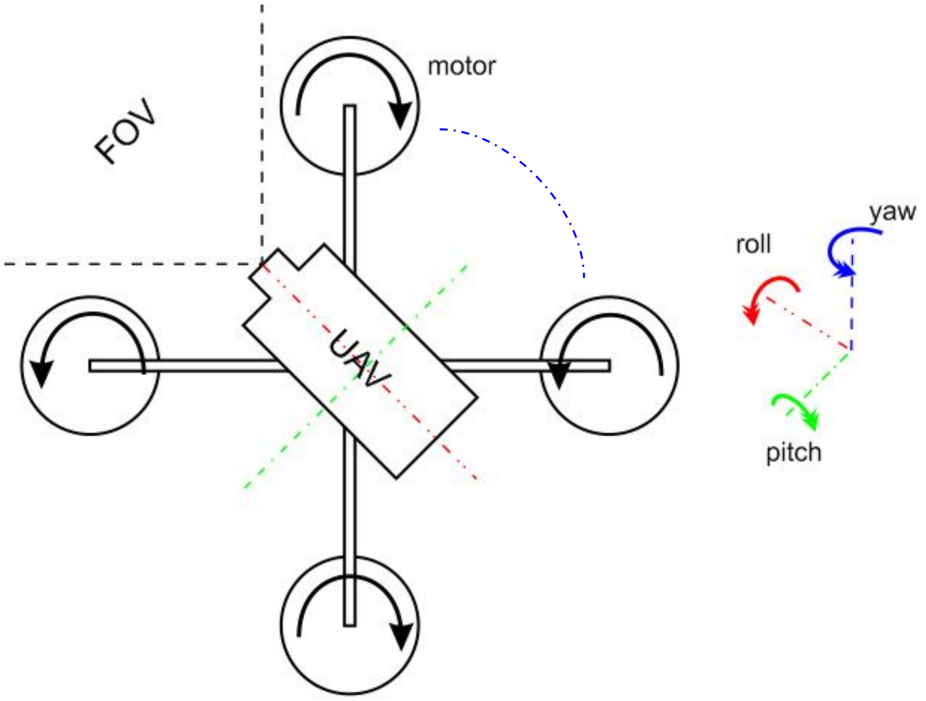
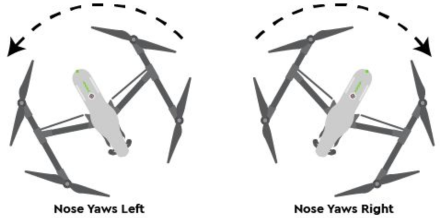
Geometric Tricopter Controller

By Tomasz Frelek

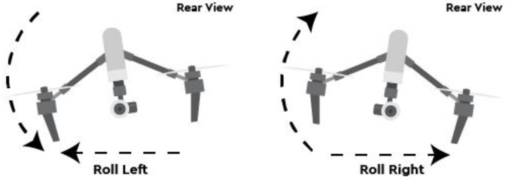
Attitude Control



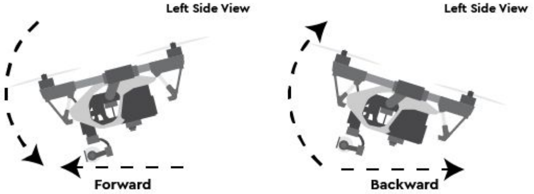
Yaw:



Roll:



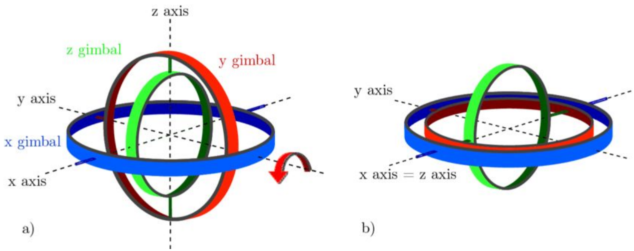
Pitch:



3D Space Representation

Representations of 3D Space (Attitude & Position)

- **Euclidean:**
  - **Mechanism:** Represents position as a vector [x, y, z] and orientation using three angles (Roll, Pitch, Yaw).
  - **Pros:** Intuitive; easy to visualize and debug.
  - **Cons:** Suffers from Gimbal Lock (mathematical singularity where degrees of freedom are lost when rotation axes align).
- **Quaternions:**
  - **Mechanism:** Represents orientation using a 4-dimensional unit vector q = [w, x, y, z].
  - **Pros:** Completely avoids Gimbal Lock; computationally efficient (no trigonometric functions required for composition).
  - **Cons:** Less intuitive, innate ambiguity (q = -q).
- **Geometric:**
  - **Mechanism:** Represents orientation using Rotation Matrices (on SO(3)) and position as vectors, treating the system as evolving on a curved manifold (SE(3)).
  - **Pros:** Works for (almost) any orientation
  - **Cons:** Mathematical complexity.



SE(3)

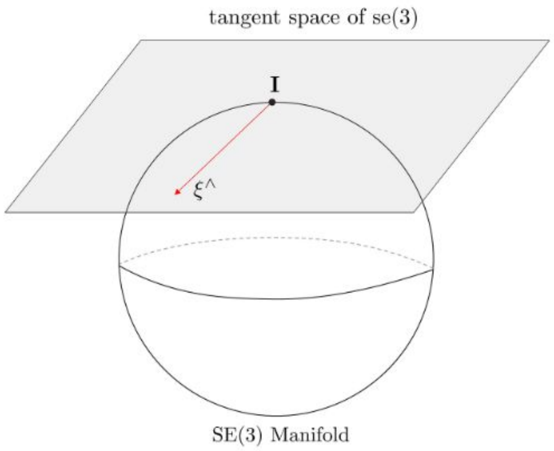
SE(3) is the semi-direct product of two distinct spaces:

- **R (Translation):** The position of the drone's center of mass (x, y, z).
- **SO(3) (Rotation):** The Special Orthogonal group. This represents orientation using 3X3 Rotation Matrices.

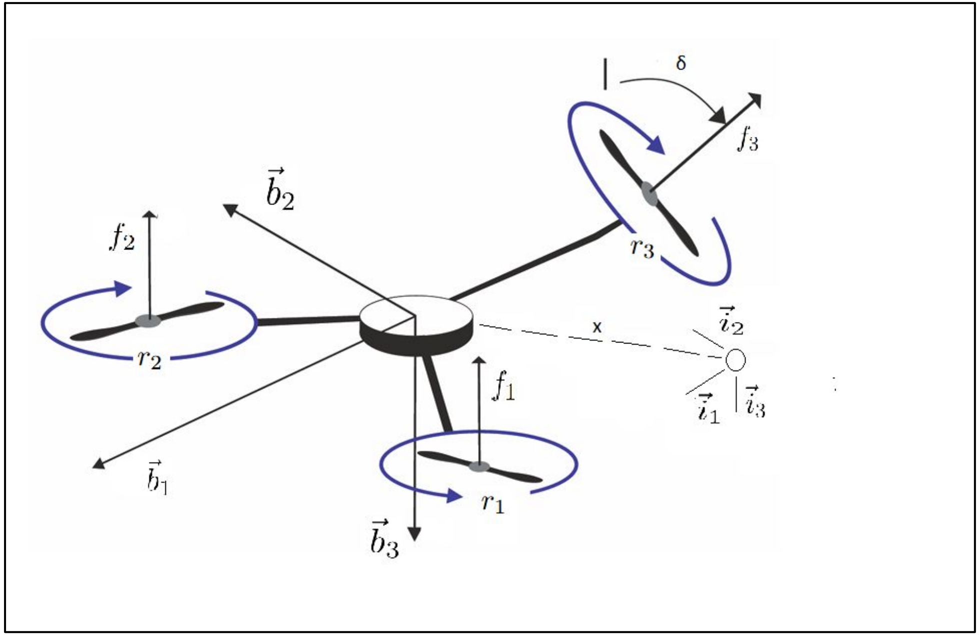
An element g in SE(3) is often written as a 4 X 4 homogeneous transformation matrix: $g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$

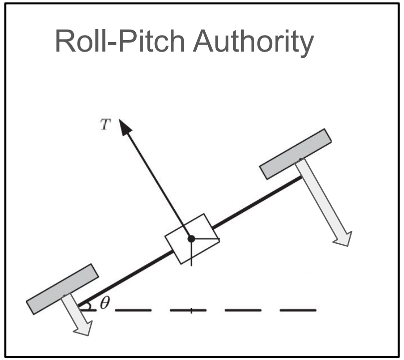
Where:

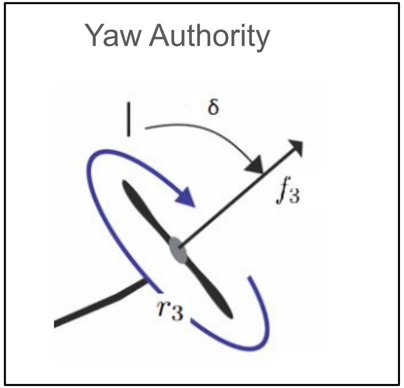
- R in R^{3x3} is the rotation matrix.
- p in R^3 is the translation vector.



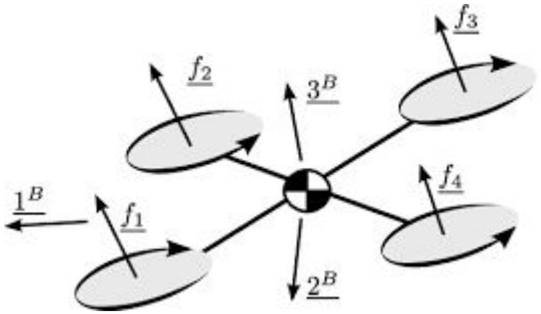
Tricopter Model







Actuation Coupling

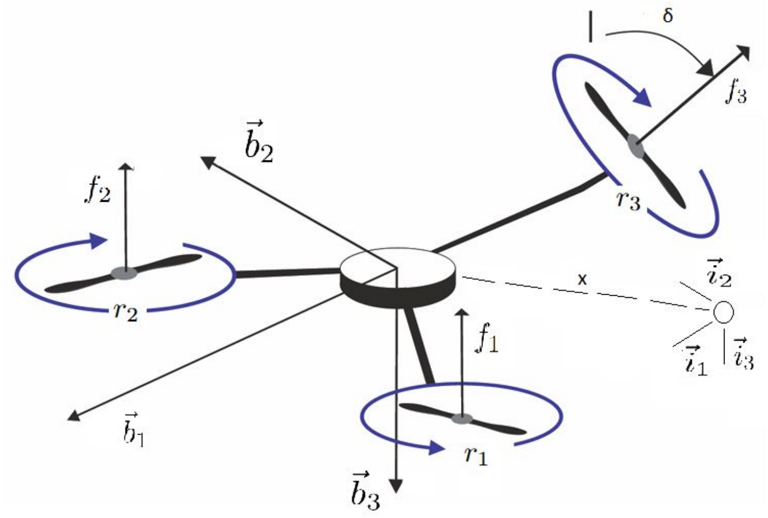


Quadrotor Actuation:

- Total thrust is strictly fixed to the body z-axis (b3).
- Yaw is generated by torque (pure moment, no side force).
- *Result:* Translational and Rotational dynamics are largely decoupled.

Tricopter Actuation:

- Yaw is generated by vectoring the rear rotor thrust.
- Tilting the servo creates a lateral force component.

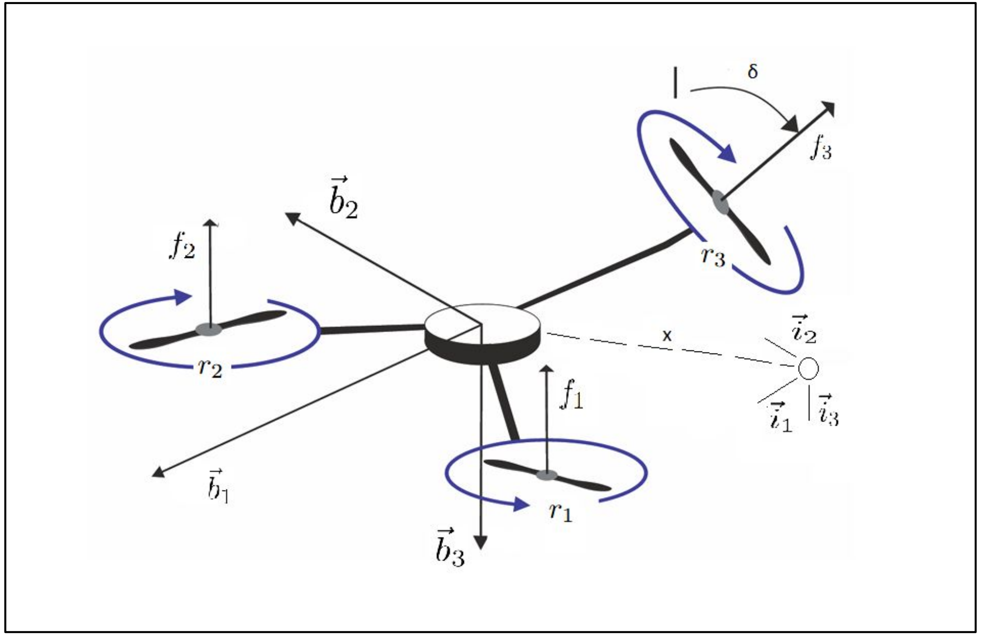


Actuation Coupling:

- You **cannot** yaw without also pushing the uav (parasitic force).

$$F_b = \begin{bmatrix} 0 \\ f_3 \sin\delta \\ -(f_1 + f_2 + f_3 \cos\delta) \end{bmatrix}$$

Definitions



Frames of Reference:

- Inertial Frame: $\{\vec{i}_1, \vec{i}_2, \vec{i}_3\}$
- Body-Fixed Frame: $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$

Actuation Inputs:

- 3 Motor Thrusts: $f_1\, f_2\, f_3$
- 1 Servo Tilt Angle: δ

Equations of Translational Motion:

$$\dot{x} = v$$

$$m\dot{v} = mge_3 + RF_b$$

Equations of Rotational Motion:

$$\dot{R} = R\hat{\Omega}$$

$$J\dot{\Omega} + \Omega \times J\Omega = M$$

Net Thrust and Moment:

$$F_b = \begin{bmatrix} 0 \\ f_3 \sin\delta \\ -(f_1 + f_2 + f_3 \cos\delta) \end{bmatrix} \qquad M = \begin{bmatrix} d_y(f_1 - f_2) \\ -d_x(f_1 + f_2) + d_t f_3 \cos\delta \\ -d_t f_3 \sin\delta \end{bmatrix}$$

Controller Overview

1. High-Level Inputs

- **Reference Trajectory:** Desired position $x_d(t)$, velocity $v_d(t)$, and heading direction b_{1d}.
- **Current State:** Actual position x, velocity v, attitude R, and angular velocity Ω.

2. Rotational Control (Inner Loop)

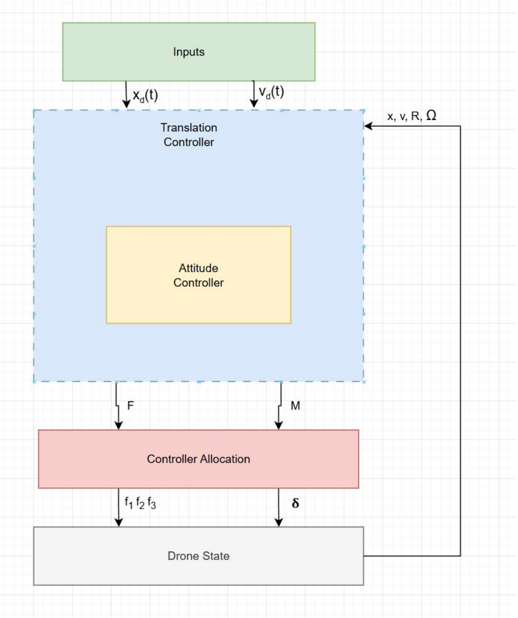
- **Attitude Construction:** Combines the desired vertical axis b_{3d} with the desired heading b_{1d} to form the full target rotation matrix R_d.
- **Error Calculation:** Computes the geometric orientation error e_R and rate error eΩ on the SO(3) manifold.
- **Virtual Moment Output:** Calculates the required body torque M to snap the vehicle to R_d.

3. Translational Control (Outer Loop)

- **Error Calculation:** Computes position and velocity errors (e_x, e_v).
- **Virtual Force Output:** The controller acts like a spring-damper system, calculating a total thrust vector to correct the path.

4. Nonlinear Control Allocation

- **Mapping:** Converts the virtual commands (f, M) into physical actuator signals using the tricopter geometry.
    - *Actuators:* 3 Motor Thrusts (f_1, f_2, f_3) + 1 Servo Angle (**δ**).
- **Coupling:** Automatically handles the geometric complexity of the tilting rear rotor.



Error Calculation

1. Position Error: $e_x = x - x_d$

2. Velocity Error: $e_v = v - v_d$

3. Attitude Error: $e_R = \frac{1}{2}(R_d^T R - R^T R_d)^\vee$

- **Configuration Error Function (ψ)** We define a scalar potential function to quantify the "distance" between the current and desired orientation.
  - This serves as an artificial potential energy that is zero only when the attitudes match perfectly.
- To generate a feedback control torque, we need a vector direction. We derive this by taking the derivative of ψ on the tangent space of SO(3).
- We calculate e_R by taking the difference between the relative rotation matrix and its transpose. This operation isolates the Skew-Symmetric part of the matrix.
  - In Lie Algebra terms, a skew-symmetric matrix is just a rotation vector disguised as a matrix.
- We use the Vee Map to extract the x, y, z components. The resulting vector e_R tells us exactly the axis we need to rotate around to fix our orientation.

4. Angular Velocity Error: $e_\Omega = \Omega - R^T R_d \Omega_d$

$$\Psi(R, R_d) = \frac{1}{2}\mathrm{tr}[I - R_d^T R]$$

- **Intrinsic:** Independent of the coordinate frame.
- **Bounded:** 0 <= ψ <= 2.
- **Minima:** ψ = 0 if and only if R = R_d.
- **R**: Current Attitude .
- **R_d**: Desired Attitude.
- **tr**: Matrix Trace operator (sum of diagonal elements).
- **I**: Identity Matrix.

Lyapunov Candidate Function

We aim to prove exponential stability of the attitude dynamics. To do this, we construct a The Lyapunov Candidate Function V_R .

- A generalized Energy Function for our system.

For a function to be a valid candidate to prove stability, it must satisfy two main rules:

1. **Positive Definite** V(x) > 0:
   The energy must be positive everywhere except at the target, where it is zero.
2. **Negative Derivative** d/dx V(x) < 0:
   The "energy" must strictly decrease over time along the system's path.

If we can define a distance from our target (Positive Definite), and we can prove that this distance is always shrinking (Negative Derivative), we must eventually hit the target (go to zero).

$$\mathcal{V}_R = \underbrace{\frac{1}{2} e_\Omega^T J e_\Omega}_{\text{Kinetic}} + \underbrace{k_R \Psi(R, R_d)}_{\text{Potential}} + \underbrace{c_2 e_R \cdot e_\Omega}_{\text{Crossing Term}}$$

Variable Definitions & Intuition

- **J (Inertia Matrix):**
  - The tensor representing the vehicle's mass distribution.
  - Represents the drone's resistance to rotation.
- **k_R (Attitude Gain):**
  - The proportional control gain.
  - The "stiffness" of the virtual spring pulling the drone upright. Higher values mean a stronger restoring force.
- **c (Interaction Constant):**
  - A small positive constant used to couple errors.
  - A standard energy function (KE + PE) only proves that the system eventually stops moving (Lyapunov Stability), adding c helps prove exponential decay.

Exponential Stability of Attitude (Proposition 1)

$$\mathcal{V}_R = \frac{1}{2} e_\Omega \cdot J e_\Omega + k_R \Psi(R, R_d) + c e_R \cdot e_\Omega$$

1. To prove exponential decay, we bound V_R by the squared norm of the state vector $z_R = [\|\|e_R\|\|, \|\|e\Omega\|\|]$. W_1 is some positive definite lower bound, W_2 is some generic upper bound (ψ)	$z_R^T W_1 z_R \leq \mathcal{V}_R \leq z_R^T W_2 z_R$ $\quad W_1 = \frac{1}{2}\begin{bmatrix} k_R & -c \\ -c & \lambda_{min}(J) \end{bmatrix}, \quad W_2 = \frac{1}{2}\begin{bmatrix} \frac{4k_R}{2-\psi} & c \\ c & \lambda_{max}(J) \end{bmatrix}$
2. We can take the time derivative of V_R	$\dot{\mathcal{V}}_R \leq -z_R^T W_3 z_R$ $\quad W_3 = \begin{bmatrix} \frac{ck_R}{\lambda_{max}(J)} & -\frac{ck_\Omega}{2} \\ -\frac{ck_\Omega}{2} & k_\Omega - c \end{bmatrix}$
3. Combining the derivative with the upper bound we get exponential differential inequality	$\alpha = \frac{\lambda_{min}(\hat{W}_3)}{\lambda_{max}(W_2)}$ $\quad \dot{\mathcal{V}}_R \leq -\frac{\lambda_{min}(W_3)}{\lambda_{max}(W_2)} \mathcal{V}_R = \mathcal{V}_R(t) \leq \mathcal{V}_R(0) e^{-\alpha t}$
4. Plug back into the bounded error and solve	$\lambda_{min}(W_1) \|\|z_R(t)\|\|^2 \leq \mathcal{V}_R(t) \leq \mathcal{V}_R(0) e^{-\alpha t}$ $\quad \|\|z_R(t)\|\| \leq \sqrt{\frac{\mathcal{V}_R(0)}{\lambda_{min}(W_1)}} e^{-\frac{\alpha}{2} t}$

Thus stability error decays exponentially fast!

Uniformly Ultimately Bounded Translational Stability (Proposition 2)

1.	For translational dynamics, we cannot prove exponential stability b/c of the parasitic force, instead we aim for UUB	$m\dot{e}_v = \underbrace{-k_x e_x - k_v e_v}_{\text{Control}} + \underbrace{R\Delta}_{\text{Parasitic Force}} + \underbrace{\xi(R)}_{\text{Vanishing}}$
2.	We again define a Lyapunov candidate function	$\mathcal{V}_x = \frac{1}{2}k_x\|\|e_x\|\|^2 + \frac{1}{2}m\|\|e_v\|\|^2 + c_1 e_x \cdot e_v$
3.	We take the time derivative	$\dot{\mathcal{V}}_x \leq \underbrace{-\lambda_{min}(M_3)\|\|z_x\|\|^2}_{\text{Stabilizing (Quadratic)}} + \underbrace{\delta_{max}\theta\|\|z_x\|\|}_{\text{Destabilizing (Linear)}}$ $\dot{\mathcal{V}}_x < 0 \implies \lambda_{min}(M_3)\|\|z_x\|\|^2 > \delta_{max}\theta\|\|z_x\|\|$
4.	Thus the ultimate bound is where the two terms equal each other	$\|\|z_x\|\| > \frac{\delta_{max}\theta}{\lambda_{min}(M_3)} \triangleq \mu$

(Almost) Global Attractiveness

The Condition for Stability

- Proposition 2 established UUB for translational errors, but this proposition is only valid when $\psi < 1$ (< 90 degrees)
- What if the initial condition is outside this region: $1 < \psi(0) < 2$

Exponential Decay of Attitude

From Proposition 1, we established that the attitude dynamics on SO(3) are exponentially stable globally .

- The attitude error function $\psi(t)$ decreases monotonically regardless of the translational state.
- Therefore, for any initial error $\psi(0)$, there exists a finite time $t^* > 0$ such that: $\psi(t^*) = 1$ and $\forall t \geq t^*, \Psi(t) < 1$

During the interval $[0, t^*)$, the attitude condition for Prop 2 is not yet met.

- However, since the interval $[0, t^*)$ is finite and the system acceleration is physically bounded, the translational errors (e_x, e_v) can only grow by a finite amount.
- The state trajectory remains bounded during this transient phase.

At time t = t*, the system enters the Region of Attraction defined in Proposition 2.

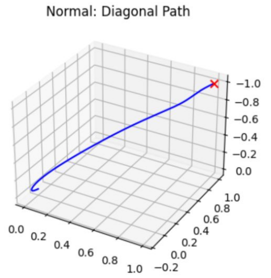
- For all t >t*, the condition $\psi(t) < 1$ holds.
- Consequently, the translational errors (e_x, e_v) stop drifting and begin to converge toward the ultimate bound.

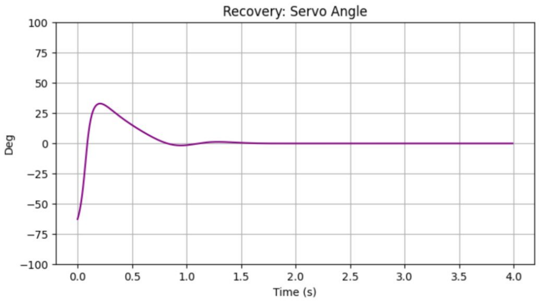
What if ψ = 2 (a rotation error of exactly 180 degrees)?

- At 180 degrees, the skew-symmetric part of the rotation error matrix vanishes: $(R_d^T R - R^T R_d) = 0 \implies e_R = 0$
- This is a singularity of measure 0
- However, any perturbation moves the state to $\psi < 2$, triggering exponential recovery!

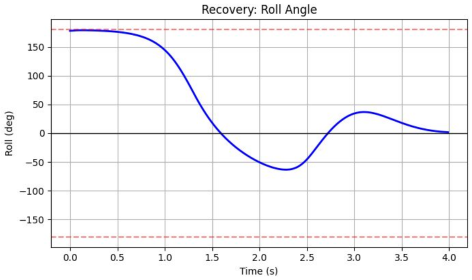
Simulation Data

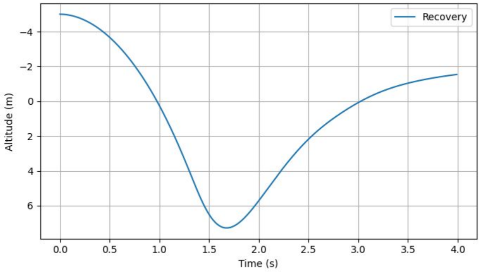
Starting stationary:
normal diagonal flight

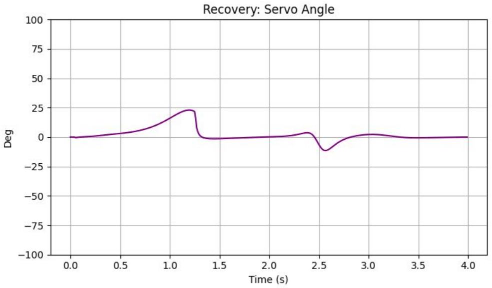




Starting overturned (178 degrees)







Thank-You!