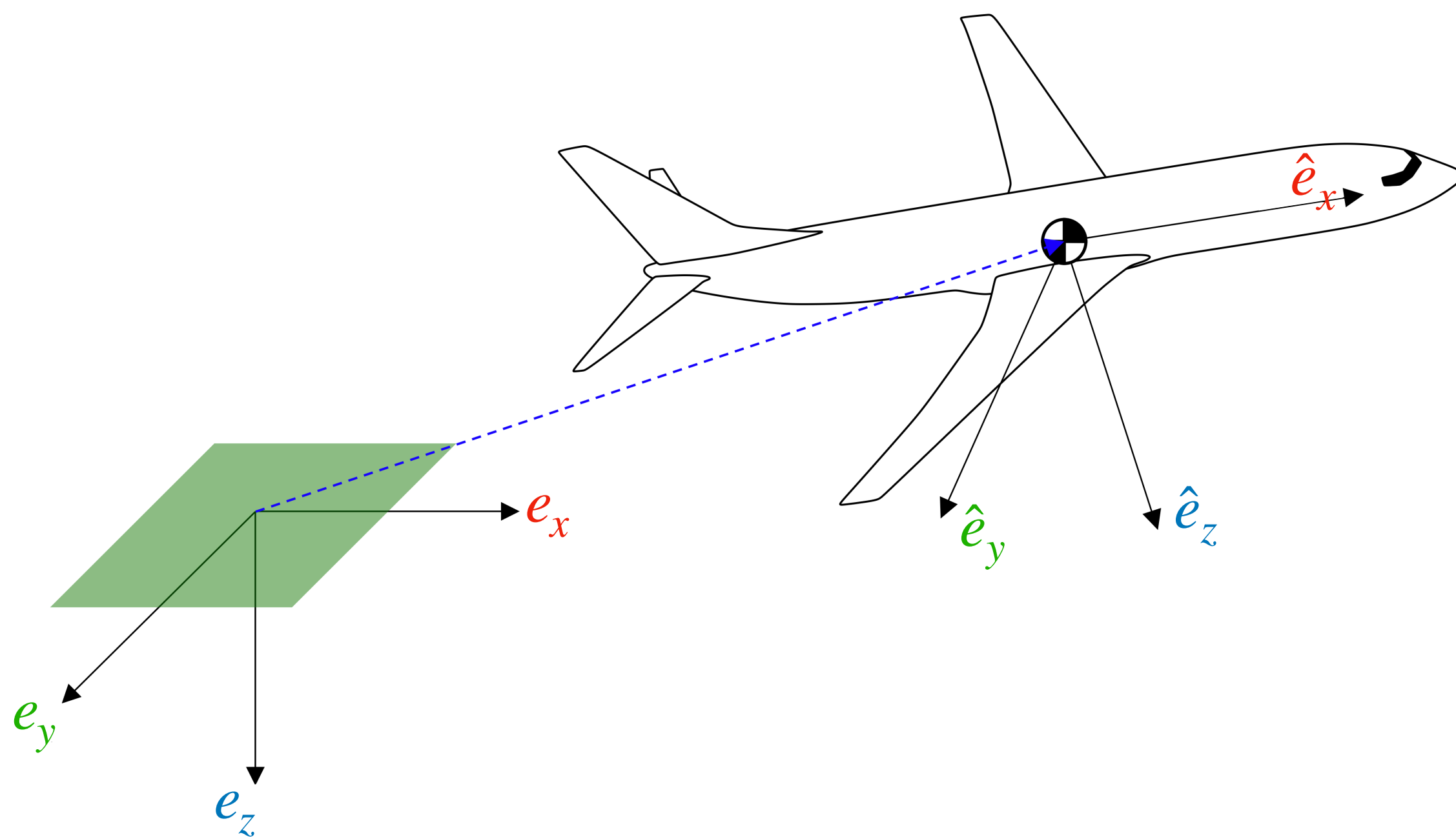


Rotational Dynamics

$$\begin{bmatrix} | & | & | \\ \dot{\hat{e}}_x & \dot{\hat{e}}_y & \dot{\hat{e}}_z \\ | & | & | \end{bmatrix} = R \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

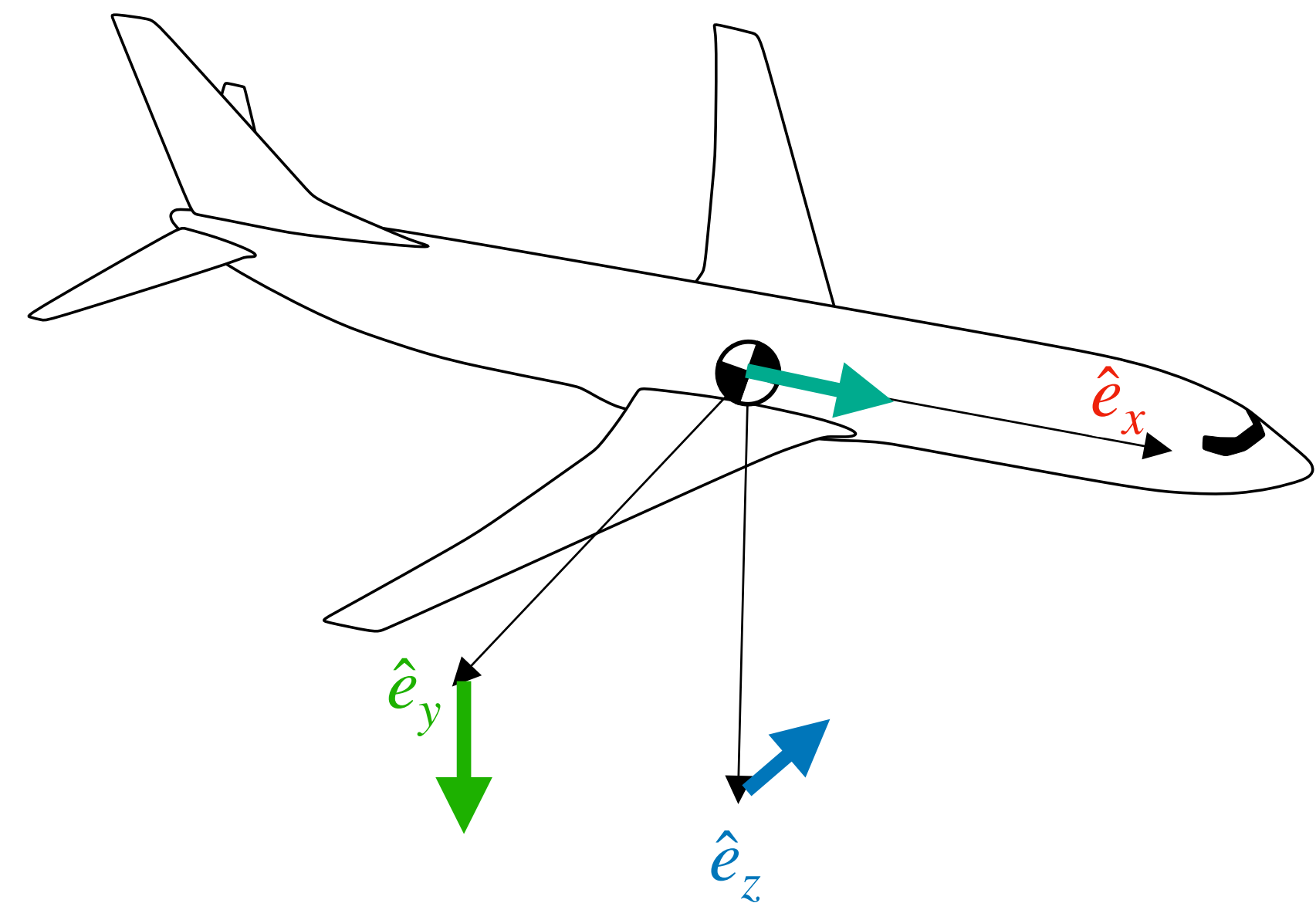
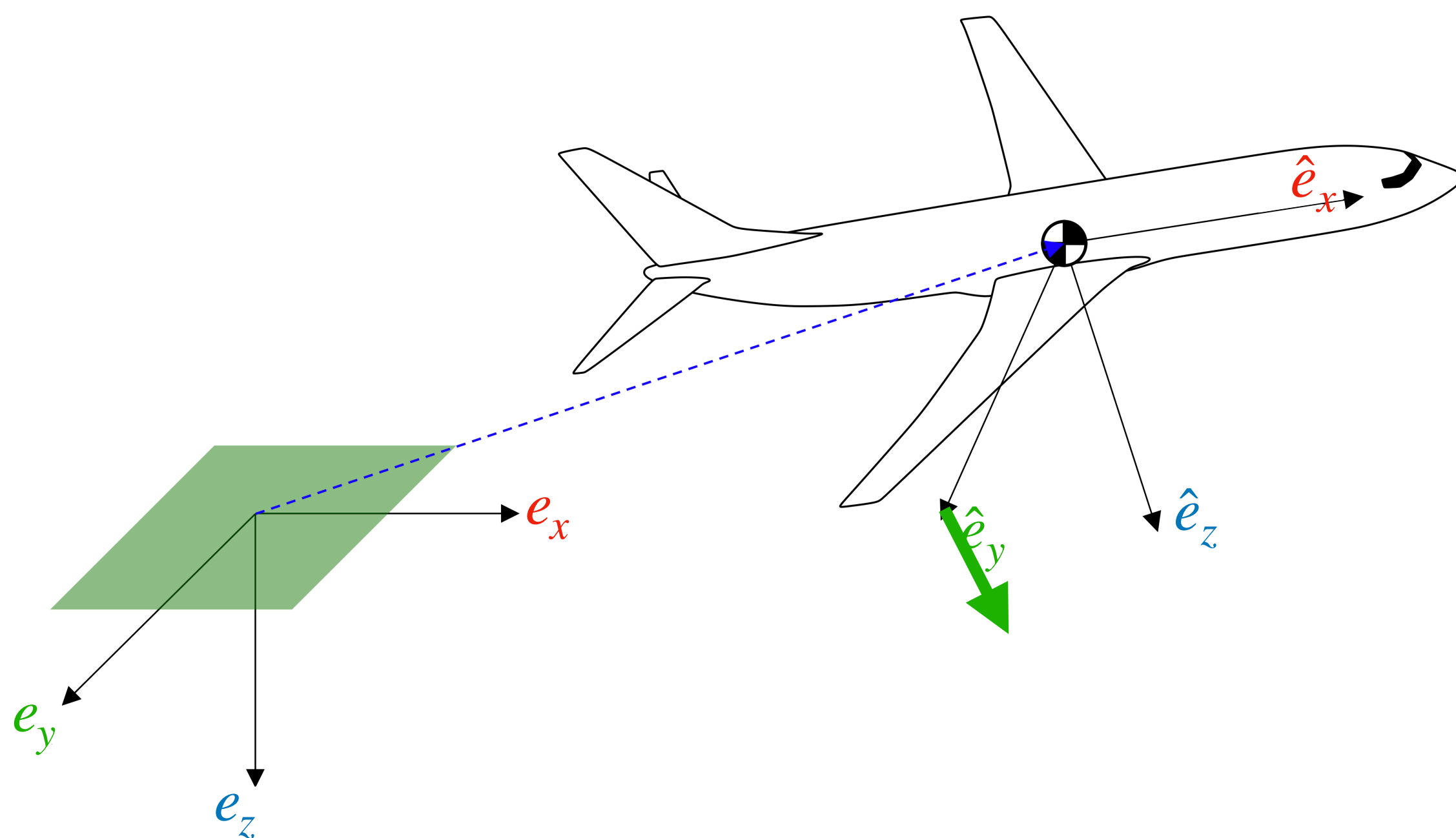


Rotational Dynamics

$$\begin{bmatrix} | & | & | \\ \dot{\hat{e}}_x & \dot{\hat{e}}_y & \dot{\hat{e}}_z \\ | & | & | \end{bmatrix} = R \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

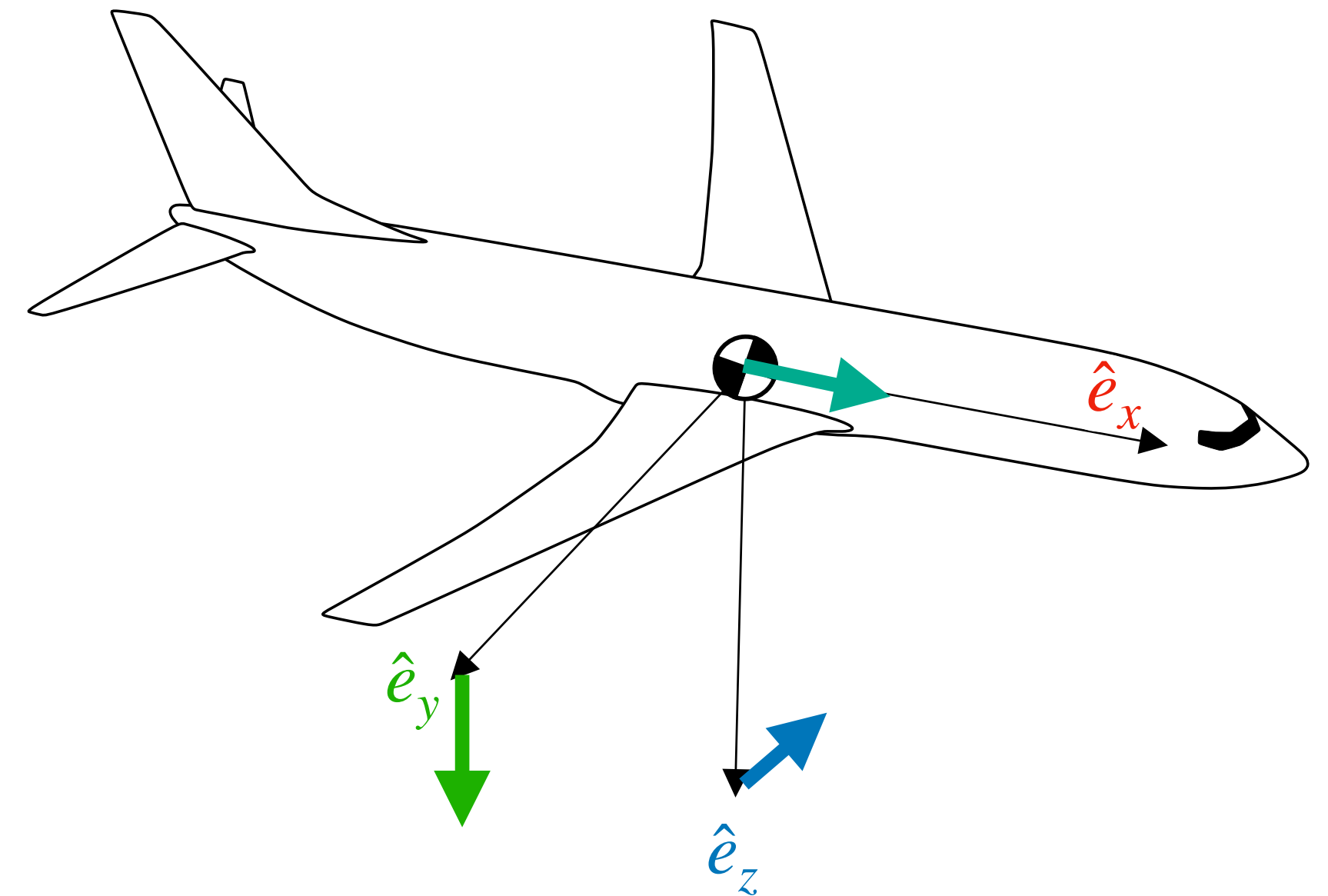
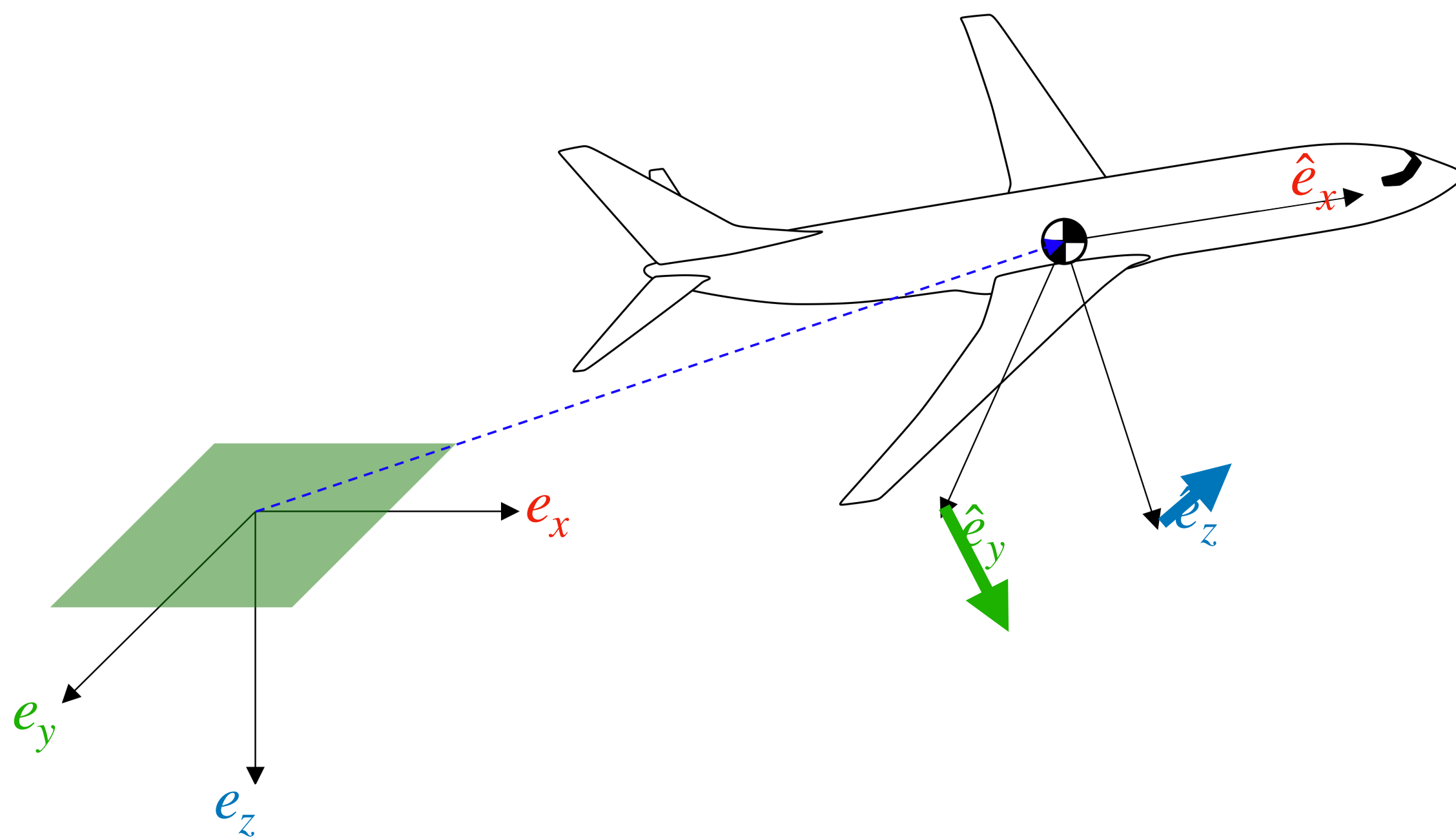


Rotational Dynamics

$$\begin{bmatrix} | & | & | \\ \dot{\hat{e}}_x & \dot{\hat{e}}_y & \dot{\hat{e}}_z \\ | & | & | \end{bmatrix} = R \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$

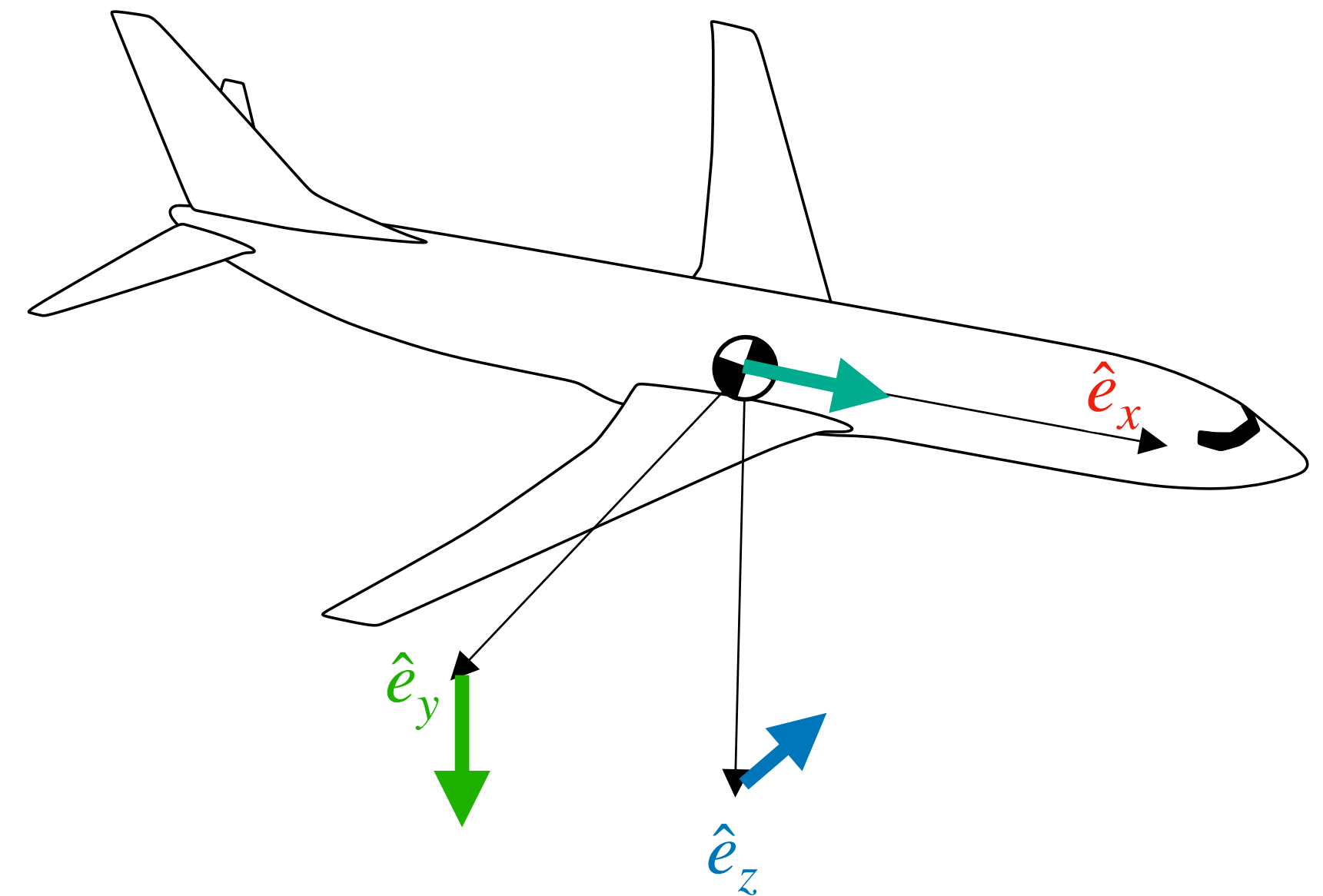
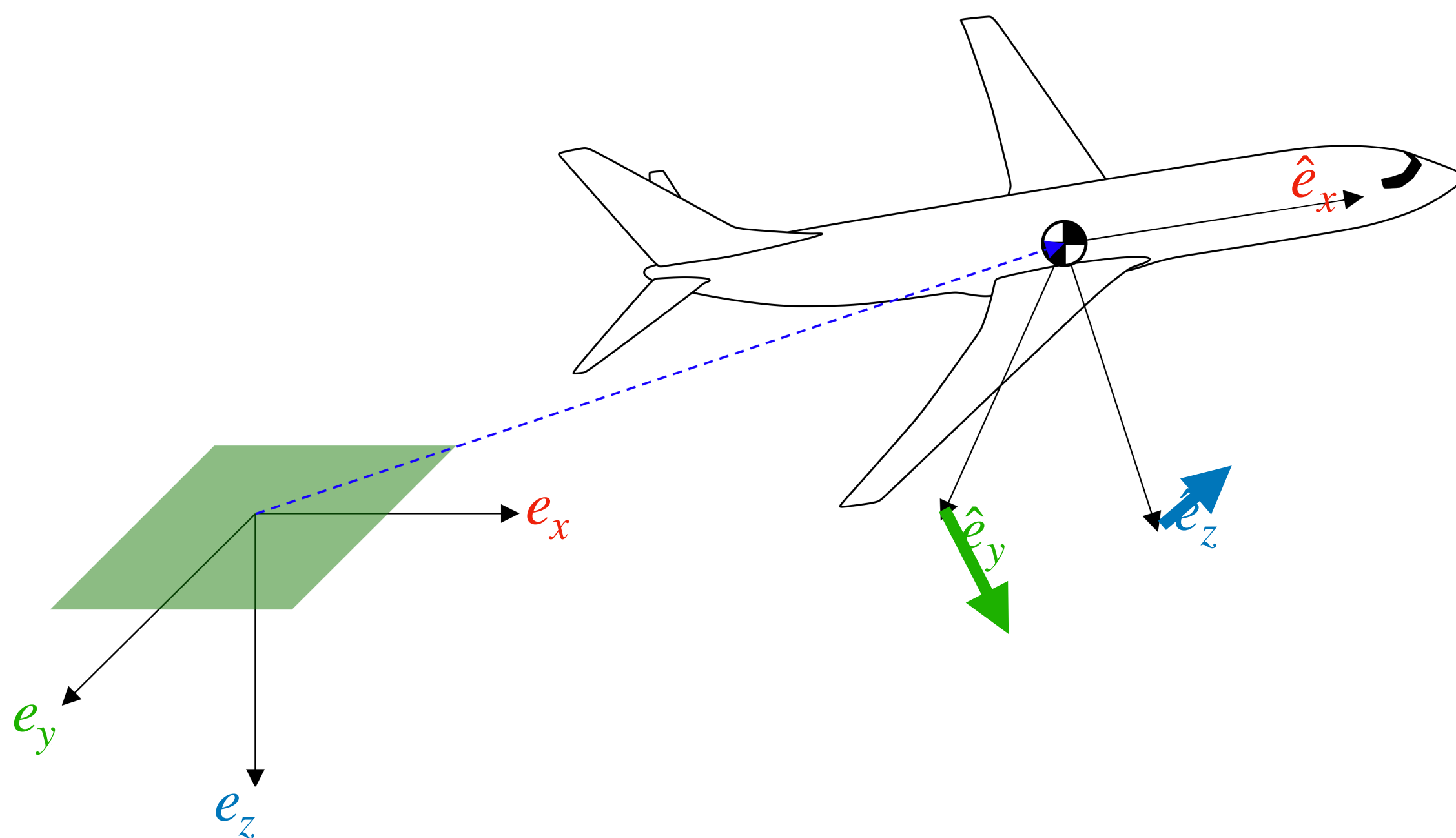


Rotational Dynamics

$$\begin{bmatrix} | & | & | \\ \dot{\hat{e}}_x & \dot{\hat{e}}_y & \dot{\hat{e}}_z \\ | & | & | \end{bmatrix} = R \begin{bmatrix} | & | & | \\ 0 & 0 & -\omega_x \\ | & \omega_x & 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}$$

$$[\omega]_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}$$



State-Space Model



$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \end{bmatrix}$$

State-Space Model



$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \end{bmatrix}$$

State-Space Model



$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \\ R[\omega]_{\times} \end{bmatrix}$$

State-Space Model



$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \\ R[\omega]_{\times} \\ J^{-1} \left(\omega \times J\omega + M_{\text{aero}}(x, u) \right) \end{bmatrix}$$

State-Space Model



$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \\ R[\omega]_{\times} \\ J^{-1} \left(\omega \times J\omega + M_{\text{aero}}(x, u) \right) \end{bmatrix}$$

State-Space Model



$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \\ R[\omega]_{\times} \\ J^{-1} \left(\underbrace{\omega \times J\omega}_{\text{gyroscopic torque}} + M_{\text{aero}}(x, u) \right) \end{bmatrix}$$

State-Space Model



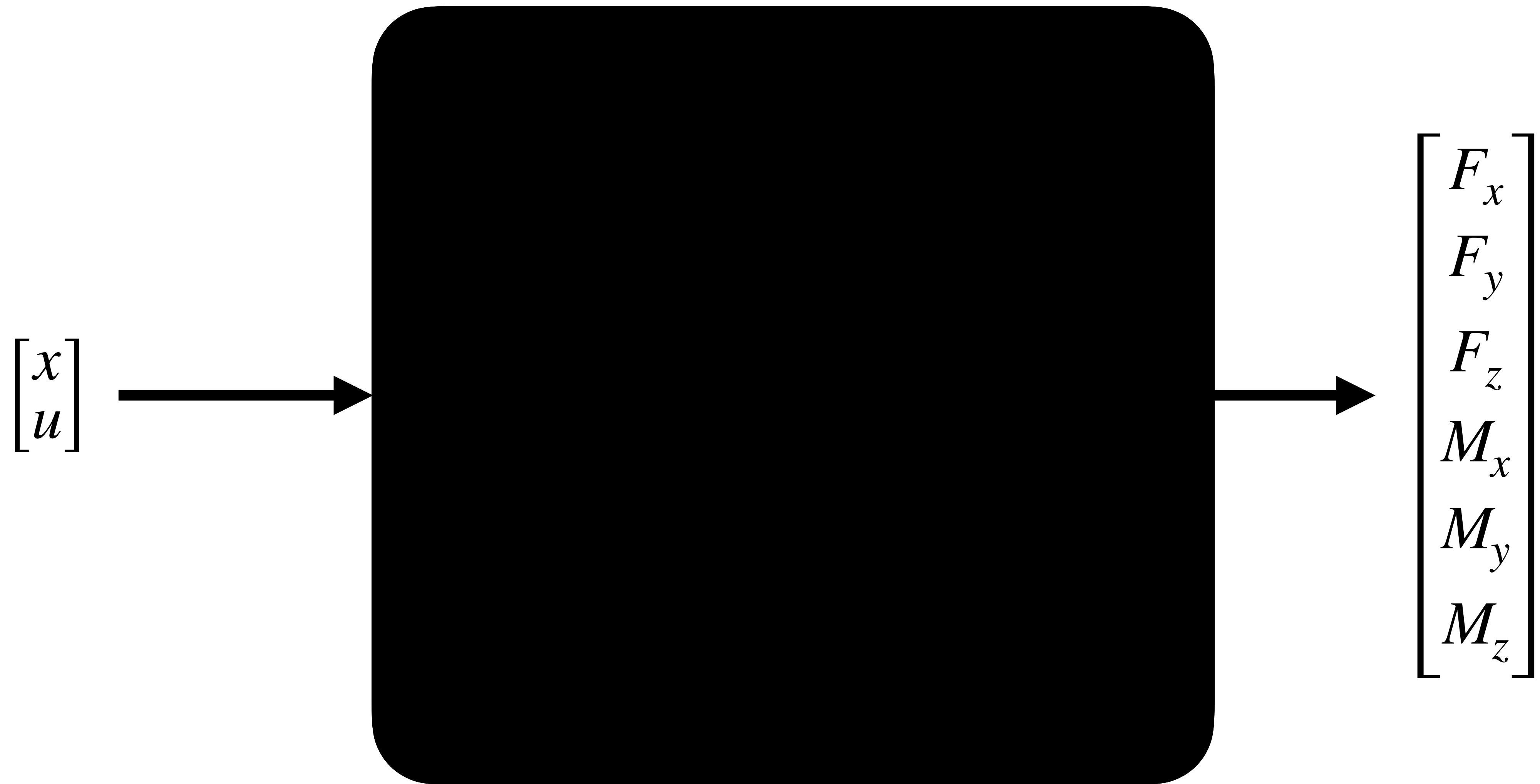
$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(F_{\text{aero}}(x, u) + F_g \right) \\ R[\omega]_{\times} \\ J^{-1} \left(\omega \times J\omega + M_{\text{aero}}(x, u) \right) \end{bmatrix}$$

Aerodynamic Model

$$F_{\text{aero}}(x, u) = \begin{bmatrix} F_x(x, u) \\ F_y(x, u) \\ F_z(x, u) \end{bmatrix}$$

$$M_{\text{aero}}(x, u) = \begin{bmatrix} M_x(x, u) \\ M_y(x, u) \\ M_z(x, u) \end{bmatrix}$$

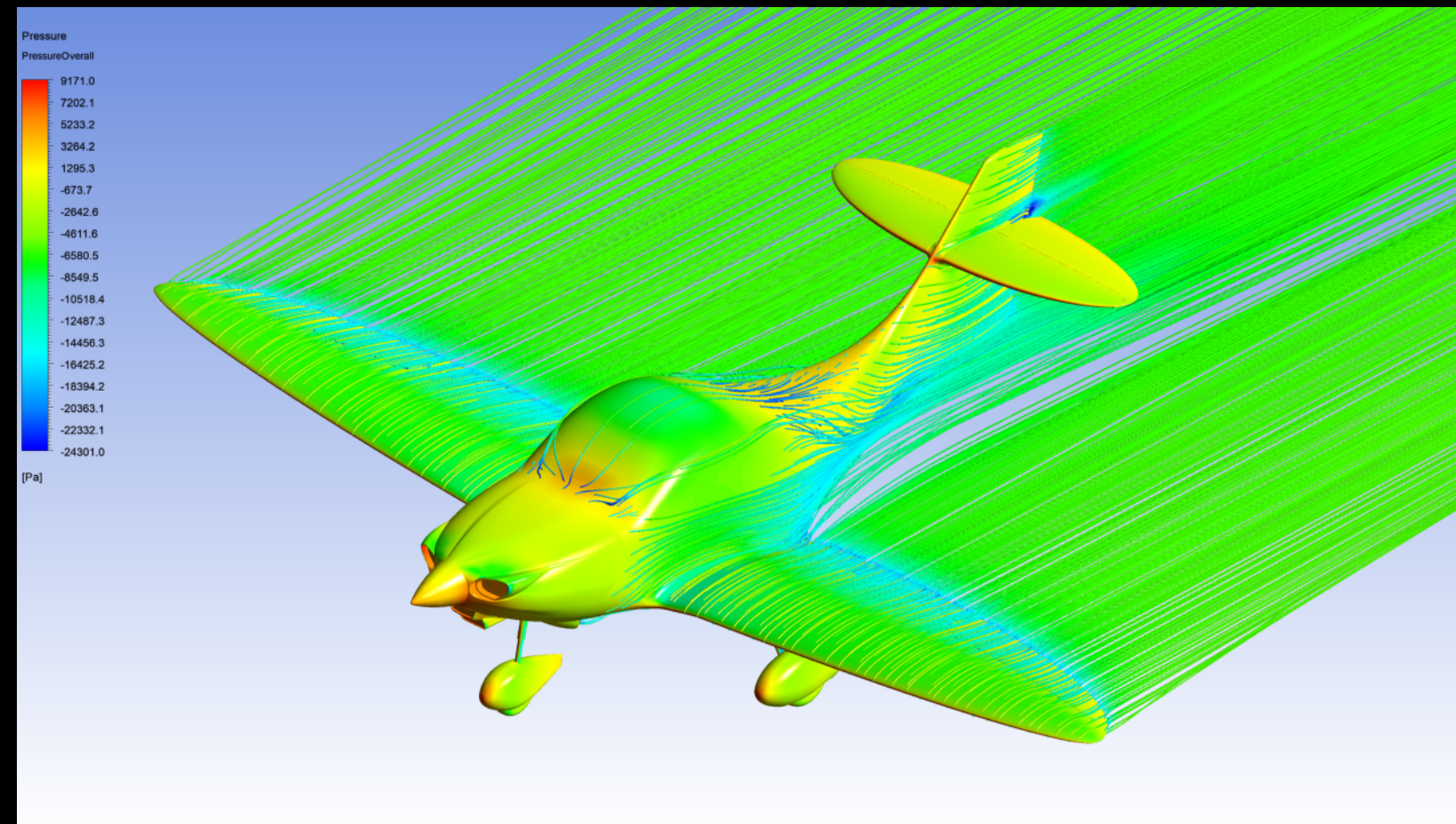
Aerodynamic Model



Aerodynamic Model

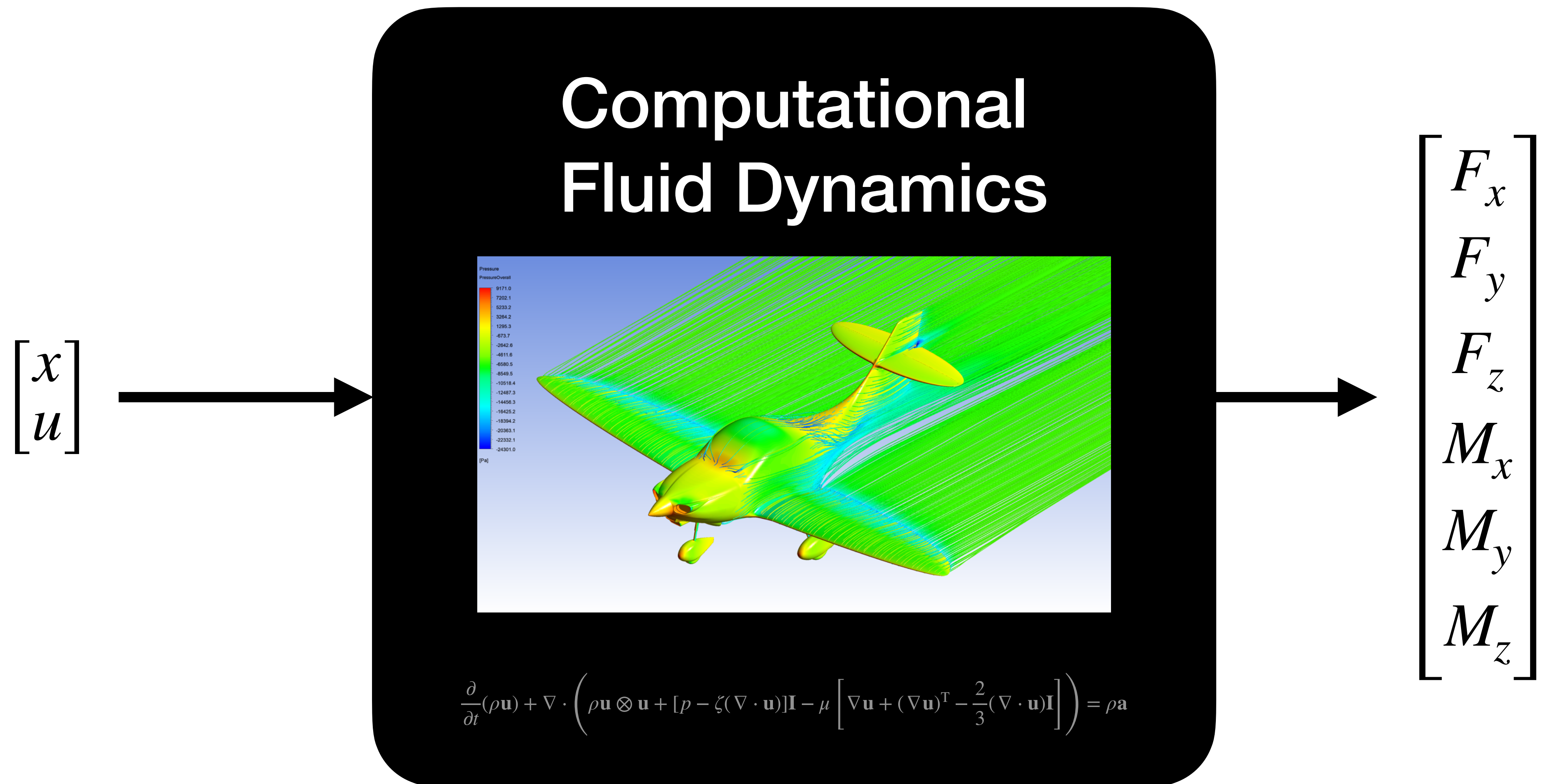
Computational Fluid Dynamics

$$\begin{bmatrix} x \\ u \end{bmatrix}$$

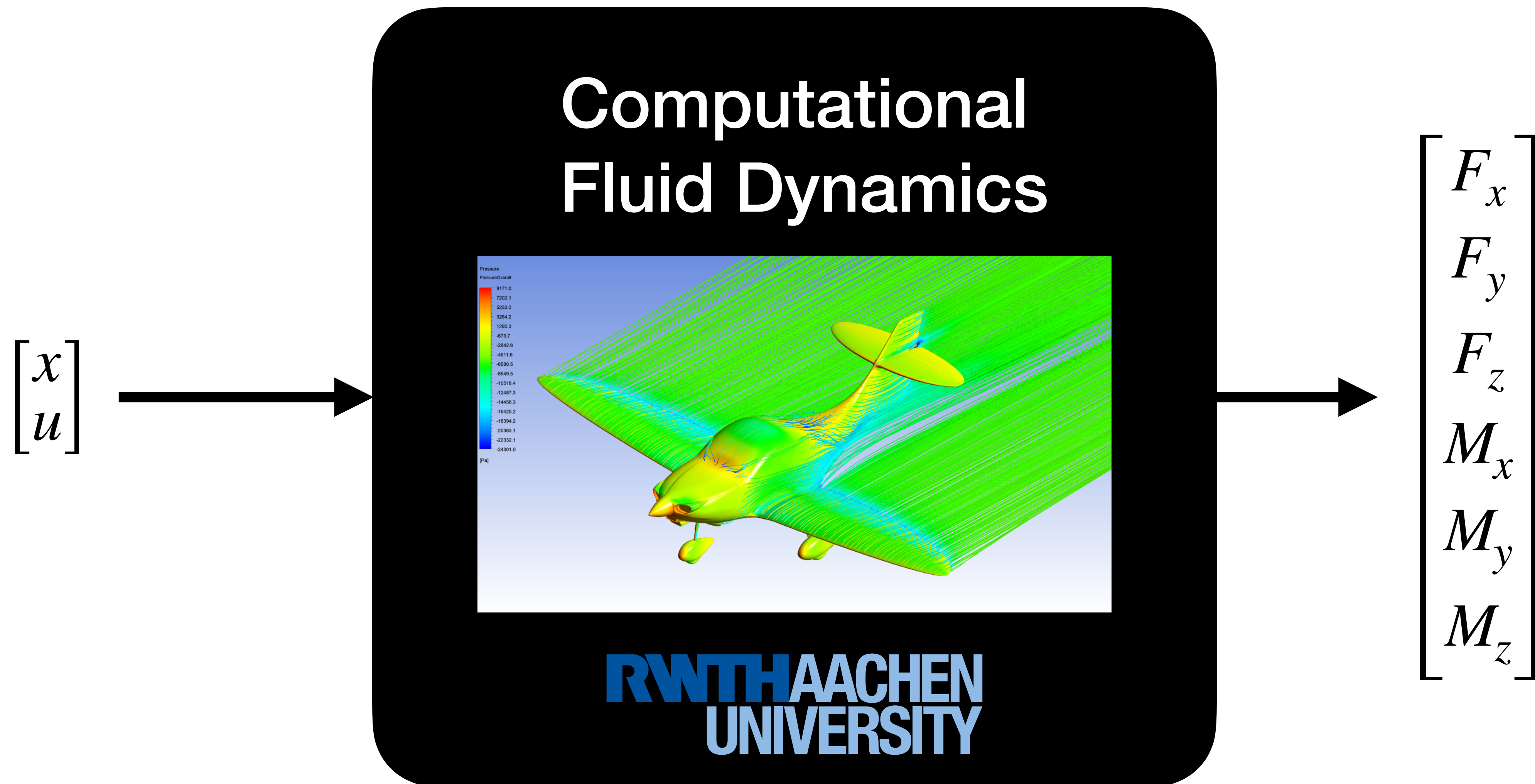


$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix}$$

Aerodynamic Model



Aerodynamic Model

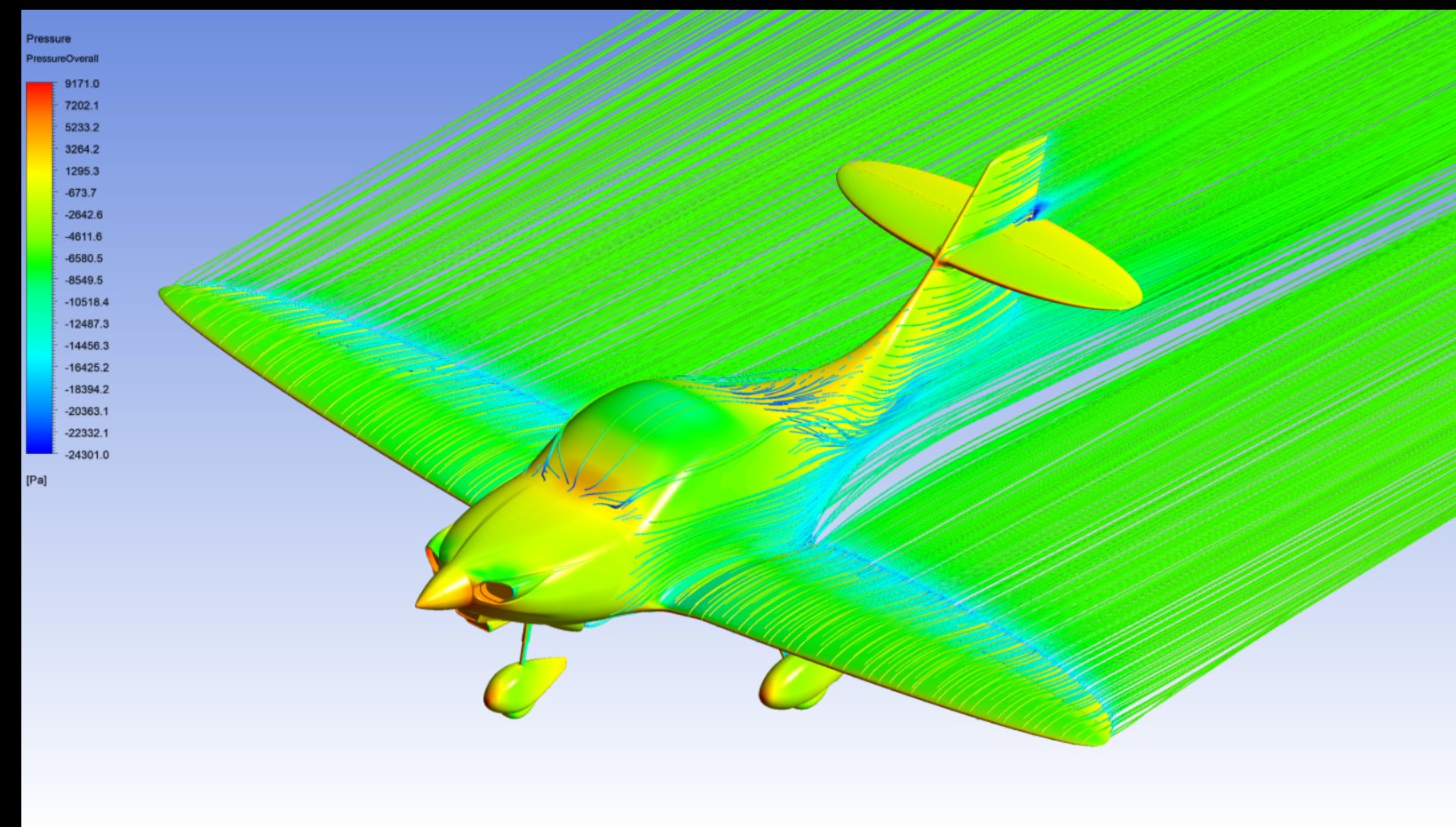


Aerodynamic Model

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix}$$



Computational
Fluid Dynamics

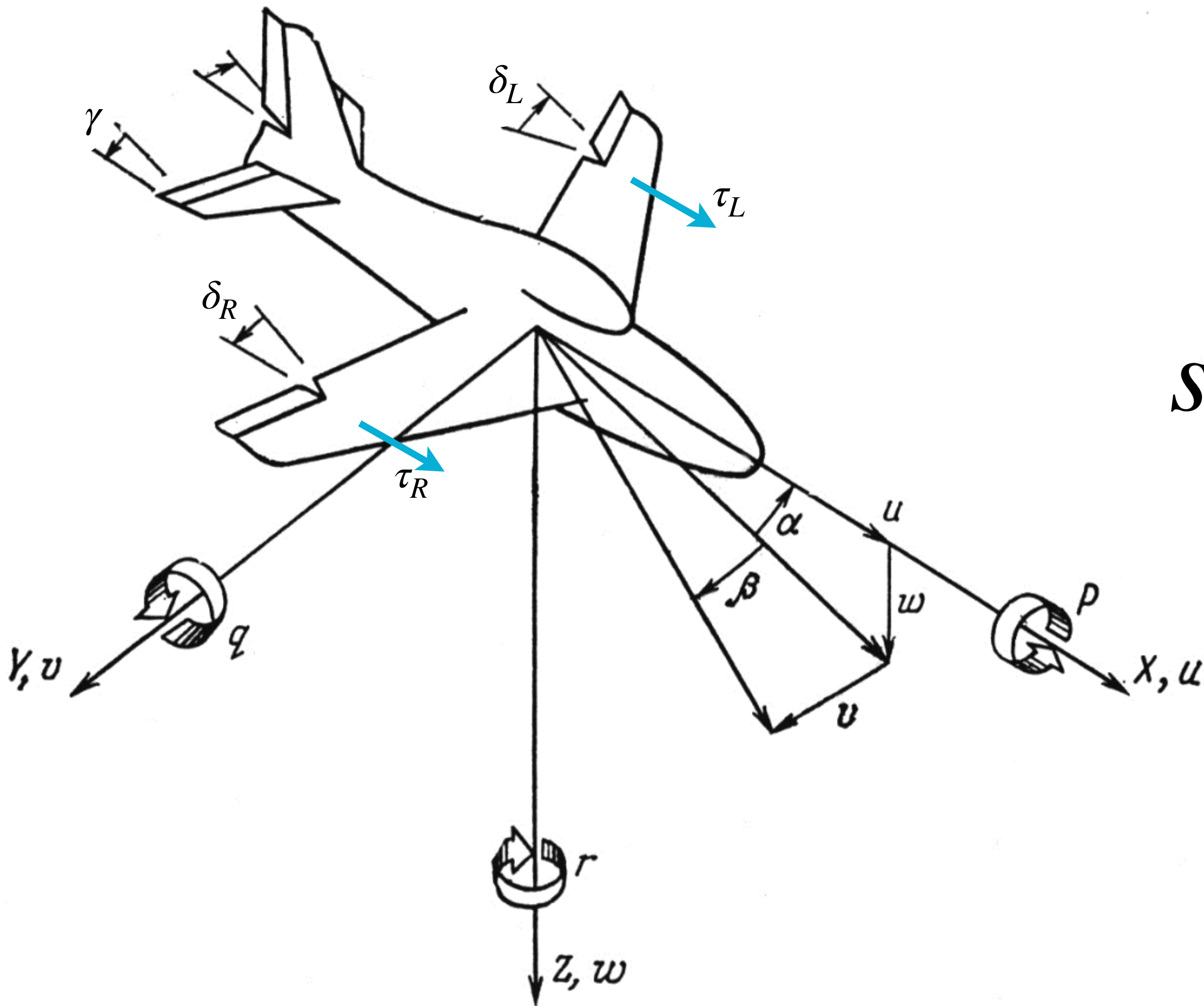


RWTHAACHEN
UNIVERSITY



$$\begin{bmatrix} F_{x,b} \\ F_{y,b} \\ F_{z,b} \\ M_{x,b} \\ M_{y,b} \\ M_{z,b} \end{bmatrix}$$

CFD Simulation Inputs

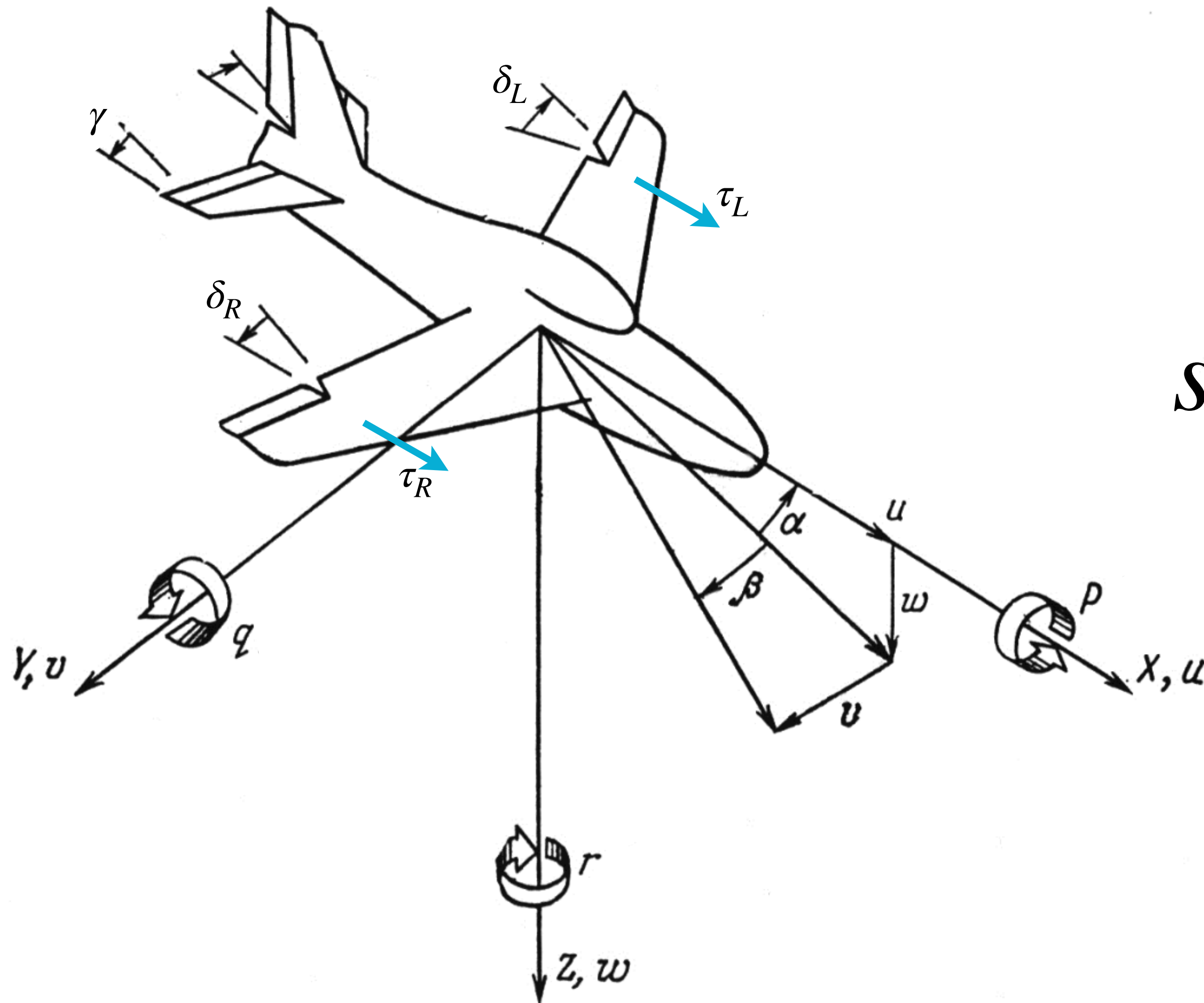


$s =$

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix}$$

$\in \mathbb{R}^{11}$

CFD Simulation Inputs



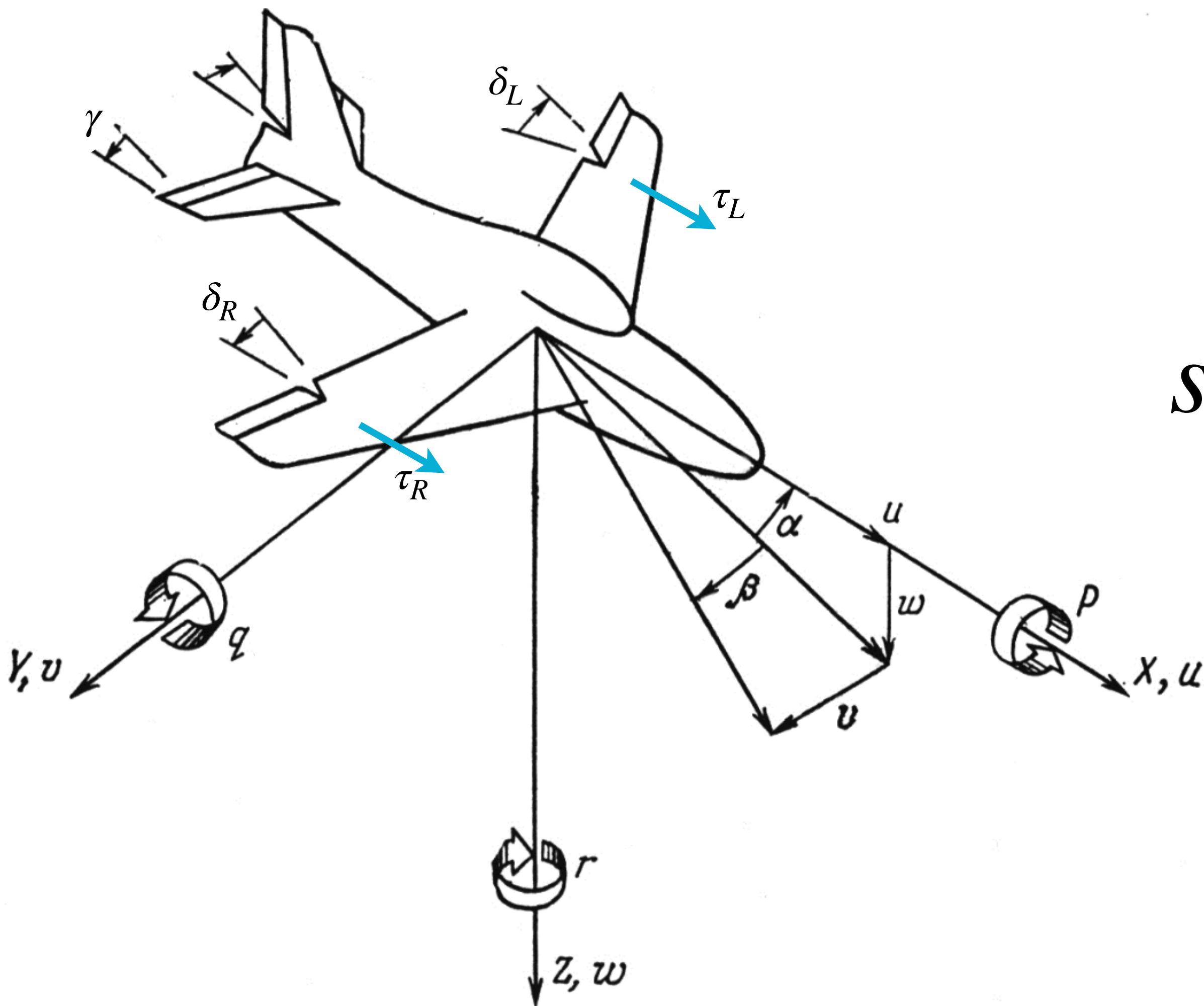
$S =$

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix}$$

$\in \mathbb{R}^{11}$

plane velocity in
NED body frame [m/s]

CFD Simulation Inputs



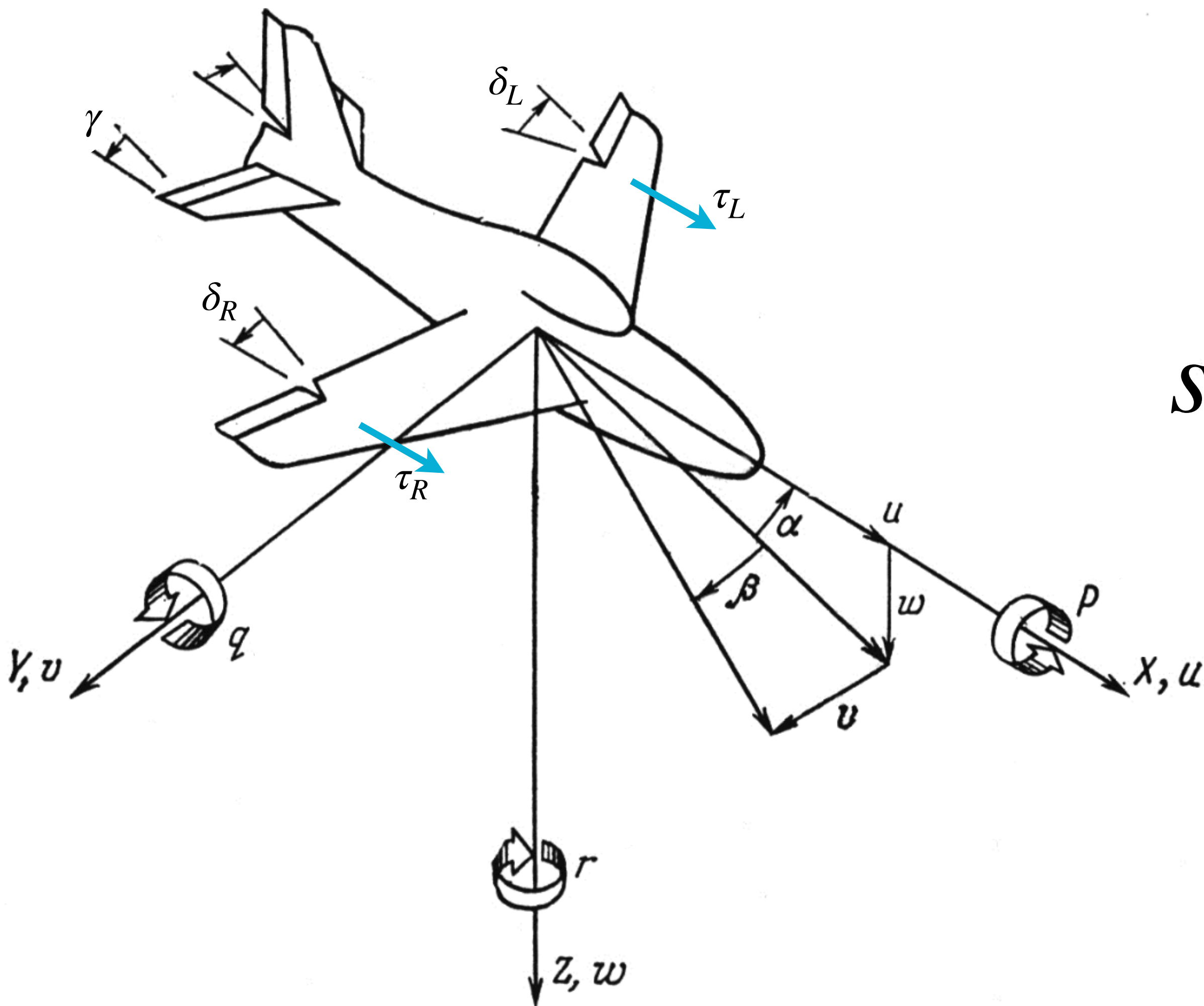
$S =$

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^{11}$$

plane velocity in
NED body frame [m/s]

plane angular velocity in
NED body frame [1/s]

CFD Simulation Inputs



$S =$

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix}$$

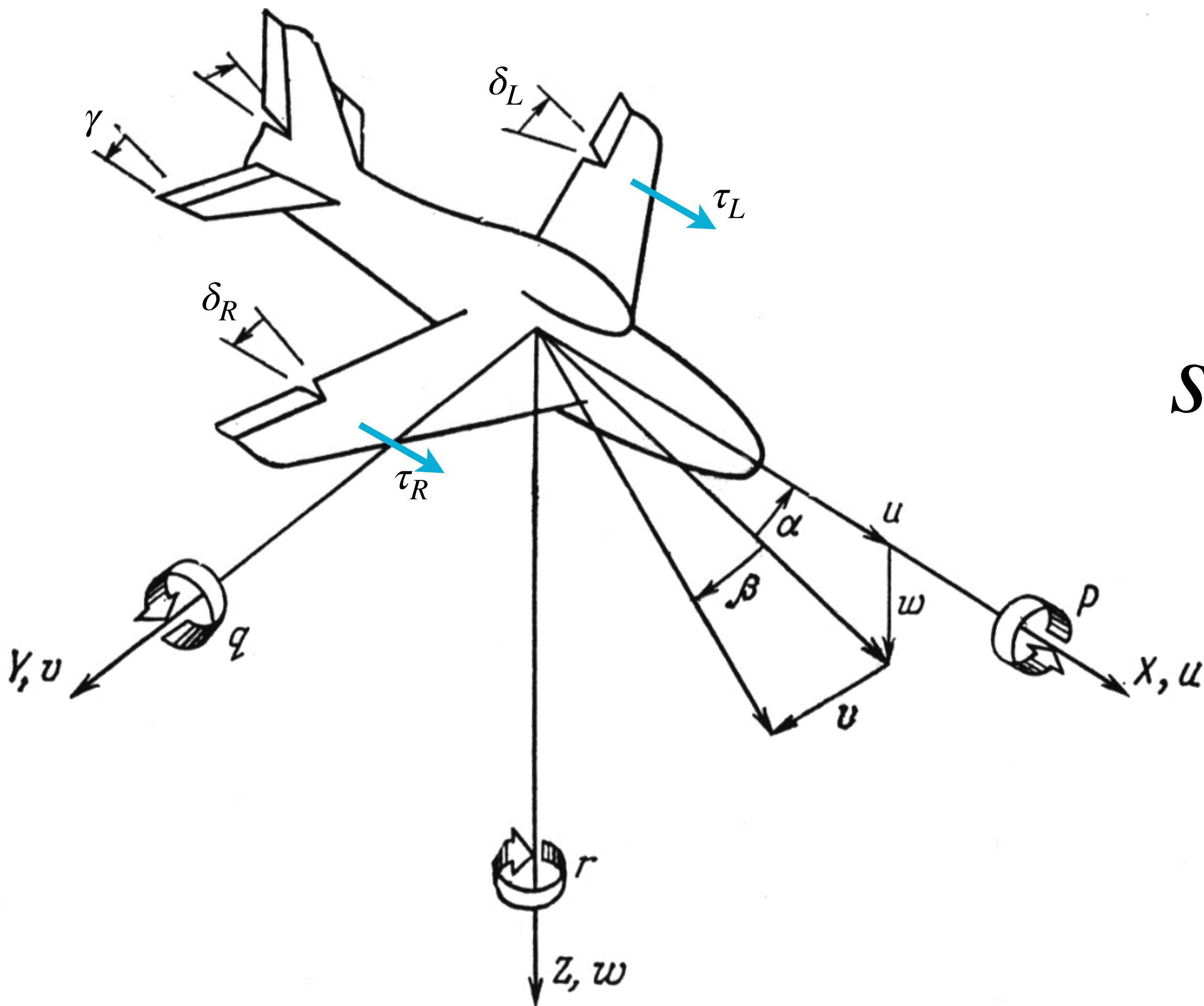
$\in \mathbb{R}^{11}$

plane velocity in
NED body frame [m/s]

plane angular velocity in
NED body frame [1/s]

Propellor thrusts [N]

CFD Simulation Inputs



$$S =$$

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix}$$

$$\in \mathbb{R}^{11}$$

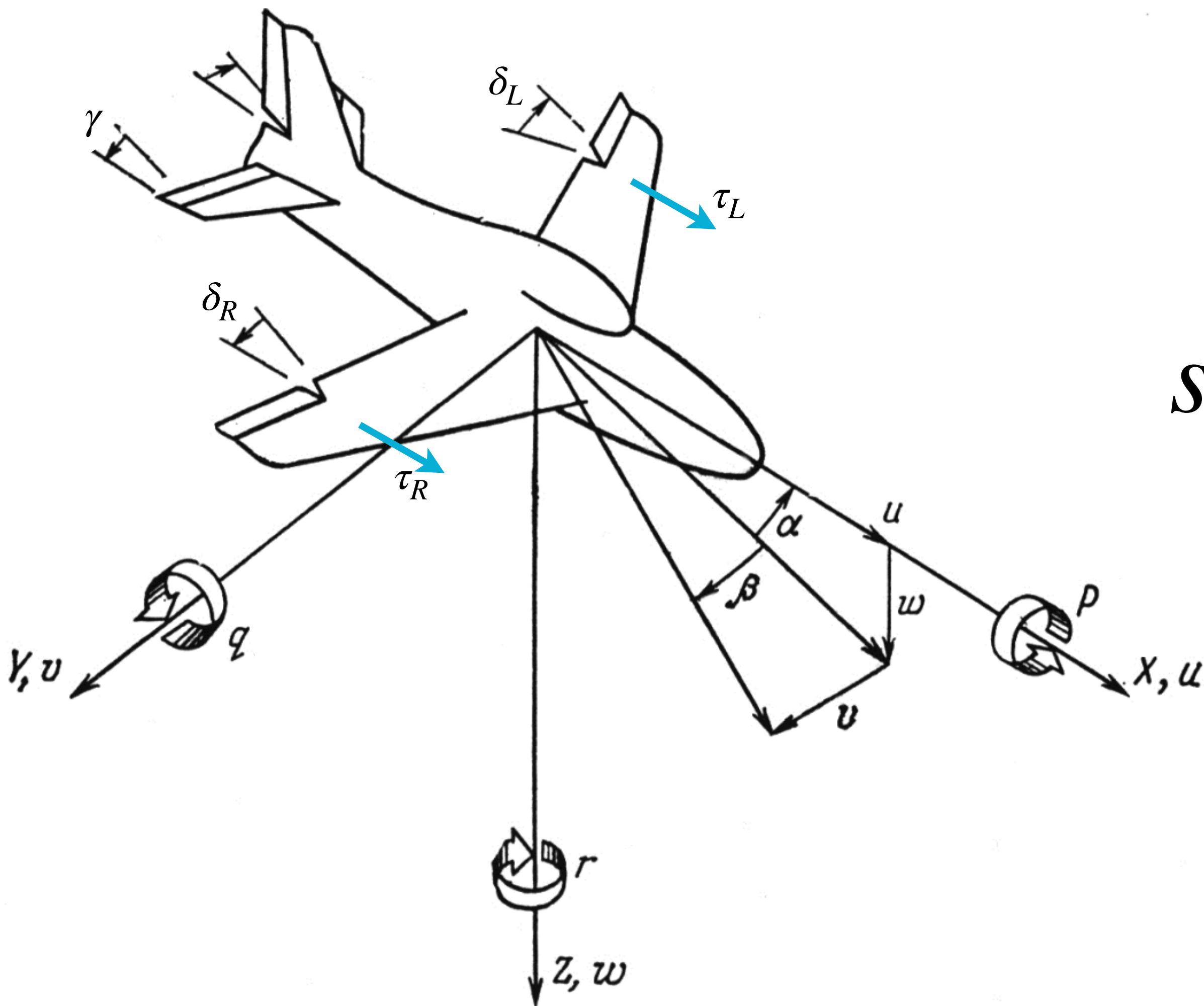
plane velocity in *NED body frame* [m/s]

plane angular velocity in *NED body frame* [1/s]

Propellor thrusts [N]

Aileron Angles

CFD Simulation Inputs



$S =$

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix}$$

$\in \mathbb{R}^{11}$

plane velocity in
NED body frame [m/s]

plane angular velocity in
NED body frame [1/s]

Propellor thrusts [N]

Aileron Angles

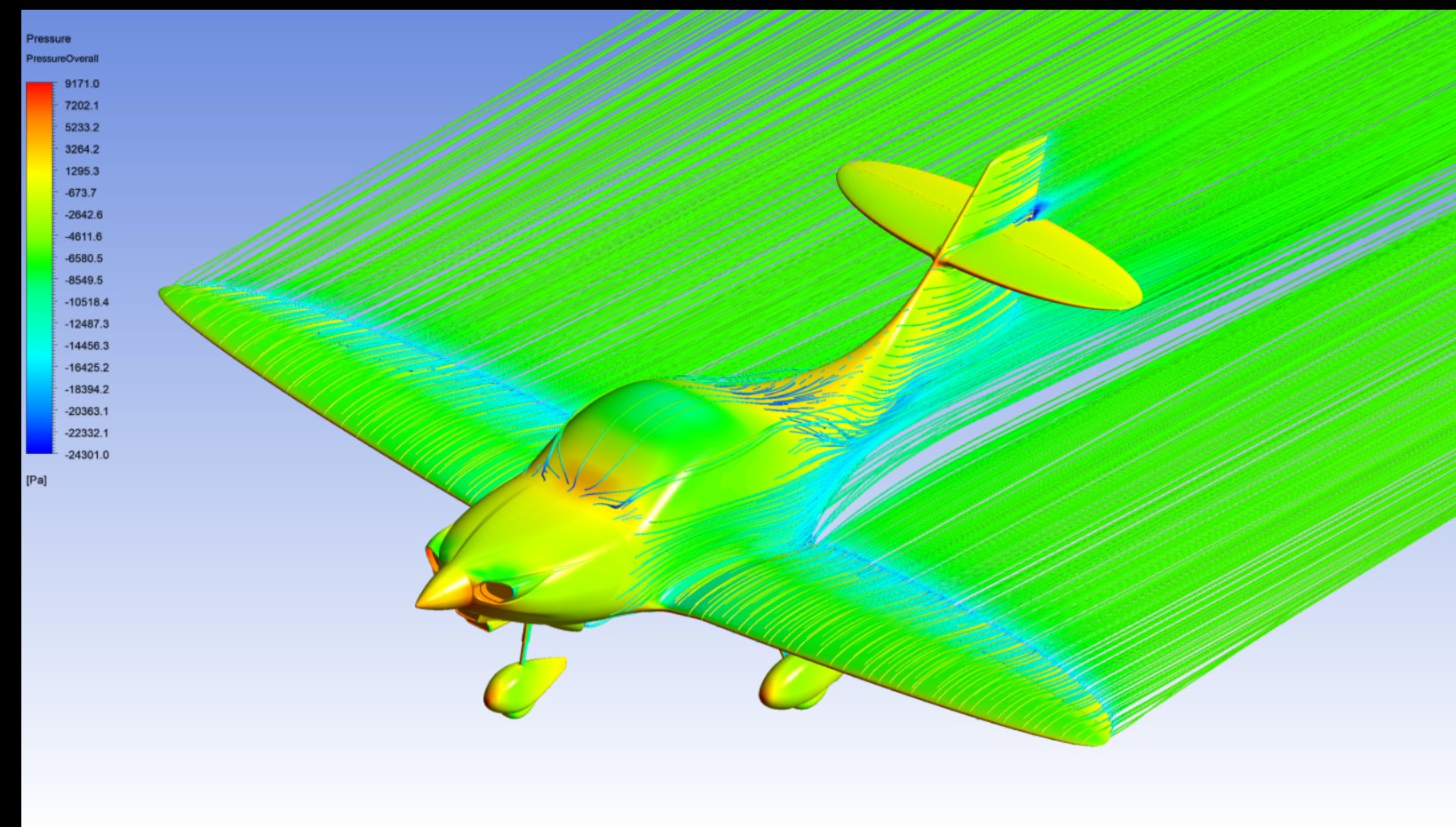
Elevator Angle

Aerodynamic Model

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix}$$



Computational
Fluid Dynamics



RWTHAACHEN
UNIVERSITY



$$\begin{bmatrix} F_{x,b} \\ F_{y,b} \\ F_{z,b} \\ M_{x,b} \\ M_{y,b} \\ M_{z,b} \end{bmatrix}$$

What we do

What we do

What we do

1. RWTH Aachen *samples* the CFD simulation

What we do

1. RWTH Aachen *samples* the CFD simulation
2. We *fit a polynomial* on the data:

$$F_{x,b}(s) \approx \tilde{F}_{x,b}(s) = \phi(s)^\top \theta_{F_x}$$

What we do

1. RWTH Aachen *samples* the CFD simulation
2. We *fit a polynomial* on the data:

$$F_{x,b}(s) \approx \tilde{F}_{x,b}(s) = \phi(s)^\top \theta_{F_x}$$

3. We use the polynomials:

$$F_{\text{aero},b}(s) \approx \tilde{F}_{\text{aero},b}(s) = \begin{bmatrix} \tilde{F}_{x,b}(s) \\ \tilde{F}_{y,b}(s) \\ \tilde{F}_{z,b}(s) \end{bmatrix}$$

$$M_{\text{aero},b}(s) \approx \tilde{M}_{\text{aero},b}(s) = \begin{bmatrix} \tilde{M}_{x,b}(s) \\ \tilde{M}_{y,b}(s) \\ \tilde{M}_{z,b}(s) \end{bmatrix}$$

What we do

1. RWTH Aachen *samples* the CFD simulation
2. We *fit a polynomial* on the data:

$$F_{x,b}(s) \approx \tilde{F}_{x,b}(s) = \phi(s)^\top \theta_{F_x}$$

3. We use the polynomials:

$$F_{\text{aero},b}(s) \approx \tilde{F}_{\text{aero},b}(s) = \begin{bmatrix} \tilde{F}_{x,b}(s) \\ \tilde{F}_{y,b}(s) \\ \tilde{F}_{z,b}(w) \end{bmatrix}$$

$$M_{\text{aero},b}(s) \approx \tilde{M}_{\text{aero},b}(s) = \begin{bmatrix} \tilde{M}_{x,b}(s) \\ \tilde{M}_{y,b}(s) \\ \tilde{M}_{z,b}(s) \end{bmatrix}$$

4. (Profit)

Estimating Aerodynamic Model from Data

Sampling the CFD Simulation

Input	Sampling Range	Unit
u	{12, 13, 14, 15, 16, 17, 18}	m/s

Sampling the CFD Simulation

Input	Sampling Range	Unit
u	{12, 13, 14, 15, 16, 17, 18}	m/s
v	{-5, -2.5, 0, 2.5, 5}	m/s

Sampling the CFD Simulation

Input	Sampling Range	Unit
u	{12, 13, 14, 15, 16, 17, 18}	m/s
v	{−5, −2.5, 0, 2.5, 5}	m/s
w	{−5, −2.5, 0, 2.5, 5}	m/s

Sampling the CFD Simulation

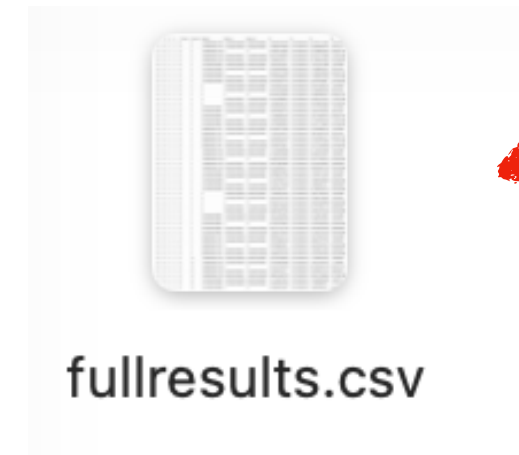
Input	Sampling Range	Unit
u	{12, 13, 14, 15, 16, 17, 18}	m/s
v	{-5, -2.5, 0, 2.5, 5}	m/s
w	{-5, -2.5, 0, 2.5, 5}	m/s
p	{-1, -0.5, 0, 0.5, 1}	rad/s
q	{-1, -0.5, 0, 0.5, 1}	rad/s
r	{-1, -0.5, 0, 0.5, 1}	rad/s
τ_L	{0, 5, 10, 15, 20}	N
τ_R	{0, 5, 10, 15, 20}	N
δ_L	{-0.44, -0.22, 0, 0.22, 0.44}	rad
δ_R	{-0.44, -0.22, 0, 0.22, 0.44}	rad
γ	{-0.56, -0.28, 0, 0.28, 0.56}	rad

We receive:



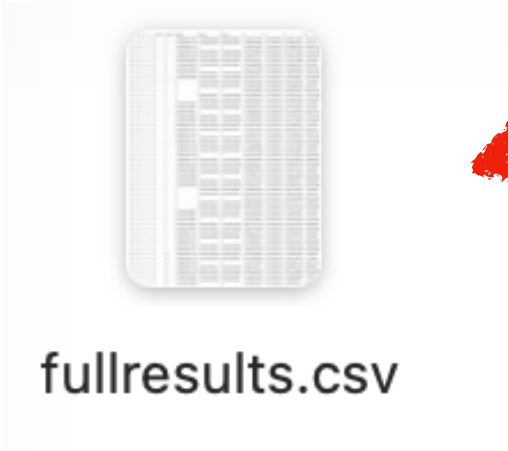
fullresults.csv

We receive:



≈ 2.5 GB

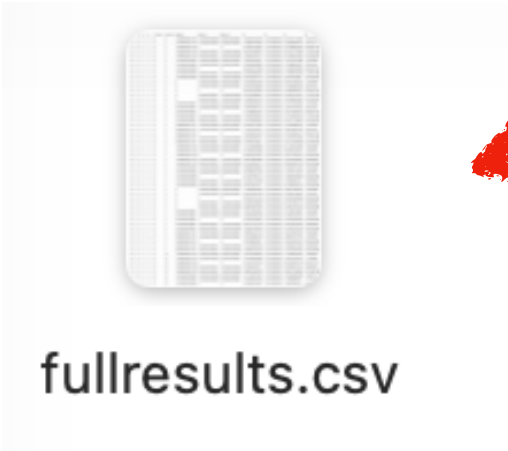
We receive:



≈ 2.5 GB

U	V	W	p	q	r	Thrust_L	Thrust_R	Elevator	Aileron_L	Aileron_R	Fx	Fy	Fz	Mx	My	Mz
12	-5	-5	-1	-1	-1	0	0	-0.5585	-0.4363	0.4363	-7.5334	1.2727	18.1376	-1.5676	3.0999	0.4545
12	-5	-5	-1	-1	-1	0	0	-0.5585	-0.2182	0.2182	-7.0389	1.2727	18.0569	-0.9201	3.0815	0.3123
12	-5	-5	-1	-1	-1	0	0	-0.5585	0	0	-6.7569	1.2727	18.1919	0.3935	3.0675	-0.0757
12	-5	-5	-1	-1	-1	0	0	-0.5585	0.2182	-0.2182	-6.8273	1.2727	18.4724	1.7131	3.0749	-0.4551
12	-5	-5	-1	-1	-1	0	0	-0.5585	0.4363	-0.4363	-7.2508	1.2727	18.7708	2.3767	3.0849	-0.5738
12	-5	-5	-1	-1	-1	0	0	-0.2793	-0.4363	0.4363	-7.2801	1.2727	17.4247	-1.5676	2.7258	0.4545
12	-5	-5	-1	-1	-1	0	0	-0.2793	-0.2182	0.2182	-6.7855	1.2727	17.3440	-0.9201	2.7074	0.3123
12	-5	-5	-1	-1	-1	0	0	-0.2793	0	0	-6.5036	1.2727	17.4790	0.3935	2.6934	-0.0757
12	-5	-5	-1	-1	-1	0	0	-0.2793	0.2182	-0.2182	-6.5740	1.2727	17.7595	1.7131	2.7008	-0.4551
12	-5	-5	-1	-1	-1	0	0	-0.2793	0.4363	-0.4363	-6.9975	1.2727	18.0580	2.3767	2.7107	-0.5738
12	-5	-5	-1	-1	-1	0	0	0	-0.4363	0.4363	-7.4676	1.2727	15.8746	-1.5676	1.9368	0.4545
12	-5	-5	-1	-1	-1	0	0	0	-0.2182	0.2182	-6.9730	1.2727	15.7938	-0.9201	1.9185	0.3123
12	-5	-5	-1	-1	-1	0	0	0	0	0	-6.6910	1.2727	15.9288	0.3935	1.9045	-0.0757
12	-5	-5	-1	-1	-1	0	0	0	0.2182	-0.2182	-6.7614	1.2727	16.2093	1.7131	1.9118	-0.4551
12	-5	-5	-1	-1	-1	0	0	0	0.4363	-0.4363	-7.1849	1.2727	16.5078	2.3767	1.9218	-0.5738
12	-5	-5	-1	-1	-1	0	0	0.2793	-0.4363	0.4363	-7.9044	1.2727	14.4375	-1.5676	1.2142	0.4545
12	-5	-5	-1	-1	-1	0	0	0.2793	-0.2182	0.2182	-7.4098	1.2727	14.3567	-0.9201	1.1958	0.3123
12	-5	-5	-1	-1	-1	0	0	0.2793	0	0	-7.1279	1.2727	14.4917	0.3935	1.1818	-0.0757

We receive:

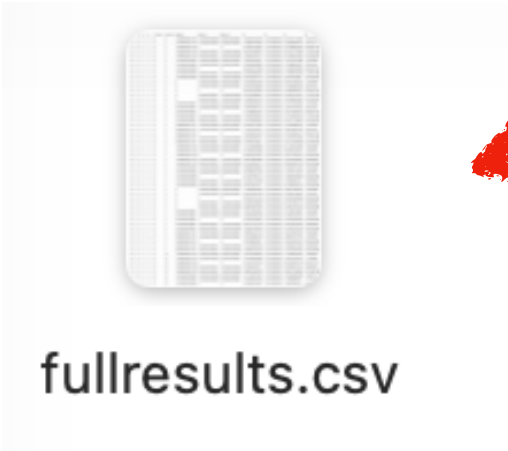


≈ 2.5 GB

Inputs

U	V	W	p	q	r	Thrust_L	Thrust_R	Elevator	Aileron_L	Aileron_R	Fx	Fy	Fz	Mx	My	Mz
12	-5	-5	-1	-1	-1	0	0	-0.5585	-0.4363	0.4363	-7.5334	1.2727	18.1376	-1.5676	3.0999	0.4545
12	-5	-5	-1	-1	-1	0	0	-0.5585	-0.2182	0.2182	-7.0389	1.2727	18.0569	-0.9201	3.0815	0.3123
12	-5	-5	-1	-1	-1	0	0	-0.5585	0	0	-6.7569	1.2727	18.1919	0.3935	3.0675	-0.0757
12	-5	-5	-1	-1	-1	0	0	-0.5585	0.2182	-0.2182	-6.8273	1.2727	18.4724	1.7131	3.0749	-0.4551
12	-5	-5	-1	-1	-1	0	0	-0.5585	0.4363	-0.4363	-7.2508	1.2727	18.7708	2.3767	3.0849	-0.5738
12	-5	-5	-1	-1	-1	0	0	-0.2793	-0.4363	0.4363	-7.2801	1.2727	17.4247	-1.5676	2.7258	0.4545
12	-5	-5	-1	-1	-1	0	0	-0.2793	-0.2182	0.2182	-6.7855	1.2727	17.3440	-0.9201	2.7074	0.3123
12	-5	-5	-1	-1	-1	0	0	-0.2793	0	0	-6.5036	1.2727	17.4790	0.3935	2.6934	-0.0757
12	-5	-5	-1	-1	-1	0	0	-0.2793	0.2182	-0.2182	-6.5740	1.2727	17.7595	1.7131	2.7008	-0.4551
12	-5	-5	-1	-1	-1	0	0	-0.2793	0.4363	-0.4363	-6.9975	1.2727	18.0580	2.3767	2.7107	-0.5738
12	-5	-5	-1	-1	-1	0	0	0	-0.4363	0.4363	-7.4676	1.2727	15.8746	-1.5676	1.9368	0.4545
12	-5	-5	-1	-1	-1	0	0	0	-0.2182	0.2182	-6.9730	1.2727	15.7938	-0.9201	1.9185	0.3123
12	-5	-5	-1	-1	-1	0	0	0	0	0	-6.6910	1.2727	15.9288	0.3935	1.9045	-0.0757
12	-5	-5	-1	-1	-1	0	0	0	0.2182	-0.2182	-6.7614	1.2727	16.2093	1.7131	1.9118	-0.4551
12	-5	-5	-1	-1	-1	0	0	0	0.4363	-0.4363	-7.1849	1.2727	16.5078	2.3767	1.9218	-0.5738
12	-5	-5	-1	-1	-1	0	0	0.2793	-0.4363	0.4363	-7.9044	1.2727	14.4375	-1.5676	1.2142	0.4545
12	-5	-5	-1	-1	-1	0	0	0.2793	-0.2182	0.2182	-7.4098	1.2727	14.3567	-0.9201	1.1958	0.3123
12	-5	-5	-1	-1	-1	0	0	0.2793	0	0	-7.1279	1.2727	14.4917	0.3935	1.1818	-0.0757

We receive:



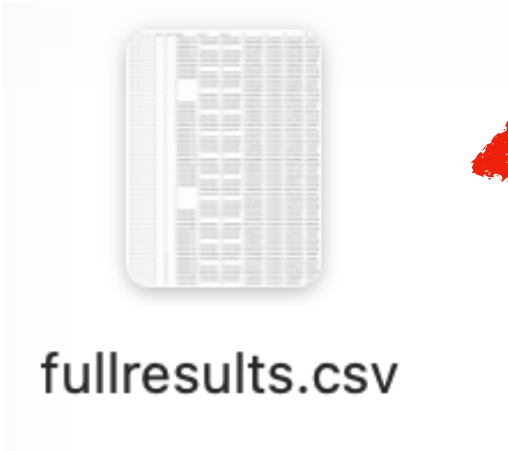
≈ 2.5 GB

Inputs

Outputs

U	V	W	p	q	r	Thrust_L	Thrust_R	Elevator	Aileron_L	Aileron_R	Fx	Fy	Fz	Mx	My	Mz
12	-5	-5	-1	-1	-1	0	0	-0.5585	-0.4363	0.4363	-7.5334	1.2727	18.1376	-1.5676	3.0999	0.4545
12	-5	-5	-1	-1	-1	0	0	-0.5585	-0.2182	0.2182	-7.0389	1.2727	18.0569	-0.9201	3.0815	0.3123
12	-5	-5	-1	-1	-1	0	0	-0.5585	0	0	-6.7569	1.2727	18.1919	0.3935	3.0675	-0.0757
12	-5	-5	-1	-1	-1	0	0	-0.5585	0.2182	-0.2182	-6.8273	1.2727	18.4724	1.7131	3.0749	-0.4551
12	-5	-5	-1	-1	-1	0	0	-0.5585	0.4363	-0.4363	-7.2508	1.2727	18.7708	2.3767	3.0849	-0.5738
12	-5	-5	-1	-1	-1	0	0	-0.2793	-0.4363	0.4363	-7.2801	1.2727	17.4247	-1.5676	2.7258	0.4545
12	-5	-5	-1	-1	-1	0	0	-0.2793	-0.2182	0.2182	-6.7855	1.2727	17.3440	-0.9201	2.7074	0.3123
12	-5	-5	-1	-1	-1	0	0	-0.2793	0	0	-6.5036	1.2727	17.4790	0.3935	2.6934	-0.0757
12	-5	-5	-1	-1	-1	0	0	-0.2793	0.2182	-0.2182	-6.5740	1.2727	17.7595	1.7131	2.7008	-0.4551
12	-5	-5	-1	-1	-1	0	0	-0.2793	0.4363	-0.4363	-6.9975	1.2727	18.0580	2.3767	2.7107	-0.5738
12	-5	-5	-1	-1	-1	0	0	0	-0.4363	0.4363	-7.4676	1.2727	15.8746	-1.5676	1.9368	0.4545
12	-5	-5	-1	-1	-1	0	0	0	-0.2182	0.2182	-6.9730	1.2727	15.7938	-0.9201	1.9185	0.3123
12	-5	-5	-1	-1	-1	0	0	0	0	0	-6.6910	1.2727	15.9288	0.3935	1.9045	-0.0757
12	-5	-5	-1	-1	-1	0	0	0	0.2182	-0.2182	-6.7614	1.2727	16.2093	1.7131	1.9118	-0.4551
12	-5	-5	-1	-1	-1	0	0	0	0.4363	-0.4363	-7.1849	1.2727	16.5078	2.3767	1.9218	-0.5738
12	-5	-5	-1	-1	-1	0	0	0.2793	-0.4363	0.4363	-7.9044	1.2727	14.4375	-1.5676	1.2142	0.4545
12	-5	-5	-1	-1	-1	0	0	0.2793	-0.2182	0.2182	-7.4098	1.2727	14.3567	-0.9201	1.1958	0.3123
12	-5	-5	-1	-1	-1	0	0	0.2793	0	0	-7.1279	1.2727	14.4917	0.3935	1.1818	-0.0757

We receive:



≈ 2.5 GB

Inputs

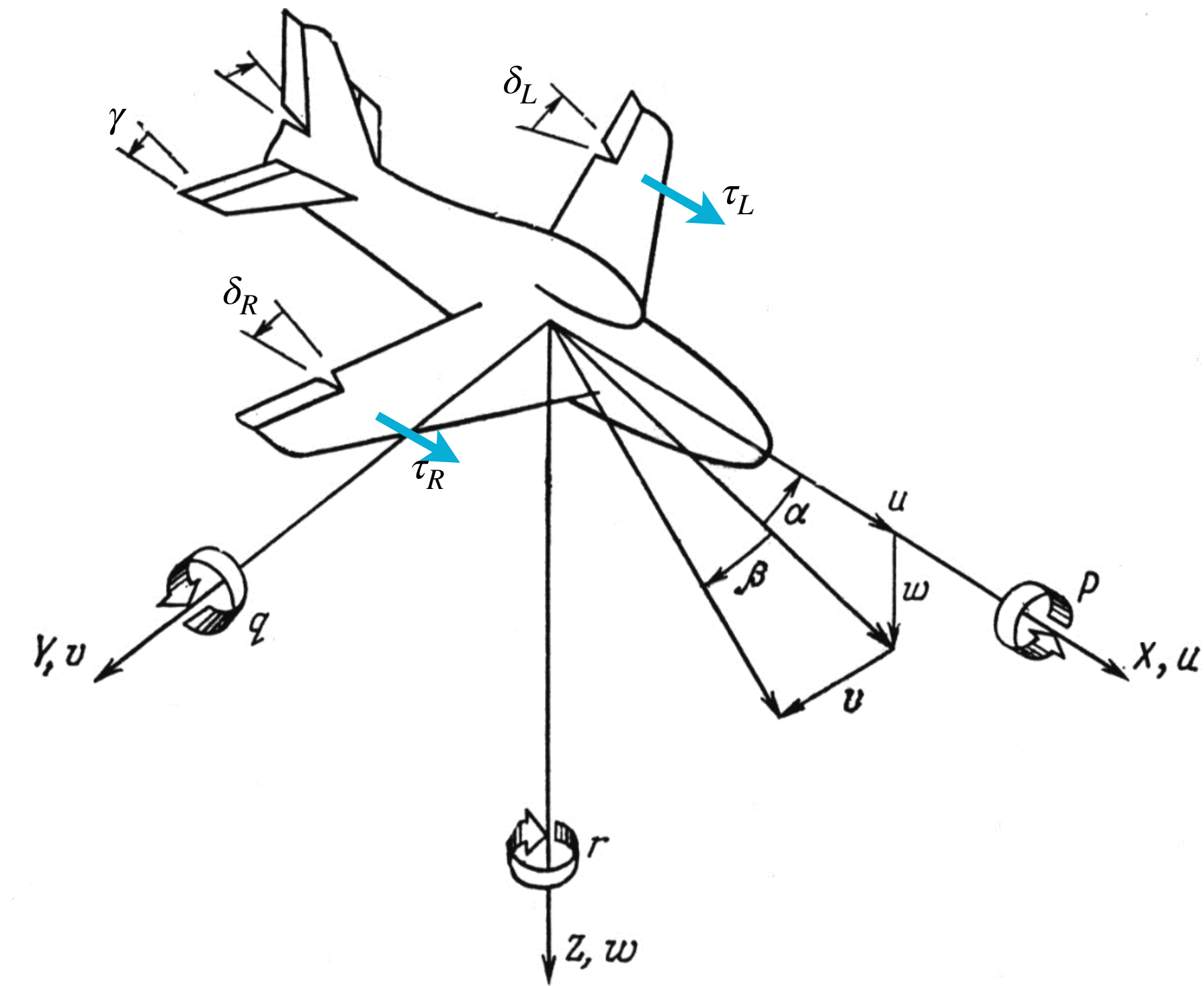
Outputs

U	V	W	p	q	r	Thrust_L	Thrust_R	Elevator	Aileron_L	Aileron_R	Fx	Fy	Fz	Mx	My	Mz
12	-5	-5	-1	-1	-1	0	0	-0.5585	-0.4363	0.4363	-7.5334	1.2727	18.1376	-1.5676	3.0999	0.4545
12	-5	-5	-1	-1	-1	0	0	-0.5585	-0.2182	0.2182	-7.0389	1.2727	18.0569	-0.9201	3.0815	0.3123
12	-5	-5	-1	-1	-1	0	0	-0.5585	0	0	-6.7569	1.2727	18.1919	0.3935	3.0675	-0.0757
12	-5	-5	-1	-1	-1	0	0	-0.5585	0.2182	-0.2182	-6.8273	1.2727	18.4724	1.7131	3.0749	-0.4551
12	-5	-5	-1	-1	-1	0	0	-0.5585	0.4363	-0.4363	-7.2508	1.2727	18.7708	2.3767	3.0849	-0.5738
12	-5	-5	-1	-1	-1	0	0	-0.2793	-0.4363	0.4363	-7.2801	1.2727	17.4247	-1.5676	2.7258	0.4545
12	-5	-5	-1	-1	-1	0	0	-0.2793	-0.2182	0.2182	-6.7855	1.2727	17.3440	-0.9201	2.7074	0.3123
12	-5	-5	-1	-1	-1	0	0	-0.2793	0	0	-6.5036	1.2727	17.4790	0.3935	2.6934	-0.0757
12	-5	-5	-1	-1	-1	0	0	-0.2793	0.2182	-0.2182	-6.5740	1.2727	17.7595	1.7131	2.7008	-0.4551
12	-5	-5	-1	-1	-1	0	0	-0.2793	0.4363	-0.4363	-6.9975	1.2727	18.0580	2.3767	2.7107	-0.5738
12	-5	-5	-1	-1	-1	0	0	0	-0.4363	0.4363	-7.4676	1.2727	15.8746	-1.5676	1.9368	0.4545
12	-5	-5	-1	-1	-1	0	0	0	-0.2182	0.2182	-6.9730	1.2727	15.7938	-0.9201	1.9185	0.3123
12	-5	-5	-1	-1	-1	0	0	0	0	0	-6.6910	1.2727	15.9288	0.3935	1.9045	-0.0757
12	-5	-5	-1	-1	-1	0	0	0	0.2182	-0.2182	-6.7614	1.2727	16.2093	1.7131	1.9118	-0.4551
12	-5	-5	-1	-1	-1	0	0	0	0.4363	-0.4363	-7.1849	1.2727	16.5078	2.3767	1.9218	-0.5738
12	-5	-5	-1	-1	-1	0	0	0.2793	-0.4363	0.4363	-7.9044	1.2727	14.4375	-1.5676	1.2142	0.4545
12	-5	-5	-1	-1	-1	0	0	0.2793	-0.2182	0.2182	-7.4098	1.2727	14.3567	-0.9201	1.1958	0.3123
12	-5	-5	-1	-1	-1	0	0	0.2793	0	0	-7.1279	1.2727	14.4917	0.3935	1.1818	-0.0757

13 671 875 Lines

Fitting a Polynomial

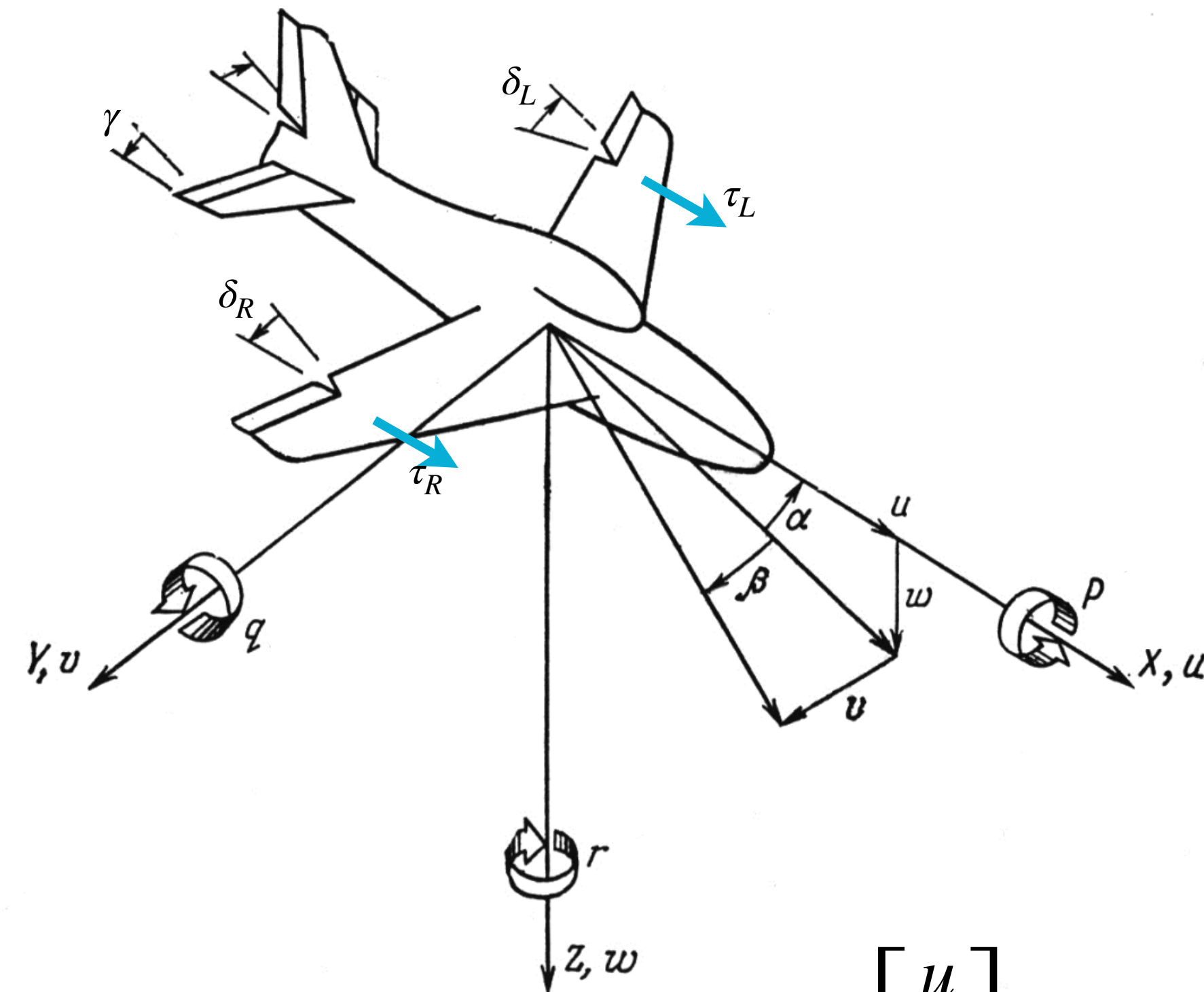
1. Regressor Selection



Fitting a Polynomial

1. Regressor Selection

$$\tilde{F}_x(s) = \phi(s)^T \theta$$

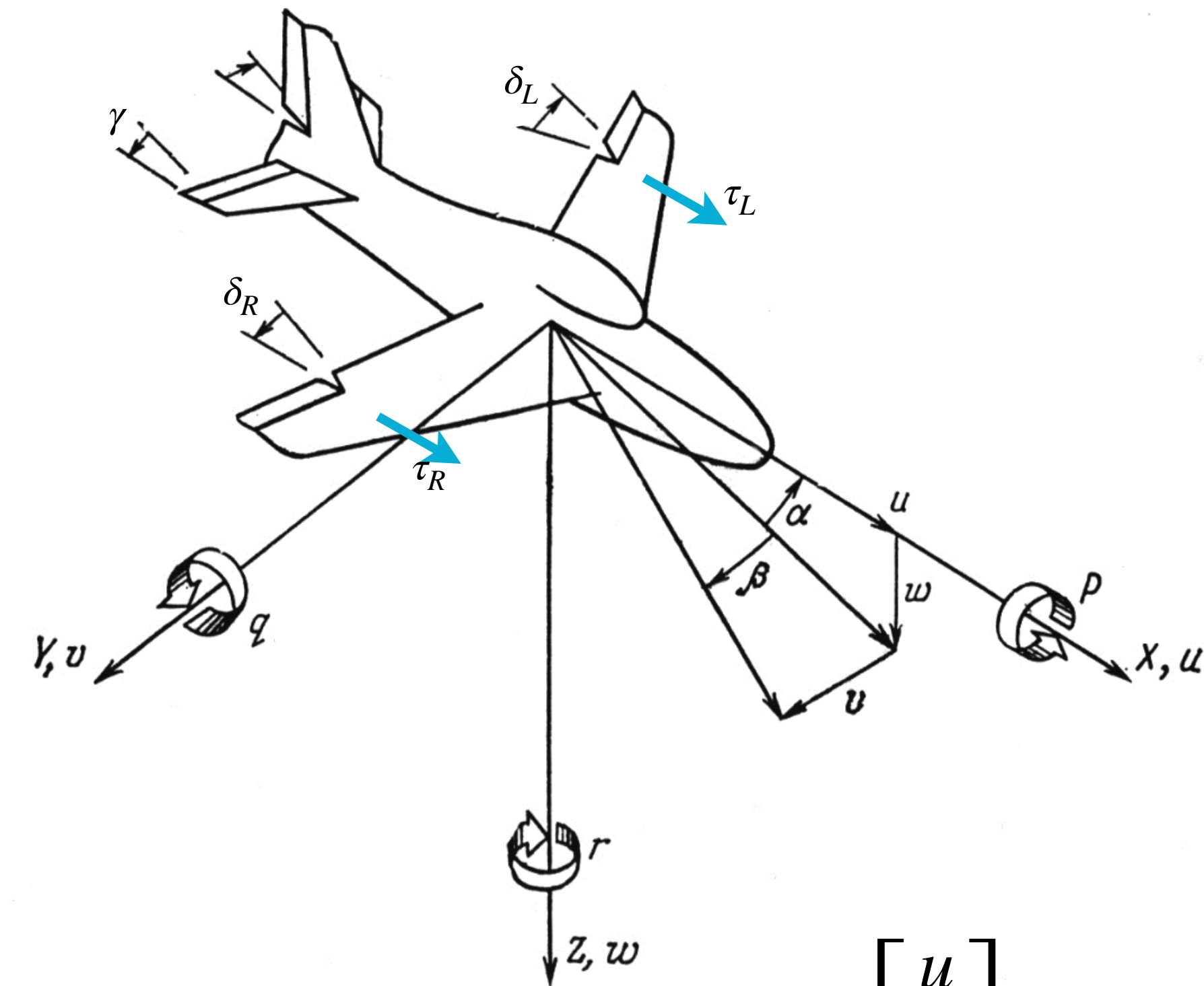


$$s = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^{11}$$

Fitting a Polynomial

1. Regressor Selection

$$\begin{aligned}\tilde{F}_x(s) &= \phi(s)^\top \theta \\ &= \theta_1 u + \theta_2 u^2 + \theta_3 u^3 + \theta_4 v + \theta_5 v^2 + \dots + \theta_{32} \beta^3 + \theta_{33}\end{aligned}$$



$$s = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^{11}$$

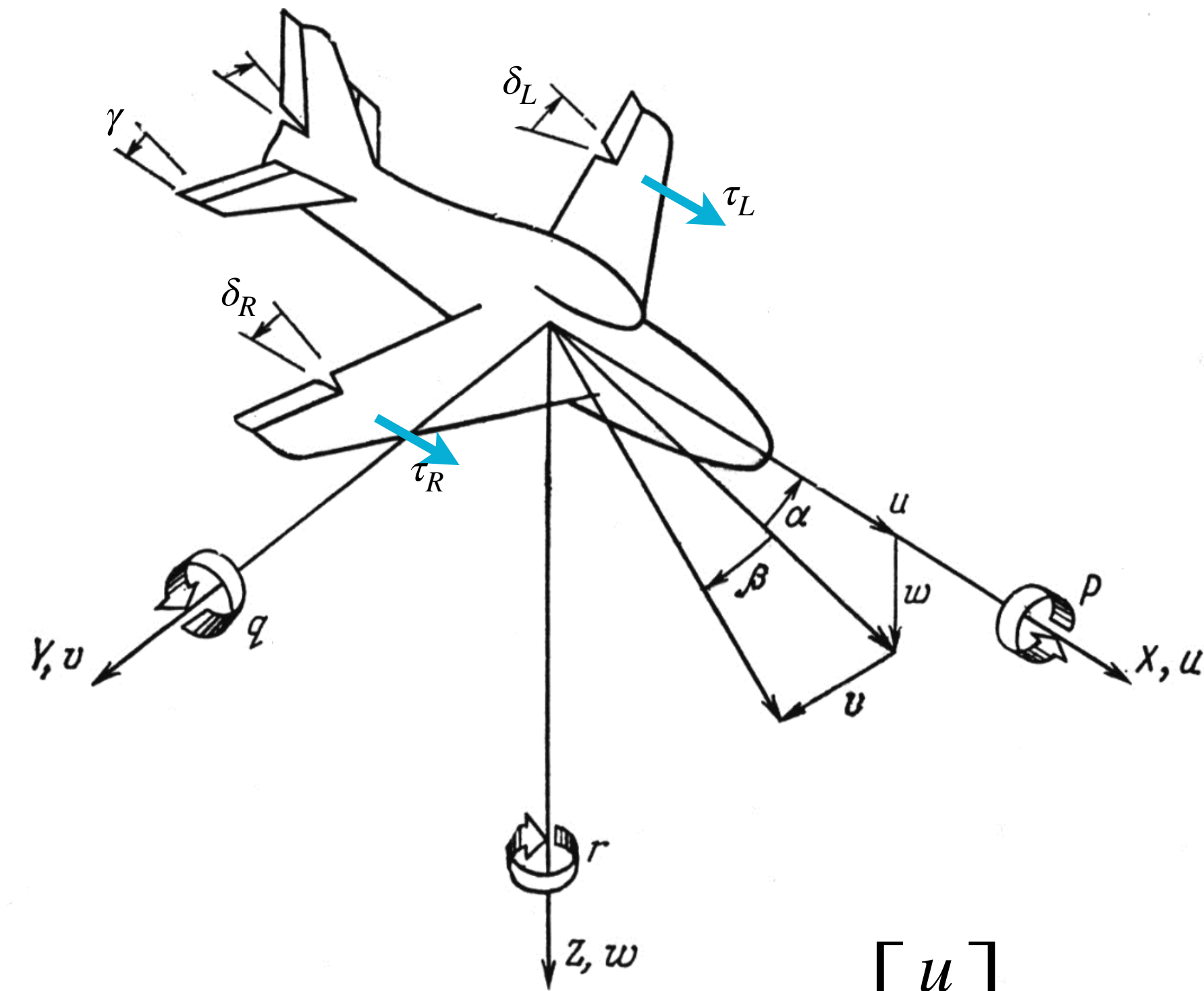
Fitting a Polynomial

1. Regressor Selection

$$\begin{aligned}\tilde{F}_x(s) &= \phi(s)^\top \theta \\ &= \theta_1 u + \theta_2 u^2 + \theta_3 u^3 + \theta_4 v + \theta_5 v^2 + \dots + \theta_{32} \beta^3 + \theta_{33}\end{aligned}$$

$$\phi(s) = \begin{bmatrix} u \\ u^2 \\ u^3 \\ v \\ v^2 \\ \vdots \\ \beta^3 \\ 1 \end{bmatrix} \in \mathbb{R}^{33}$$

$$s = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^{11}$$



Fitting a Polynomial

1. Regressor Selection

$$\begin{aligned}\tilde{F}_x(s) &= \phi(s)^\top \theta \\ &= \theta_1 u + \theta_2 u^2 + \theta_3 u^3 + \theta_4 v + \theta_5 v^2 + \dots + \theta_{32} \beta^3 + \theta_{33}\end{aligned}$$

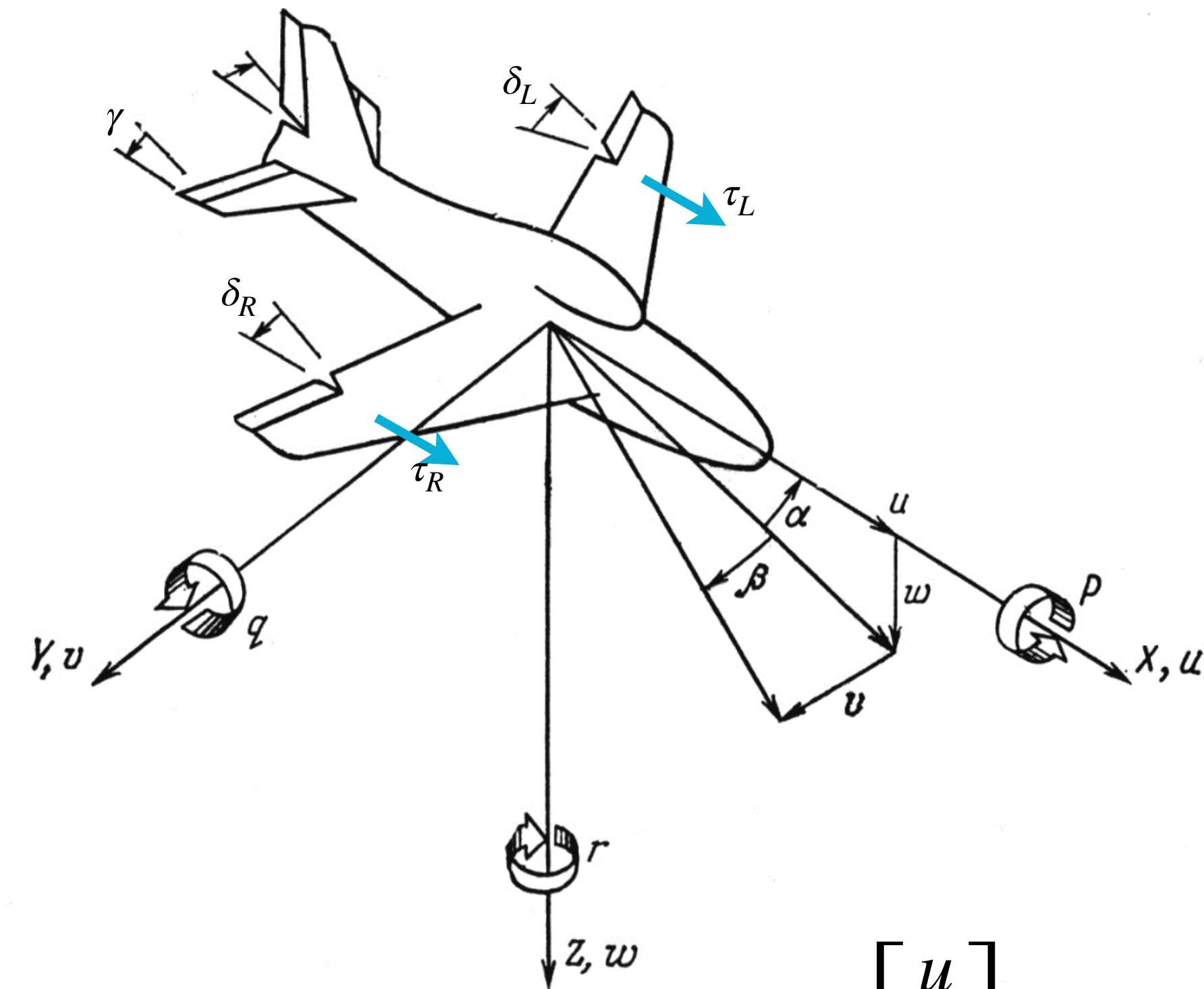
$$\phi(s) = \begin{bmatrix} u \\ u^2 \\ u^3 \\ v \\ v^2 \\ \vdots \\ \beta^3 \\ 1 \end{bmatrix} \in \mathbb{R}^{33}$$

Expert Knowledge

use 'angle of attack' as regressor

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right) \approx \frac{w}{u}$$

$$s = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^{11}$$



Fitting a Polynomial

1. Regressor Selection

$$\begin{aligned}\tilde{F}_x(s) &= \phi(s)^\top \theta \\ &= \theta_1 u + \theta_2 u^2 + \theta_3 u^3 + \theta_4 v + \theta_5 v^2 + \dots + \theta_{32} \beta^3 + \theta_{33}\end{aligned}$$

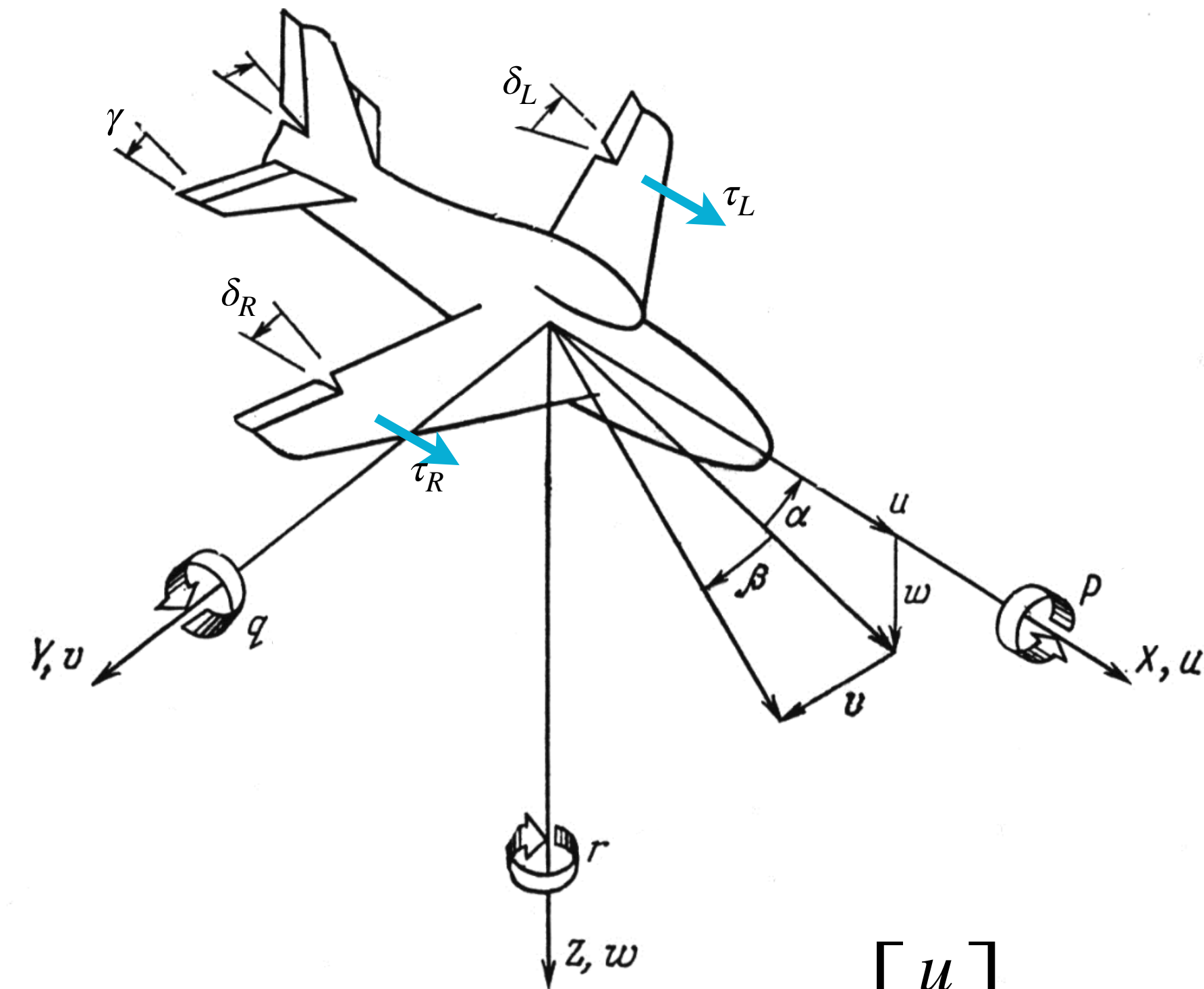
$$\phi(s) = \begin{bmatrix} u \\ \vdots \\ w/u \\ (w/u)^2 \\ (w/u)^3 \\ \vdots \\ \beta^3 \\ 1 \end{bmatrix} \in \mathbb{R}^{36}$$

Expert Knowledge

use 'angle of attack' as regressor

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right) \approx \frac{w}{u}$$

$$s = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \tau_L \\ \tau_R \\ \delta_L \\ \delta_R \\ \gamma \end{bmatrix} \in \mathbb{R}^{11}$$



$$N = 13\,671\,875$$

$$d = 33$$

Fitting a Polynomial

2. Least-Squares Fitting Problem

$$N = 13\,671\,875$$

$$d = 33$$

Fitting a Polynomial

2. Least-Squares Fitting Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^N \left(\tilde{F}_x(s_i) - F_{x,i} \right)^2$$

$$N = 13\,671\,875$$

$$d = 33$$

Fitting a Polynomial

2. Least-Squares Fitting Problem

$$\begin{aligned}\theta^* &= \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^N \left(\tilde{F}_x(s_i) - F_{x,i} \right)^2 \\ &= \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^N \left(\phi(s_i)^\top \theta - F_{x,i} \right)^2\end{aligned}$$

$$N = 13\,671\,875$$
$$d = 33$$

Fitting a Polynomial

2. Least-Squares Fitting Problem

$$\begin{aligned}\theta^* &= \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^N \left(\tilde{F}_x(s_i) - F_{x,i} \right)^2 \\ &= \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^N \left(\phi(s_i)^\top \theta - F_{x,i} \right)^2 \\ &= \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \left\| \Phi^\top \theta - y \right\|_2^2\end{aligned}$$

$$N = 13\,671\,875$$
$$d = 33$$

Fitting a Polynomial

2. Least-Squares Fitting Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^N \left(\tilde{F}_x(s_i) - F_{x,i} \right)^2$$

$$= \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^N \left(\phi(s_i)^\top \theta - F_{x,i} \right)^2$$

$$= \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \left\| \Phi^\top \theta - y \right\|_2^2$$

$$= (\Phi^\top \Phi)^{-1} \Phi^\top y$$

$$N = 13\,671\,875$$

$$d = 36$$

Computational Complexity

$$\theta^* = (\Phi^T \Phi)^{-1} \Phi^T y$$

$$N = 13\,671\,875$$
$$d = 36$$

Computational Complexity

$$\theta^* = (\Phi^T \Phi)^{-1} \Phi^T y$$

Q: How long does this takes to compute?
1ms vs. 1s vs. 1000s

$$N = 13\,671\,875$$

$$d = 36$$

Computational Complexity

$$\theta^* = (\Phi^T \Phi)^{-1} \Phi^T y$$

$$N = 13\,671\,875$$
$$d = 36$$

Computational Complexity

$$\theta^* = (\Phi^\top \Phi)^{-1} \Phi^\top y$$

(A) Compute $\Phi^\top \Phi$

$$N = 13\,671\,875$$
$$d = 36$$

Computational Complexity

$$\theta^* = (\Phi^\top \Phi)^{-1} \Phi^\top y$$

(A) Compute $\Phi^\top \Phi$

(B) Compute $\Phi^\top y$

$$N = 13\,671\,875$$
$$d = 36$$

Computational Complexity

$$\theta^* = (\Phi^T \Phi)^{-1} \Phi^T y$$

(A) Compute $\Phi^T \Phi$

(B) Compute $\Phi^T y$

(C) Solve $(\Phi^T \Phi)\theta = \Phi^T y$

Computational Complexity

$$N = 13\,671\,875$$

$$d = 36$$

$$y \in \mathbb{R}^N$$

$$\Phi \in \mathbb{R}^{N \times d}$$

$$\theta^* = (\Phi^\top \Phi)^{-1} \Phi^\top y$$

(A) Compute $\Phi^\top \Phi$

(B) Compute $\Phi^\top y$

(C) Solve $(\Phi^\top \Phi)\theta = \Phi^\top y$

Q: Which is the most expensive step?

Computational Complexity

$$N = 13\,671\,875$$

$$d = 36$$

$$y \in \mathbb{R}^N$$

$$\Phi \in \mathbb{R}^{N \times d}$$

$$\theta^* = (\Phi^\top \Phi)^{-1} \Phi^\top y$$

(A) Compute $\Phi^\top \Phi$

(B) Compute $\Phi^\top y$ — $\mathcal{O}(N \cdot d)$

(C) Solve $(\Phi^\top \Phi)\theta = \Phi^\top y$

Q: Which is the most expensive step?

Computational Complexity

$$N = 13\,671\,875$$

$$d = 36$$

$$y \in \mathbb{R}^N$$

$$\Phi \in \mathbb{R}^{N \times d}$$

$$\theta^* = (\Phi^\top \Phi)^{-1} \Phi^\top y$$

(A) Compute $\Phi^\top \Phi$ $\quad \quad \quad - \mathcal{O}(N \cdot d^2)$

(B) Compute $\Phi^\top y$ $\quad \quad \quad - \mathcal{O}(N \cdot d)$

(C) Solve $(\Phi^\top \Phi)\theta = \Phi^\top y$

Q: Which is the most expensive step?

Computational Complexity

$$N = 13\,671\,875$$

$$d = 36$$

$$y \in \mathbb{R}^N$$

$$\Phi \in \mathbb{R}^{N \times d}$$

$$\theta^* = (\Phi^\top \Phi)^{-1} \Phi^\top y$$

(A) Compute $\Phi^\top \Phi$ — $\mathcal{O}(N \cdot d^2)$

(B) Compute $\Phi^\top y$ — $\mathcal{O}(N \cdot d)$

(C) Solve $(\Phi^\top \Phi)\theta = \Phi^\top y$ — $\mathcal{O}(d^3)$

Q: Which is the most expensive step?

Computational Complexity

$$N = 13\,671\,875$$

$$d = 36$$

$$y \in \mathbb{R}^N$$

$$\Phi \in \mathbb{R}^{N \times d}$$

$$\theta^* = (\Phi^\top \Phi)^{-1} \Phi^\top y$$

(A) Compute $\Phi^\top \Phi$ — $\mathcal{O}(N \cdot d^2)$

(B) Compute $\Phi^\top y$ — $\mathcal{O}(N \cdot d)$

(C) Solve $(\Phi^\top \Phi)\theta = \Phi^\top y$ — $\mathcal{O}(d^3)$

$$(A) > (B) \gg (C)$$

Q: Which is the most expensive step?

$$N = 13671875$$

$$d = 36$$

Fitting a Polynomial

2. Least-Squares Fitting Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \|\Phi^\top \theta - y\|_2^2$$

$$N = 13671875$$

$$d = 36$$

Fitting a Polynomial

2. Least-Squares Fitting Problem with Lasso Regularization

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \|\Phi^\top \theta - y\|_2^2$$

$$N = 13671875$$

$$d = 36$$

Fitting a Polynomial

2. Least-Squares Fitting Problem with Lasso Regularization

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

$$N = 13671875$$

$$d = 36$$

Fitting a Polynomial

2. Least-Squares Fitting Problem with Lasso Regularization

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

we want *sparse* solutions

$$N = 13671875$$

$$d = 36$$

Fitting a Polynomial

2. Least-Squares Fitting Problem with Lasso Regularization

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

we want *sparse* solutions

Q: Why does this give a sparse solution?

$N = 13671875$

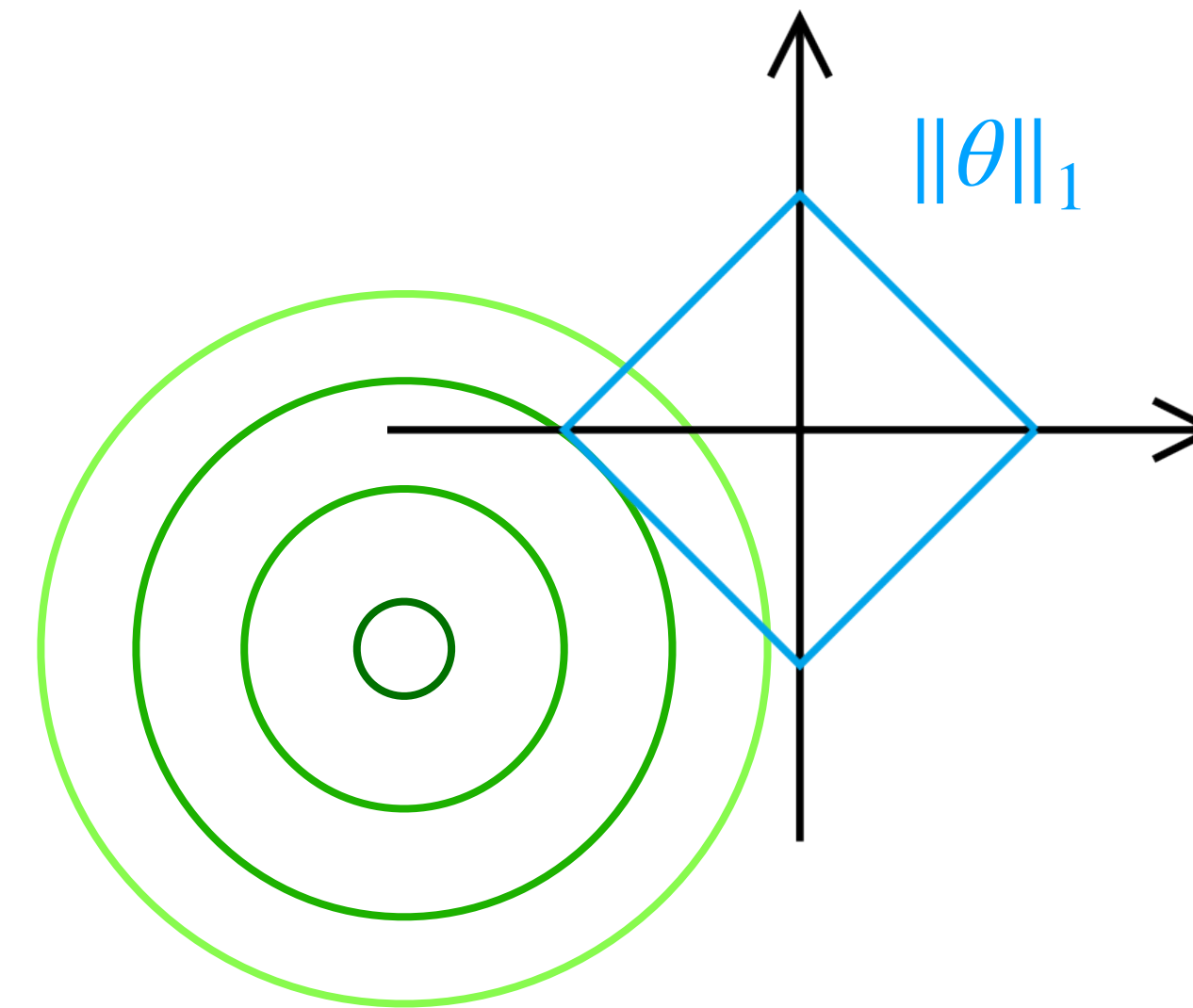
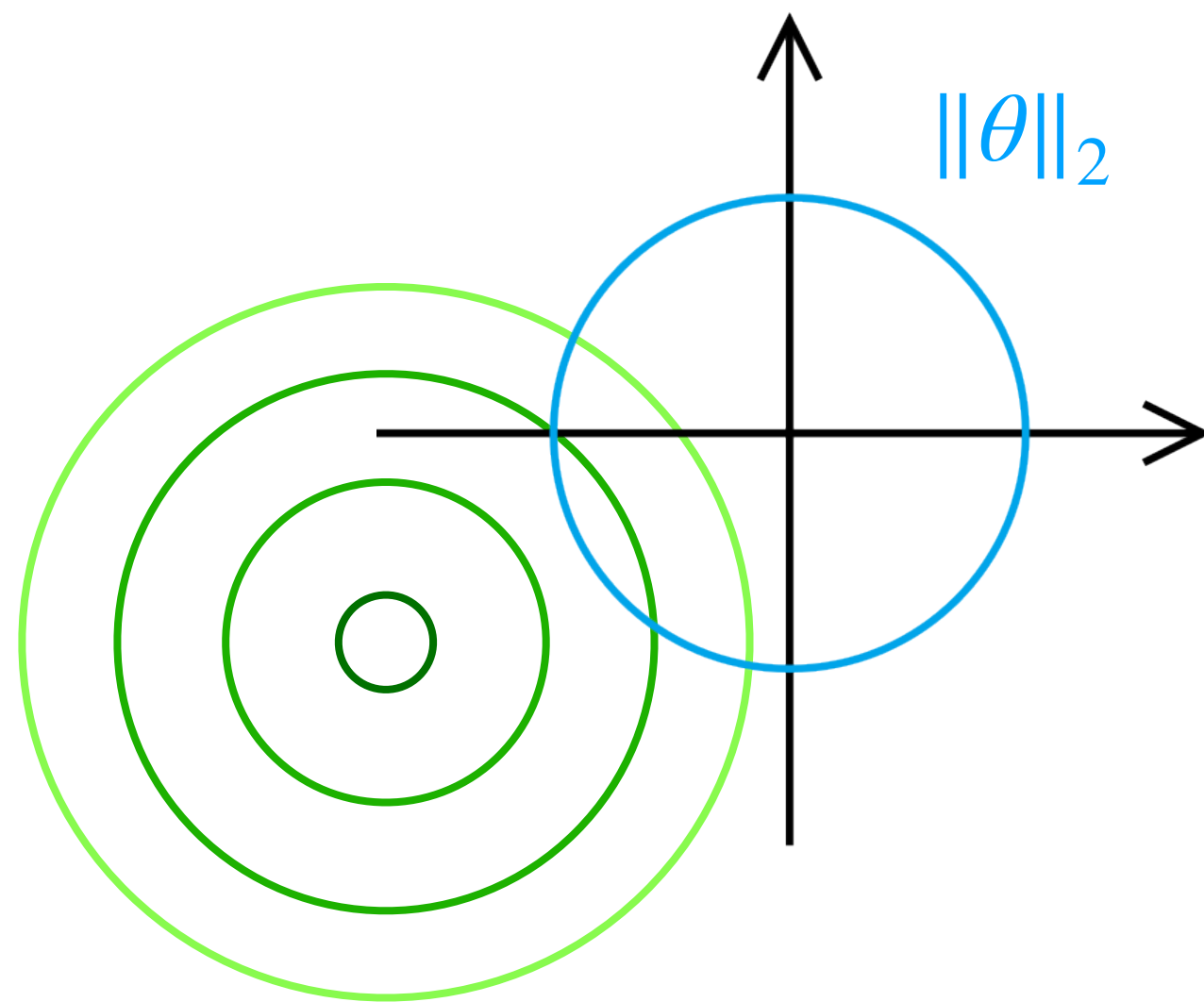
$d = 36$

Fitting a Polynomial

2. Least-Squares Fitting Problem with Lasso Regularization

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

we want *sparse* solutions



Q: Why does this give a sparse solution?

$$N = 13\,671\,875$$

$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

$$N = 13\,671\,875$$
$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation

$$N = 13\,671\,875$$
$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation



$$N = 13\,671\,875$$
$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation



$$N = 13\,671\,875$$
$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation



$$N = 13\,671\,875$$
$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation



$$N = 13\,671\,875$$
$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation



`linear_model.Lasso()`

$$N = 13\,671\,875$$
$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation



`linear_model.Lasso()`

which uses *coordinate gradient descent*

$$N = 13\,671\,875$$
$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

Solution

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation



`linear_model.Lasso()`

which uses *coordinate gradient descent*

$$N = 13\,671\,875$$

$$d = 36$$

Fitting a Polynomial

3. Solving the Problem

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \|\Phi^\top \theta - y\|_2^2 + \gamma \|\theta\|_1$$

Implementation



`linear_model.Lasso()`

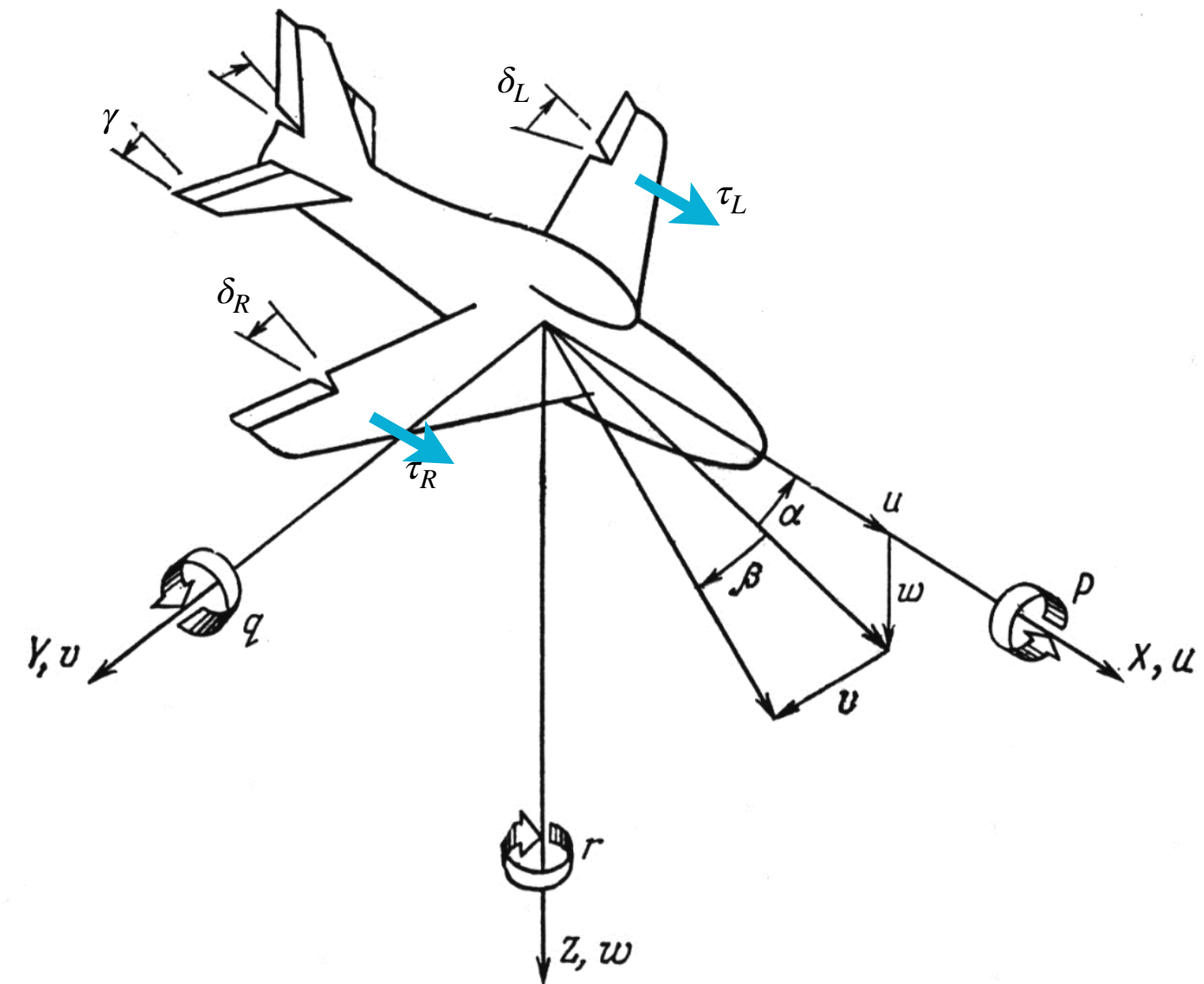
which uses *coordinate gradient descent*

Solution

$$\begin{aligned} \tilde{F}_x(s) = & -1.145 \cdot u \\ & + 0 \cdot u^2 \\ & + 0 \cdot u^3 \\ & + 0 \cdot v \\ & - 0.0498 \cdot v^2 \\ & + 0 \cdot v^2 \\ & \vdots \\ & + 0 \cdot \beta^3 \\ & + 15.3012 \cdot 1 \end{aligned}$$

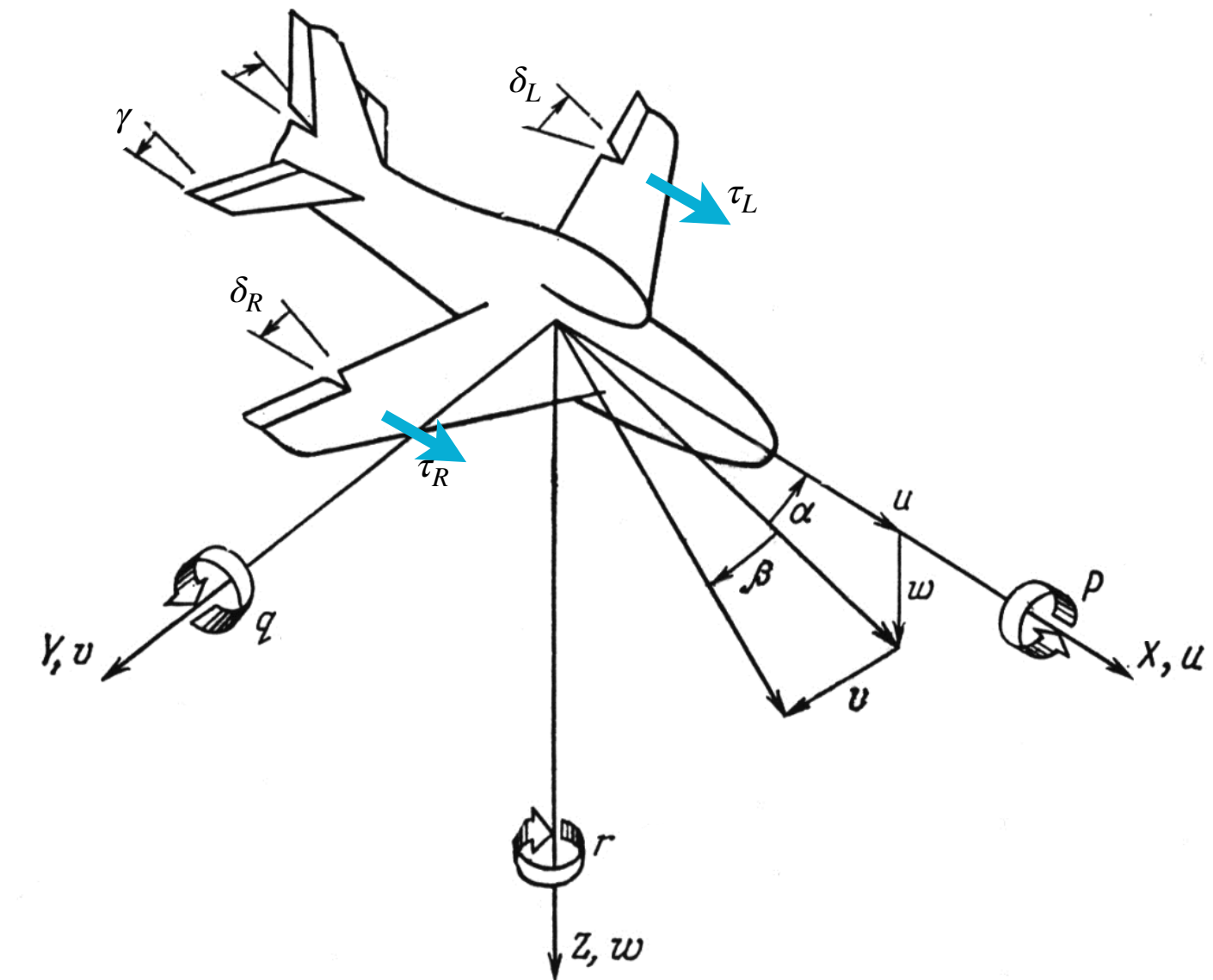
Fitting Results

$$\tilde{F}_x(s) = \tilde{F}_x(u, v, w, p, q, r, \tau_L, \tau_R, \delta_L, \delta_R, \gamma)$$



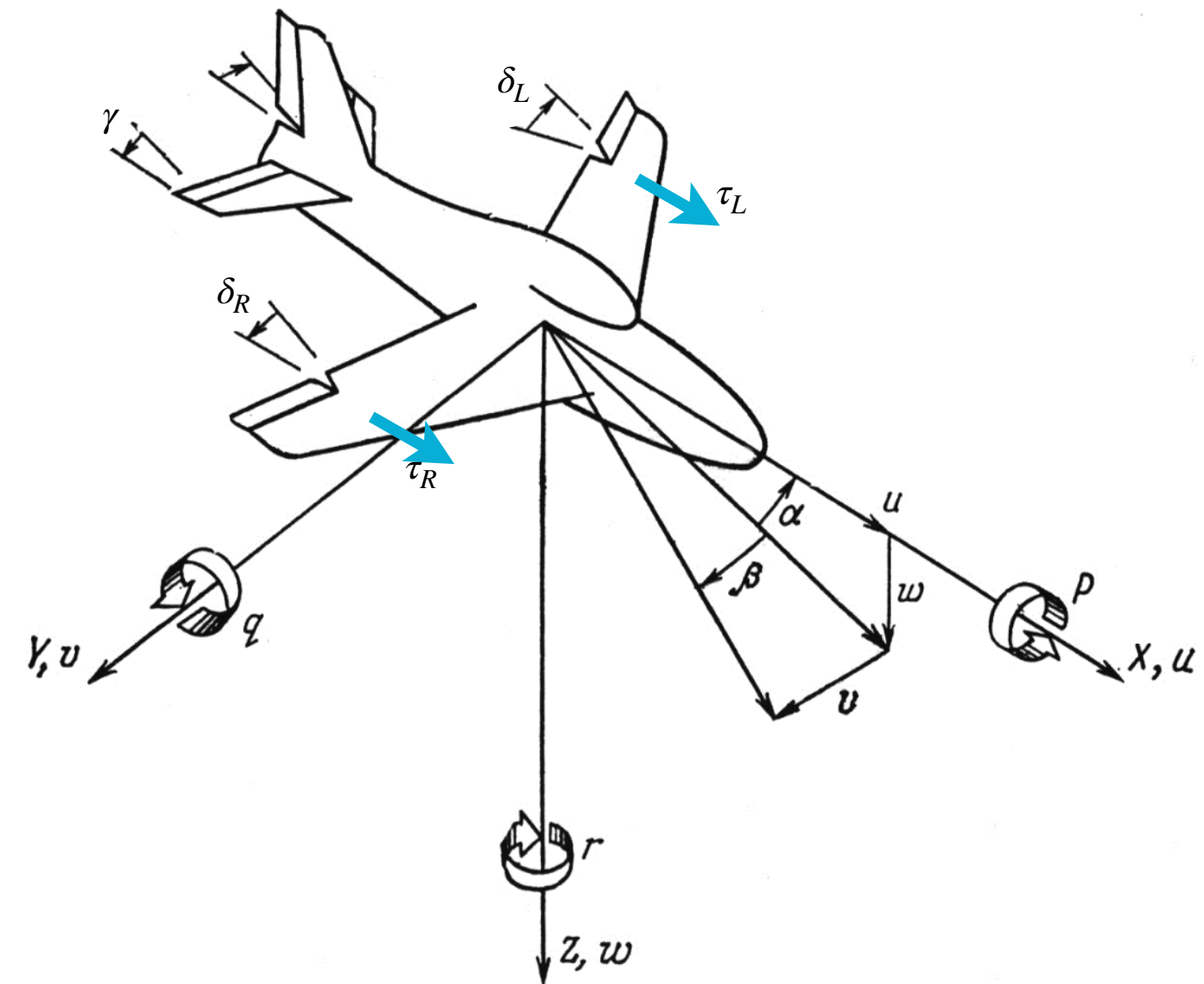
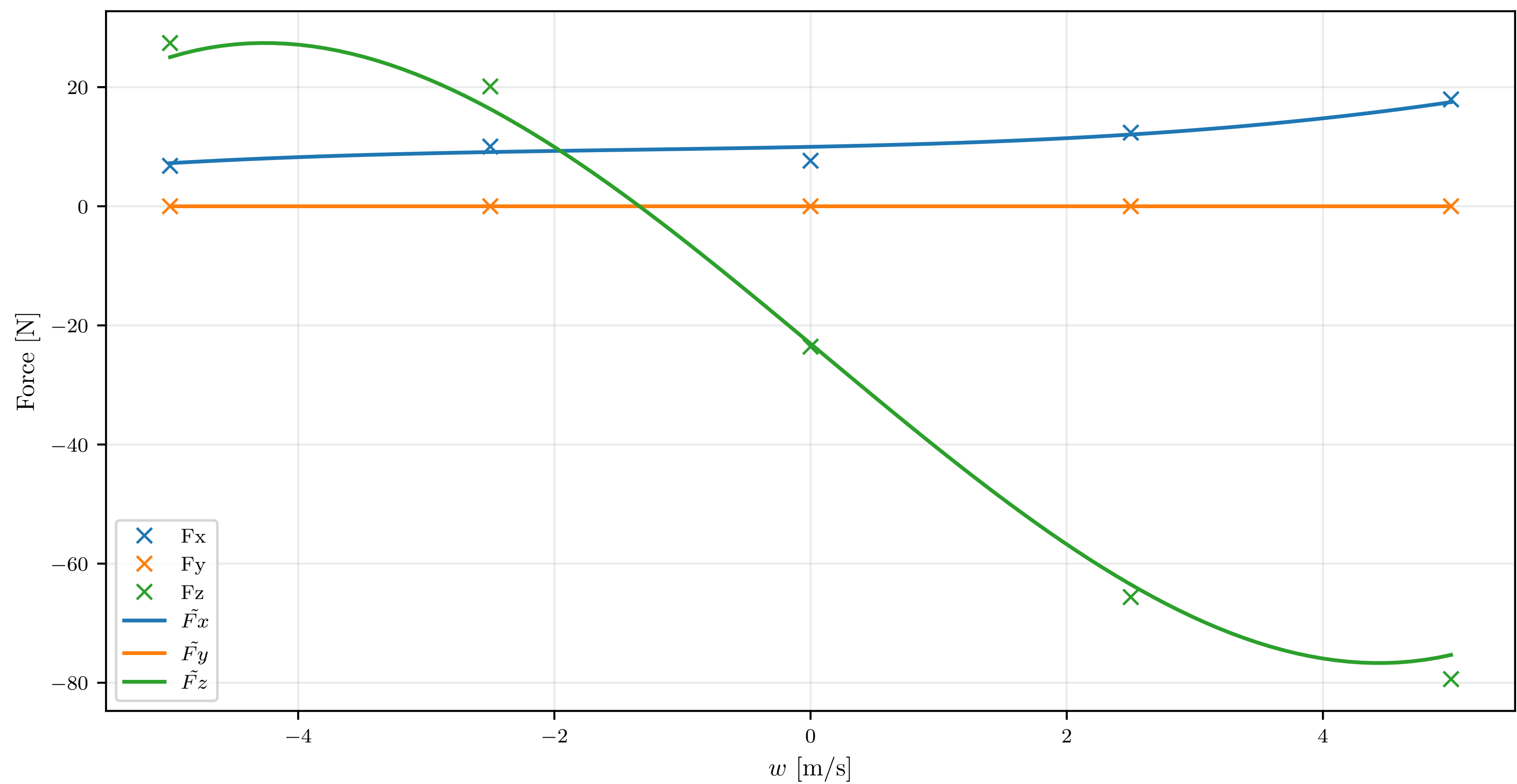
Fitting Results

$$\begin{aligned}\tilde{F}_x(s) &= \tilde{F}_x(u, v, w, p, q, r, \tau_L, \tau_R, \delta_L, \delta_R, \gamma) \\ &= \tilde{F}_x(18 \text{ m/s}, 0, w, 0, 0, 0, 15 \text{ N}, 15 \text{ N}, 0, 0, 0)\end{aligned}$$



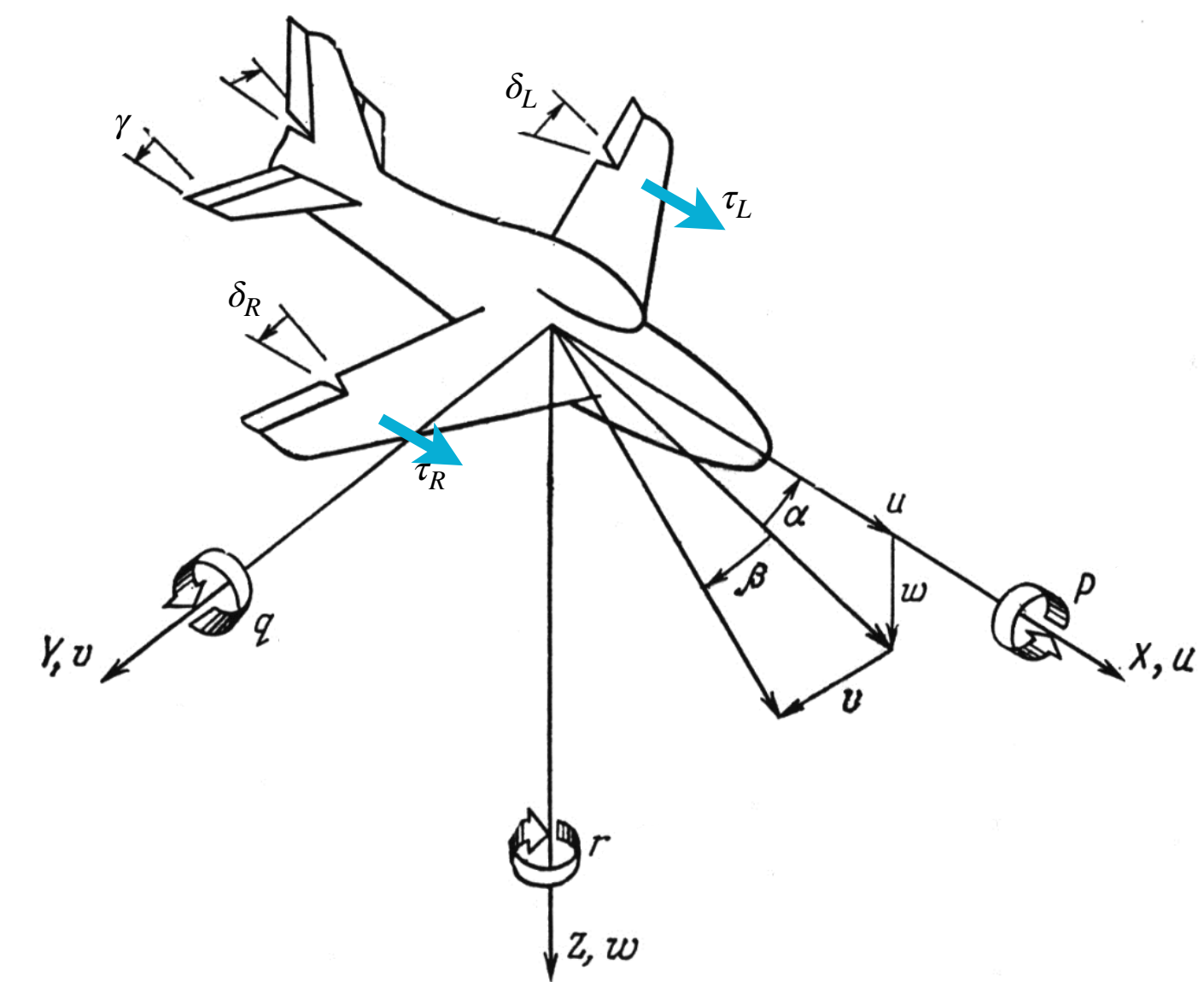
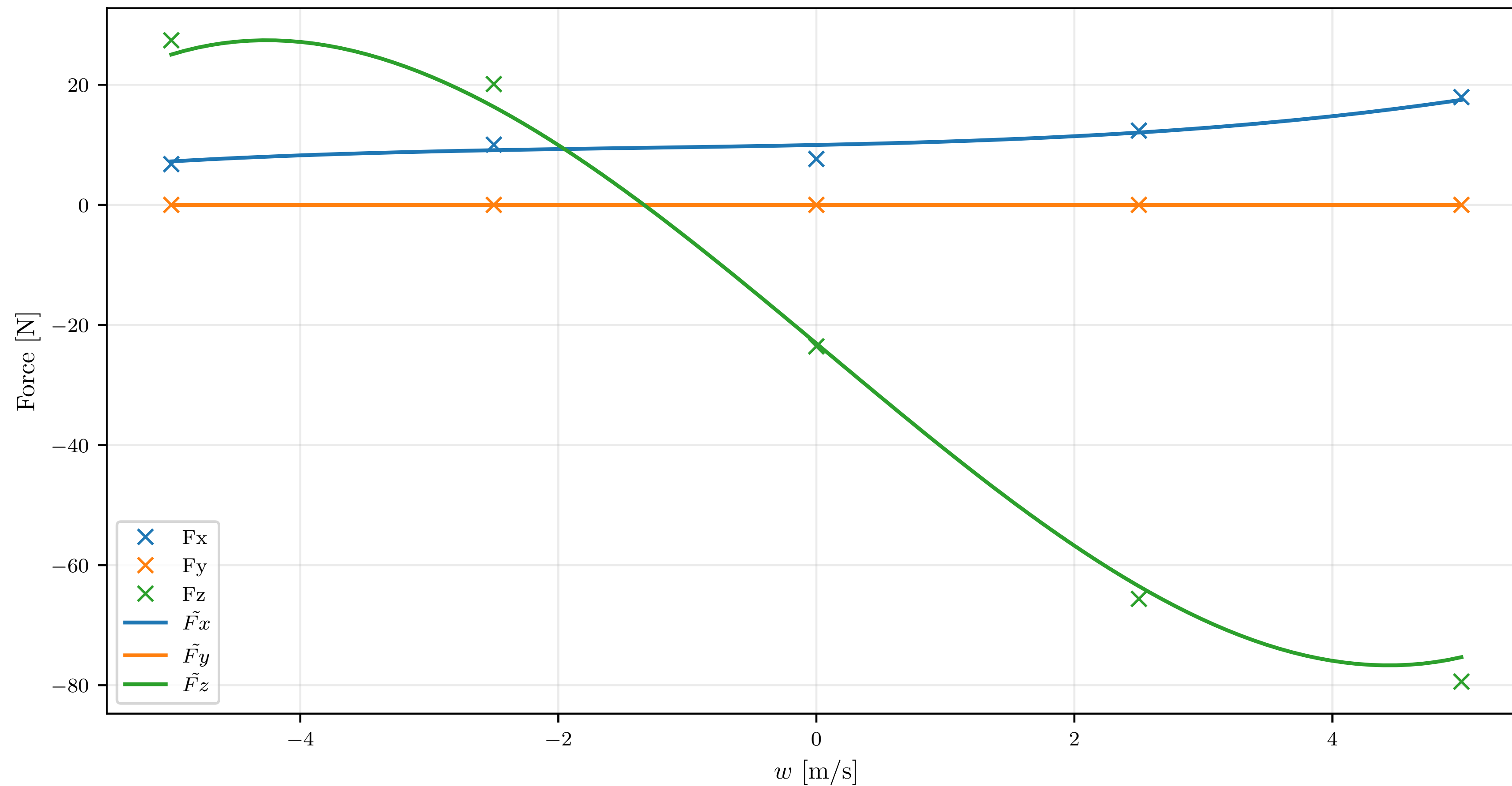
Fitting Results

$$\begin{aligned}\tilde{F}_x(s) &= \tilde{F}_x(u, v, w, p, q, r, \tau_L, \tau_R, \delta_L, \delta_R, \gamma) \\ &= \tilde{F}_x(18 \text{ m/s}, 0, w, 0, 0, 0, 15 \text{ N}, 15 \text{ N}, 0, 0, 0)\end{aligned}$$



Fitting Results

$$\begin{aligned}\tilde{F}_x(s) &= \tilde{F}_x(u, v, w, p, q, r, \tau_L, \tau_R, \delta_L, \delta_R, \gamma) \\ &= \tilde{F}_x(18 \text{ m/s}, 0, w, 0, 0, 0, 15 \text{ N}, 15 \text{ N}, 0, 0, 0)\end{aligned}$$



Q: Do these Results make sense?

... Putting it all Together

The Final Model

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{R} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ m^{-1} \left(R \tilde{F}_{\text{aero,b}}(x, u) + F_g \right) \\ R[\omega]_{\times} \\ J^{-1} \left(\omega \times J\omega + \tilde{M}_{\text{aero,b}}(x, u) \right) \end{bmatrix}$$

